INFERENCE-DRIVEN SEMANTIC ANALYSIS

Martha Stone Palmer

SDC - A Burroughs Company
and
University of Pennsylvania

ABSTRACT

A primary problem in the area of natural language processing is the problem of semantic analysis. This involves both formalizing the general and domain-dependent semantic information relevant to the task involved, and developing a uniform method for access to that information. Natural language interfaces are generally also required to have access to the syntactic analysis of a sentence as well as knowledge of the prior discourse to produce a semantic representation adequate for the task. This paper briefly describes previous approaches to semantic analysis, specifically those approaches which can be described as using *templates*, and corresponding multiple levels of representation. It then presents an alternative to the template approach, inference-driven semantic anaylsis, which can perform the same tasks but without needing as many levels of representation.

1. Introduction

Inference-driven semantic analysis is specifically designed for finite, well-defined, i.e., limited, domains. The domain on which a Prolog implementation of this method was tested consists of physics word problems for college students involving pulley systems. Each problem is stated in English sentences that completely describe a miniature world of physical objects and relationships between those objects. The goal of the natural language processor is to produce a semantic representation of each problem that is detailed enough to enable a computer program to produce the correct solution of the problem. This semantic representation consists of a set of partially instantiated logical terms known as *semantic predicates*.

The formalization of the domain is essential for solving the following basic problems which are associated with the semantic processing of text.

(1) establishing referents for the noun phrases;

(2) finding appropriate mappings from the syntactic constituents of the parse into the underlying semantic representation of the verb, (For inference-driven semantic analysis this representation is defined primarily by a semantic predicate associated with the verb and semantic roles acting as arguments to the predicate. The syntactic consituents are essentially mapped directly onto these semantic roles.);

(3) using pragmatic information to assign fillers to semantic roles do not have an explicit syntactic realization, (the term "pragmatic" is used to refer to both discourse knowledge and general and domain-dependent information);

(4) applying inference rules to *expand* the representation of the verb into a more detailed representation that fulfills the requirements of the processing task;

(5) constraining allowable inferences so that this semantic representation does not become explosive;

(6) appropriately integrating the final representation of the clause with the representations of prior clauses.

Previous approaches to semantic analysis suffer from one of two drawbacks. If an attempt is being made to capture linguistic generalizations such as "case," the processing methods degenerate into verb specific procedures that are completely domain dependent, making the implementations difficult to transport to other domains [Schank], [Simmons]. Placing the emphasis on more transparent processing, i.e., separating the relevant linguistic information from the computational methods used to process that information, results in processing techniques that use several levels of representation. While computationally more modular, this approach does not adequately capture linguistic generalizations [Woods], [Pereira and Warren]. Not being able to use these generalizations efficiently leads to unecessary redundancies which make these systems cumbersome for large domains [Palmer, 81].

In general, these more modular semantic processors can be described as using basically three levels of semantic representation which are illustrated below. The first level, referred to here as the *template level*, corresponds to a set of patterns that represent the possible syntactic realizations of sentential units for an individual verb. Each <slot> in the template represents the position of a syntactic constituent in a particular realization. The slots usually have semantic markers associated with them, referring to the semantic role the syntactic consituent is expected to play with respect to the verb. (The semantic markers in the example refer to physical objects, PHYS-OBJ, and location points on those objects, LOC-PT.) Specific syntactic parses can be matched directly with these templates. Matching parses onto templates achieves the mapping of the syntactic consituents onto the underlying semantic representation, task 2 from above.

The second step is to match the templates with an intermediate level, the *canonical level* which is sometimes termed the "case-frame level." The canonical level consists basically of the verb or the predicate chosen to represent the verb, and a union of all of the semantic roles the syntactic consituents can be associated with. Inference rules can then be applied to this intermediate level to first fill unfilled semantic roles, task 3, and then expand the representation of the verb, task 4, to produce the final and third level, the *predicate level*. Care must be taken to constrain the application of these inference rules, task 5, and then task 6, the integration of the representation with the prior discourse, must be achieved.

SENTENCE:
A particle is attached to the end of a string.

TMEPLATE:
<PHYS-OBJ1> attach to <LOC-PT2> of <PHYS-OBJ2>

CANONICAL FORM:
attach(<PHYS-OBJ1>,<PHYS-OBJ2>,
<LOC-PT1>,<LOC-PT2>)

PREDICATES:
contact(PHYS-OBJ1,PHYS-OBJ2)
locpt(LOC-PT1,PHYS-OBJ1)
locpt(LOC-PT2,PHYS-OBJ2)
sameplace(LOC-POINT1,LOC-POINT2)

This paper presents an alternative to the template approach which is capable of performing the six tasks simultaneously: inference-driven semantic analysis. Section 2 introduces domain-specific inference rules as lexical entries for verbs and gives an example of how these rules can be applied to derive appropriate semantic representations. Section 3 suggests the task of finding appropriate mappings from syntactic constituents to semantic roles, (task 2), as being largely responsible for the development of the template approach with its corresponding intermediate level of semantic representation. The method by which inference-driven semantic analysis performs the mappings while expanding the verb representation, (task 4), without having to make use of a similar intermediate level of representation, is explained here as well as in section 5. Sections 4, 6, and 7 describe how the remaining tasks are also performed during the expansion of the verb representation. The only task not addressed specifically is task 1, reference evaluation. Noun phrases are assumed to be fully determined along the lines suggested by Mellish's Incremental Evaluation [Mellish].

2. Domain-specific inference rules

The lexical entries of the verbs take the form of inference rules. Producing a semantic representation of a particular sentential unit can be seen as "proving" that the verb involved has been used appropriately for the domain. Prolog is an obvious language for such an approach and the rules are expressed as Prolog clauses. (In conventional Prolog notation Q <- P corresponds to P -> Q; strings starting with upper-case letters correspond to variables; and predicates, function symbols, and constants are all lower-case.) For example, rule R1, the lexical entry for "attach," can be read as, "A contact between an object, (O1), and another object, (O2), can be expressed using the verb attach."

R1:
attach <- contact(object1(O1),object2(O2)).

The goal is to use R1 in producing an adequate semantic representation for sentence like,

S1:
"A particle is attached to a string at its right end."

S1 contains two major syntactic constituents, the "particle" which is the, SUBJECT and the "string," which is the object of "TO." (Prepositional phrases are indicated by "PP" preceded by the preposition involved, as in "TO-PP.") The next section explains how the following semantic representation is produced,
contact(object1(particle),object2(string)), given a predicate representation of the syntactic information:

SUBJECT(particle), TO-PP(string)

3. Performing Mappings

Semantic representations generally associate a distinct set of semantic roles with each individual verb. One of the main goals of any semantic processor is to provide an appropriate mapping between the syntactic constituents of a parsed clause and the semantic roles associated with the verb. Three factors complicate the mapping:

(1) the large number of choices available for syntactic realization of any particular semantic role,
(2) the ability of syntactic constituents to indicate several different types of roles given appropriate contexts, and
(3) semantic role interdependencies, i.e., the appropriateness of a mapping for a particular semantic role is often dependent on the mappings given to the other semantic roles.

These complications have previously been coped with by creating sets of templates for a verb, one for each syntactic realization. Each set of templates must have associated with it an individual set of domain-specific inference rules so that deeper semantic representations can then be derived. Domains often involve semantically or syntactically similar verbs that still have to be dealt with on an individual basis using this approach, resulting in unecessary redundancies.

Inference-driven semantic analysis uses a set of "mapping rules" to guide the instantiation of predicate arguments with the referents of surface syntactic constituents. The mapping rules make use of intuitions about syntactic cues for indicating semantic roles first embodied in the notion of case [Fillmore, 68]. For the application of these rules to be useful, it is essential they preserve the same semantic role interdependencies handled by templates. This is accomplished by making the application of the mapping rules "situation-specific." Some mapping rules, such as "SUBJECT to AGENT," are quite general and can apply in many situations. Other rules, such as "WITH-PP to INSTRUMENT," are much less general, and can only apply under a set of specific circumstances. For some verbs, in order for the WITH-PP rule to apply, the AGENT must be mentioned explicitly in the syntactic realization, as in "John broke the vase with a hammer." Checking for the mention of the AGENT disallow inappropriate applications of this rule, such as, "* The vase broke with a hammer." As described below, the application of mapping rules can be constrained by the use of a *predicate environment.*

An example of an unconstrained mapping rule involves the object1 from R1. Object1's are similar to PATIENTS, and like PATIENTS can usually be indicated by the SUBJECT. For an unconstrained rule, the predicate environment is simply a variable, Y, as in M1.

M1: object1(X) <- SUBJECT(X) / Y

Given S1, application of this rule would result in object1(O1) being instantiated with "particle."

The mapping rule for the object2 from the "attach" example is not as general, and gives an example of how rules can be constrained. An object2 of a contact relationship can be indicated by a TO-PP, but an object2 of a support relationship cannot. In order to make the application of the rules situation-specific, the predicate environment can be partially instantiated. It contains information about the "context" of the semantic role, exemplified here by the relation name and the other arguments. The predicate environment for object2 on the right hand side of R1 is "contact(object1(O1),object2(O2))." By associating a "contact" predicate environment with the TO-PP mapping rule, as in the following example, the rule can be restricted to object2's which are arguments to contact predicates. This allows the term object2(O2) to be instantiated with the "string."

M2: object2(X) <- TO-PP(X) / contact(Y,object2(X))

The associated predicate environments are equivalent to Joshi-Levy Local Constraints, and as such act as filters on the possible mappings for the arguments, so that the final set of mappings arrived at is within the context-sensitive limitations of the domain [Joshi and Levy], [Palmer, 83].

Section 8 describes the implementation of the semantic processor that aplies Rule M2 to the "attach" example, instantiating object2(O2) with "end." This achieves the preliminary representation "contact(object1(particle),object2(end))" mentioned in the previous section.

4. Filling gaps in semantic roles

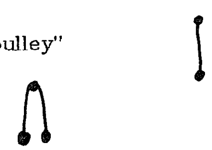
It is generally accepted that many semantic roles, such as AGENTS and INSTRUMENTS, are syntactically optional, and do not always appear in the surface structure of a sentential unit. Just because these roles are not mentioned does not guarantee that they do not need to be filled. In "The door was opened with a key," the passivization and the presence of the INSTRUMENT "key" indicate clearly that an AGENT exists although s/he is not referred to. It is sometimes possible to deduce the referent of the AGENT from pragmatic information about the local context, as in:

How did the burglar get inside? The door was opened with a key.

For the semantic processor to perform this type of deduction it must have access to domain-dependent information in the form of inference rules, i.e., pragmatic information. Semantic role fillers that are not made explicit in the syntactic realization of a verb can sometimes be retrieved from the local context or hypothesized from general knowledge about the domain. There are examples of these implicit semantic role fillers in the mechanics domain. In "the pulley is suspended from a pulley," it is clear from pragmatic information about suspension that a STRING, or some type of flexible line segment, is doing the "suspending," but it is never mentioned explicitly. This sentential unit is followed by "and offset by a particle," meaning that the pulley is being counter-balanced by a particle, as in the following figure. The appropriate representation for "offset" can only be achieved if pragmatics can supply "string" as a default value in the preceding representation of "suspend." The way this is accomplished is explained in more detail below.

"pulley is suspended from a pulley"

"and offset by a particle"



Inference-driven semantic analysis makes a distinction between semantic roles that are syntactically obligatory or optional, and semantic roles that are semantically obligatory or optional. Traditionally, syntactically obligatory semantic roles have to occur in a syntactic realization of the sentential unit and syntactically optional roles do not. Roles that are considered to be syntactically optional but semantically obligatory are termed *essential* roles, and the classification of semantic roles as semantically optional, essential or obligatory is used to constrain the application of pragmatic inference rules as follows: Semantically optional roles are simply marked as "absent," essential roles are filled by deduction, and unfilled obligatory roles cause failure resulting in the derivation of a new set of mappings. Fillers for essential roles can be deduced in any of three ways.

(1) There can be known default values associated with the role.

(2) A possible filler can be hypothesized from general world knowledge. (Default values are really just short cuts to hypothesizing fillers.)

(3) The filler can be supplied by context as in the burglar example.

In this domain, intermediaries, (similar to INSTRUMENTS), are considered to be essential roles, so when an intermediary is not mentioned explicitly as in "a pulley is suspended from another pulley," the processor allows pragmatics to supply a "string" as a default value. The inference rules associated with "offset," in trying to represent a "counter-balancing" event, need to know what the "pulley" has been supported by in order to copy the support relationship. Since the "supporter" of the pulley, the "string" was filled in the analysis of "a pulley is suspended from another pulley," local context can now supply that "string" as the supporter of the "particle."

This section and the preceding section have explained how semantic roles can be filled by syntactic constituents or by pragmatic deduction, tasks 2 and 3. These two tasks are simply two different methods of finding instantiations for the predicate arguments, and can be performed as part of the application of rule R1. The ability to instantiate arguments by syntactic consituents or by pragmatic deduction is essential for the correct integration of the sentence representation within the current model of the scene being described, as explained in the following sections.

5. Inferring relationships BETWEEN semantic roles

In the template approach, the canonical level lists the semantic roles associated with the verb, but does not make the relationships between these semantic roles explicit. What does it mean for John to be the AGENT of "break" and for the vase to be the PATIENT? In going from the canonical level to the predicate level, inference rules must be applied to spell out these relationships. This is equivalent to task 4, expanding the verb representation.

For inference-driven semantic analysis, the application of further inference rules such as R2 make the relationships between the semantic roles explicit. The initial inference rule, R1, differed from the canonical level by not including all of the semantic roles that could be included in the canonical level. Now the arguments of R2 provide the semantic roles that were not included in R1. An example sentence containing several of the optional semantic roles found in this rule is, "The string has a weight attached at its left end." The following rule, R2, can be read as "If a location point on an object, locpt(L1), and a location point on another object, locpt(L2) are at the sameplace, then the objects are in contact with each other." Location points are classified as essential roles, so that if they are not filled by syntactic constituents they have to be deduced.

R2:

```
contact(object1(O1),object2(O2)) <-
    locpt(locpt(L1),object1(O1)),
    locpt(locpt(L2),object2(O2)),
    sameplace(locpt(L1),locpt(L2)).
```

6. Controlling inferences

Limiting the application of further inference rules to rules involving semantic roles helps to solve the problem of uncontrolled inferences, task 5. Other inference rules may come into play when pragmatic information is used to fill semantic roles, but the application of pragmatic rules is constrained by the role-filling task. R2 also demonstrates the high degree of generalization that can be achieved, since rules such as R2 are shared by most of the verbs in the domain. Support and location relationships all eventually require contacts being made explicit. They can also all have location points for the contacts indicated by AT prepositional phrases, another generalization.

7. Integrating semantic representations

To achieve an appropriate representation for a sentence, the piece of the scene being described must be represented accurately, and this representation must be integrated correctly with the current model of the scene derived from previous sentences, the 6th task. Semantic roles play an important part in the necessary integration since pragmatic information can sometimes use objects that have already been described to fill roles that are not mentioned explicitly in the sentence. This was illustrated by the "offset" example. Another important component of successful integration is reference evaluation. Previously described objects can be referred to directly in order to provide new information about them. Correct evaluation of such references is crucial to distinguishing between the information in the sentence that is "given" and the information that is "new."

In summary, a key component of inference-driven semantic analysis is the division of domain-specific inference rules into rules that explicitly spell out relationships between semantic roles, and rules that provide domain information useful for deducing fillers for these roles. As the first set of rules is applied to expand the semantic representation, the second set can be used along with syntactic mapping rules to instantiate the arguments of the first set. In our example, mapping rules M1 and M2 achieved the following instantiations for the arguments of R1, where the sentence being analyzed was, "A particle is attached to a string at its right end."

contact(object1(particle),object2(string))

The R2 is applied to further expand the representation, and the following predicates are produced:

locpt(locpt(particle),object1(particle))
locpt(locpt(rtend),object2(string))
sameplace(locpt(particle),locpt(rtend))

Another mapping rule gets applied to fill in the location point of the string with "rtend," and pragmatics decides that a particle, being the shape of a point, can be its own location point. These inferences correspond to an appropriate semantic representation of the sentence, and are produced directly from the original set of syntactic constituents by applying the aforementioned rules. This process simultaneously performs the six tasks outlined in the introduction. In performing these tasks, it is not necessary to go through levels of representation corresponding to templates and case-frames, since all of the information normally contained at these levels is now contained in the mapping rules and the inference rules themselves.

8. Implementation

The semantic processor draws inferences and instantiates arguments by imposing a procedural interpretation on the inference rules very similarly to the way that Prolog imposes a procedural interpretation on Horn clauses. The verb inference rules are in fact Horn clauses, and the arguments to the predicates are terms that consist of function symbols with one argument. The procedural interpretation drives the application of the inference rules, and allows the function symbols to be "evaluated" as a means of instantiating the arguments. The predicate environments associated with the constraints on instantiation correspond to possible snapshots of the procedural interpetation of the rules. These allow the same argument to be constrained differently depending on the instantations of the other arguments or on the particular predicate. The inferences that are drawn in this way correspond to the set of predicates that make up the semantic representation of the clause.

ACKNOWLEDGEMENTS

I would especially like to thank Jim Weiner, Bonnie Webber and Barbara Grosz for their many useful insights into problems in semantic analysis, their comments on this paper, and their encouragement and support.

References

Bundy, et-al, "Solving Mechanics Problems Using Meta-Level Inference, Expert Systems in the Micro-Electronic Age," Michie, D.(ed), Edinburgh University Press, Edinburgh, UK, 1979

Fillmore, C., The case for case, Universals in Linguistic Theory, Bach and Harms (eds.) New York; Holt, Rinehart and Winston, pp. 1-88. 1980.

Joshi, Aravind K., and Levy, Leon S., an expanded version of "Phrase Structure Trees Bear More Fruit Than You Would Have Thought," 18th Annual Meeting of the Association for Computational Linguistics," University of Pennsylvania, Philadelphia, PA, June, 1980.

Kowalski, Robert, Logic for Problem-Solving, North-Holland Pub. Co., 1979.

Levin, B. "Predicate-Argument Structures in English," Master's Thesis Proposal, MIT, 1977.

Levin, B. "Instrumental With and the Control Relation in English," MIT Master's Thesis, MIT AI Memo 552, 1979. 1976.

Palmer, M., "A case for Rule-Friven Semantic Processing," ACL Conference Proceedings, Stanford University, 1981

Palmer,M., "Driving Semantics for a Limited Domain," PhD Dissertation, Universtiy of Edinburgh, pending, 1983

Pereira, Fernando C.N. and David H.D. Warren. Definite clause grammars for language analysis- a survey of the formalism and a comparison with augmented transition networks. "Artificial Intelligence" 13:3(1980), 231-278.

Schank, Roger C., ed Conceptual Information Processing. Amsterdam: North Holland, 1975.

Simmons, R.F., Semantic Networks: Their Computation and Use for Understanding English Sentences, Computer Models of Thought and Language, Schank and Colby (eds.) San Francisco: W.H. Freeman and Co., 1973.

Woods, William A. Progress in natural language understanding: An application to lunar geology. AFIPS Conference Proceedings 42 (1973b), 441-450.

Woods, William A., Procedural Semantics as a Theory of Meaning, in Elements of Discourse Understanding, (eds. Joshi, Sag, Webber), Cambridge U. Press, 1981, pp. 300-334.