COLING 82, J. Horecký (ed.)
North-Holland Publishing Company

DIALOGIC: A CORE NATURAL-LANGUAGE PROCESSING SYSTEM

Barbara Grosz, Norman Haas,[1] Gary Hendrix,[1] Jerry Hobbs, Paul Martin,
Robert Moore, Jane Robinson, Stanley Rosenschein

Artificial Intelligence Center
SRI International
Menlo Park, California 94025
U.S.A.

The DIALOGIC system translates English sentences into representations of their literal meaning in the context of an utterance. These representations, or "logical forms," are intended to be a purely formal language that is as close as possible to the structure of natural language, while providing the semantic compositionality necessary for meaning-dependent computational processing. The design of DIALOGIC (and of its constituent modules) was influenced by the goal of using it as the core language-processing component in a variety of systems, some of which are transportable to new domains of application.

OVERVIEW

The DIALOGIC system translates English sentences into representations of their literal meaning in the context of an utterance. These representations, or "logical forms," are intended to be a purely formal language that is as close as possible to the structure of natural language, while providing the semantic compositionality necessary for meaning-dependent computational processing. The design of DIALOGIC (and of its constituent modules) was influenced by the goal of using it as the core language-processing component in a variety of systems, some of which are transportable to new domains of application.

Currently DIALOGIC is a core component of four systems being developed within several different research projects at SRI. One is the TEAM project,[2] whose goal is to provide natural-language access to large data bases through systems that are easily adaptable to a wide range of new applications. Another, the KLAUS project,[3] is a longer-range effort to address basic research problems in natural-language semantics, commonsense reasoning, and the pragmatics. A third project[4] is investigating the problem of providing natural-language access to text.[4] A fourth, in which DIALOGIC also plays an important role, is examining the development of formal grammars.[5]

DIALOGIC is divided into five modules coordinated by the DIAMOND executive system. DIAMOND is a modification of the executive system used in the SRI speech-understanding project [Walker 1978] and also in a task-dialogue interpretation

---

[1] Now working at Machine Intelligence Corporation in Sunnyvale, California.

[2] Sponsored by the Defense Advanced Research Projects Agency under Contract N00039-80-C-0645.

[3] Sponsored by the Defense Advanced Research Projects Agency under Contract N00039-80-C-0575.

[4] Funded by the National Library of Medicine under Grant 1R01-LM03611.

[5] Sponsored by the National Science Foundation under Grant IST-8103550.

system [A. Robinson, 1980]. It provides the formal language for defining the grammar and the control for parsing English sentences and translating them into logical-form expressions.

The five modules are (1) the DIAGRAM grammar; (2) a set of semantic translators; (3) a set of basic semantic functions; (4) a scoping algorithm (for quantifiers and sentence operators); (5) a set of basic pragmatic functions. The remainder of this paper describes these components of DIALOGIC and presents an example illustrating how they coordinate in the interpretation of an utterance. A description of the logical form that is the target of DIALOGIC's interpretation processes may be found in [Moore, 1981].

DIAGRAM

DIAGRAM is a general grammar of English. It now contains about 125 rule schemata, equivalent to about 800 individual rules. These define all common sentence types, complex auxiliaries and modals, complex noun phrases, nominalized sentences, all the common quantifiers, relative clauses, verbs with sentential complements, comparative and measure expressions, subordinate clauses and other adverbial modifiers. Conjunction, however, is limited to a few place-holders, pending further study of the problems it poses for constraining the number of syntactic analyses. A detailed description of DIAGRAM is contained in [J. Robinson, 1982]. Formally, DIAGRAM is an augmented phrase-structure grammar. The lexicon categorizes words and associates attributes with them that are used in the rules. Each rule has associated with it a constructor that expresses the constraints on its application and also a translator (described in the next section) that produces the corresponding logical form.

Phrases inherit attributes from their constituents and acquire attributes from the larger phrases that contain them. These attributes are used to impose context-sensitive constraints upon the acceptance of an analysis. Before constructing a node in the parse tree corresponding to the application of a rule, the executive invokes the rule's constructor to test for admissibility. In addition to accepting or rejecting a rule application, the constructors can assign scores that allow listing alternative analyses in a preferred order. The result of applying the grammar to analysis of an input is one or more annotated parse trees.

Attributes and annotations are not limited to syntactic information. The translators, described next, specify how the translation of a phrase into logical form is to be defined in terms of the attributes of the words and phrases that compose it. This coupling of syntax and semantics (for which attribute grammars [Tienari 1980] were originally designed) is convergent with current formal theories of natural language that advocate constructing a syntax and semantics that "work in tandem" [Dowty et al. 1981; Kaplan and Bresnan (to appear); Gazdar (to appear); Landsbergen 1976.]

Future work on DIAGRAM includes efforts to extend both its coverage and its formalism. In extending the formalism, our dual objective is to capture certain linguistic generalizations (e.g., dative movement) and to make the task of developing a large grammar more manageable. To accomplish this, we are exploring the use of metarules [Gazdar to appear].

TRANSLATORS

Following the syntactic analysis of an utterance, a sequence of semantic translators is invoked to build the logical form that corresponds to a literal interpretation of the utterance in context. The translator for each phrase-structure rule specifies how the various constituents of the phrase are to be combined to form an interpretation of the whole phrase. It prescribes the predicate-argument structures that correspond to the grammatical construction or, more generally, the operator-operand structures.

Although the translators operate top-down (the translator for each node invokes the translators for its children), the translation is in effect built bottom-up--since, typically, the first thing a translator for a nonterminal node does is to invoke the translators for each of its constituents, usually left to right. However, the top-down nature of the translation process is significant, because it means that information located above a node and to its left is available when the node is translated. In addition to producing the logical form, the translators determine the syntactic constraints upon and preferences for either coreference or noncoreference of noun phrases, especially pronouns, following an algorithm described in [Hobbs, 1976].

BASIC SEMANTIC FUNCTIONS

To insulate changes in the grammar from those that occur in logical form, the construction of the latter is isolated from the translator procedures by calls on basic semantic functions [Konolige, 1979]). The actual construction of a logical form is done in two phases: (1) logical-form fragments (lffs) are attached to the parse tree by the basic semantic functions; (2) the final logical form is assembled from these by the scoping algorithm.

Lffs are assigned only to certain nodes in the parse tree. Usually the lff at an NP node will encode the properties held by the entity the NP describes [e.g., "X such that EMPLOYEE(X) & OLD(X)" for "old employee"] and the fragment for a clause-level construction (e.g., a VP) will encode the predicate-argument structure of the clause.

The basic semantic functions also leave markers on the parse tree to indicate such things as the type of quantifier or determiner associated with a noun phrase. These markers are used by the scoping algorithm to determine the final logical form for the utterance. (Note that the lffs and markers left by the basic semantic functions may be viewed as further annotations to the parse tree.)

DIALOGIC currently includes eleven basic semantic functions. Six of these do most of the work of building lffs for standard noun phrases and clauses. The others are concerned with adding such things as mode, degree, and adverbial modification to clauses. As more precise specifications are defined for encoding these phenomena in logical form, we expect to collapse some of this latter group.

SCOPING OF QUANTIFIERS AND OTHER SENTENTIAL OPERATORS

The scoping algorithm is designed to collect the logical-form fragments from the parse tree and produce the possible scopings of quantifiers and other scoped operators. The scoping algorithm used in DIALOGIC (adapted from that in Hendrix, 1978) produces all the scopings that do not violate the hard rules of English scoping, and then ranks them according to a score computed by a set of specialist critics. Each critic is a function that returns a score for some aspect of the conflicting rules of quantification in English; e.g., the left-right scope critic lowers the score of scopings that involve permuting the left-outermost default ordering of quantifiers. All critics receive equal weight in the present implementation, but the design of the system does allow for differential weighting.

The current set of critics is concerned with such things as changes in sentence order and the relative scoping of quantifiers of different strengths. The scoping of nonstandard quantifiers and of the generalized negative ("not," "no one," "nothing," "none") remain to be done.

BASIC PRAGMATIC FUNCTIONS

Basic pragmatic functions are intended to fulfill several roles in DIALOGIC, all concerned with certain kinds of indeterminacies in logical form whose resolution requires pragmatic information. The four primary uses of basic pragmatic

functions in the current system are (1) to provide a context-specific interpretation of certain terms that have only vague meanings in themselves (e.g., prepositions like "of" and "in," or inherently vague verbs like "have"); (2) to establish the specific relationship underlying any given noun-noun combination; (3) to identify the referents of pronouns; and (4) to interpret a limited range of metonymy (e.g., the use of "blonds" to mean "people with blond hair"). At present, only a small core of pragmatic functions is implemented, each of which handles only a subset of the cases it is intended to cover.

EXAMPLE

To illustrate how the different modules of DIALOGIC contribute to the interpretation of an utterance, we shall consider the example,

"What SRI employees have children older than 15 years?"

The logical form for this query--the target for the interpretation processes--is (lowercase is used to indicate variables, uppercase to indicate constants and predicates):

```
[QUERY (WH employee1 (AND (EMPLOYEE employee1)
                          (EMPLOYEES-COMPANY-OF employee1 SRI))
                    (SOME child2 (CHILD child2) (AND (CHILD-OF employee1 child2)
                                                     ((*MORE* OLD) child2 (YEAR 15]
```

This corresponds roughly to a formal representation for "who is each employee such that the company of the employee is SRI and some child of the employee is older than fifteen years?"

During DIAMOND's parsing phase, the parse tree in Figure 1 is constructed. At this point, the attributes annotating the tree encode such properties as the type of noun (count, mass, unit) and syntactic number. These attributes have been used during the parsing phase to rule out certain alternative structures.

Once this structure is built, the translators are invoked. In combination with the basic semantic functions, the translators assign additional attributes to nodes in the tree, encoding such information as the quantifiers (type, strength, and the variables they bind) and heads of noun phrases. For example, the head of the WHNP, "what SRI employees", is a variable of type EMPLOYEE that is bound by a wh-type quantifier. Attributes also encode the underlying predicate-argument structures for verb phrases and adjectives, and the lffs to be used in constructing the final logical form for the utterance.

In the sentence of Figure 1, the nodes WHNP and S are annotated as being quantified, WHNP with a wh-type quantifier and S as a "query." Although every rule has an associated translator, only some of these result in lffs being attached to nodes. For this example, the nodes marked with ** in the original parse tree are the only ones for which lffs are produced.

The fragment attached to each of these nodes is as follows:

```
NOUN1       (*NN* employee1 SRI)
NOUN2       (EMPLOYEE employee1)
PREDICATE   (*HAVE employee1 child2)
NOUN3       (CHILD child2)
NCOMP       ((*MORE* OLD) child2 (YEAR 15))
```

EMPLOYEE, CHILD, and OLD are monadic predicates that are part of the conceptual model of the domain. *MORE* maps a predicate into a comparative along the scale corresponding to the predicate. *NN* and *HAVE are dummy predicates that indicate the need to invoke the basic pragmatic functions.

After the translation process is complete, the final logical form is assembled by a procedure that considers alternative quantifier scopings (using the quantifier-

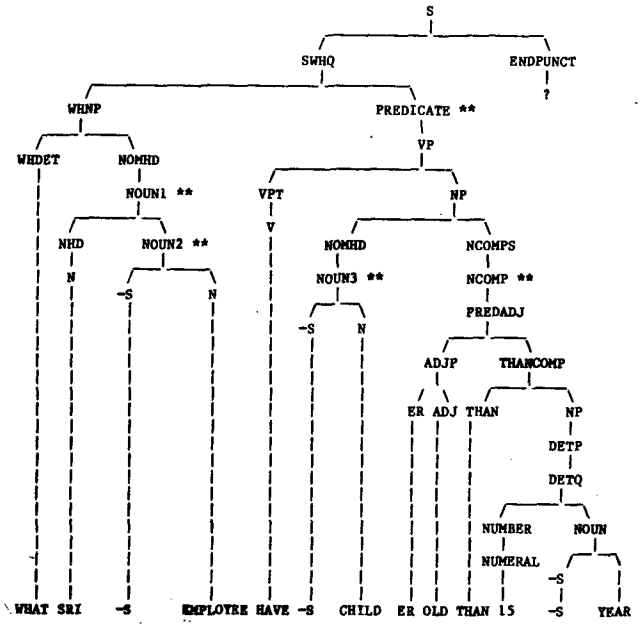

Figure 1 Parse Tree for "What SRI employees have children older than 15 years?"

related annotations left on the parse tree) and invokes the basic pragmatic functions as needed. The basic pragmatic functions use information in the conceptual model of the domain to transform (*NN* employeel SRI)--corresponding to the noun-noun compound "SRI employee"--into (EMPLOYEE-OF employeel SRI) and (*HAVE employeel child2) into (CHILD-OF employeel child2).

The nodes with either quantifier or logical-form markings are the only ones considered by the TEAM scoping algorithm. Besides the WH quantifying employeel, TEAM recognizes that a default existential quantifier must be created for child2, so SOME is added. The scope rules force QUERY to have the widest scope; this position is contested only if there are multiple sentential markers. Both orderings of the WHAT and SOME quantifiers are generated. The two resulting quantified statements correspond to (WHAT employeel (SOME child ...)...) and (SOME child (WHAT employee ...)...)

Next the scope critic functions evaluate the different scopings; only three of the critics are relevant. One critic considers the left-right node ordering and prefers the first scoping because it comes closer to the surface form. One critic prefers scopings in which WH outscopes an adjacent existential; it too upgrades the score of the first and downgrades the score of the second. The other critic knows that default existential quantifiers need the narrowest possible scope; it too selects the first.

SUMMARY

Because of the modularization in DIALOGIC, changes in one part of the system reverberate very little in other components. Changes in the constraints imposed on the phrase-structure rules in the grammar have no effect on any other part of the system. A change in a rule itself necessitates a change in the corresponding translator, but the basic semantic functions do not need to be revised. Similarly, a change in the logical form or in the data structures within which it is implemented requires a corresponding change in the basic semantic functions, but not in the grammar or translators.

In addition to extending DIALOGIC as mentioned in the foregoing sections, we are also investigating possible revisions of the translation phase (as currently performed by the translators and basic semantic functions) to allow translation into logical form to be specified declaratively. In this new approach [Rosenschein and Shieber (to appear)], logical types are associated with the phrasal categories, and the translation of a phrase is synthesized from the translations of its immediate constituents according to a local rule, which typically involves functional application.

REFERENCES

Dowty, D. R., Wall, R. E., and Peters, S. [1981] Introduction to Montague Semantics (D.Reidel Publishing Co., Dordrecht, Holland, 1981).

Gazdar, G. [to appear]. "Phrase Structure Grammar," in P. Jacobson and G. K. Pullum, eds. On the Nature of Syntactic Representation.

Hendrix, G. G. [1978] "Semantic Aspects of Translation," in Understanding Spoken Language, D. E. Walker, ed., pp. 193-226 (Elsevier, New York, New York, 1978).

Hobbs, J. [1976] "Pronoun Resolution," Research Report 76-1, Department of Computer Sciences, City College, City University of New York, New York, New York (August, 1976).

Kaplan, R. M. and Bresnan, J. W. [to appear] "Lexical-Functional Grammar: A Formal System for Grammatical Representation," Occasional Paper 13, Center for Cognitive Science, Massachusetts Institute of Technology, Cambridge, Massachusetts.

Konolige, K. G. [1979] "A Framework for a Portable Natural Language Interface to Large Data Bases," Technical Note 197, Artificial Intelligence Center, SRI International, Menlo Park, California (October, 1979).

Landsbergen, S. P. J. [1976] "Syntax and Formal Semantics of English in PHLIQA1," in COLING 76, Preprints of the 6th International Conference on Computational Linguistics, No. 21, Ottawa, Ontario, Canada (June 28-July 2, 1976).

Moore, R. C. [1981] "Problems in Logical Form," in Proceedings of the 19th Annual Meeting of the Association for Computational Linguistics, pp. 117-124, The Association for Computational Linguistics, SRI International, Menlo Park, California (July 1981).

Robinson, Ann E. [1980] "Interpreting Natural-Language Utterances in Dialogs About Tasks," Technical Note 210, Artificial Intelligence Center, SRI International, Menlo Park, California (March 1980).

Robinson, J. J. [1982] "DIAGRAM: A Grammar for Dialogues," Communications of the ACM, Vol. 25, No. 1, pp. 27-47 (January 1982).

Rosenschein, S. J. and Shieber, S. M. [To appear] "Translating English into Logical Form," in Proceedings of 20th Annual Meeting of Association for Computational Linguistics Conference, Toronto, Canada (June 1982).

Tienari, M. [1980] "On the Definition of an Attribute Grammar," in Neil D. Jones, ed., Semantic-Directed Compiler Generation. Proceedings of a Workshop, Aarhus, Denmark (January, 1980). G. Goos and J. Hartmanis, eds., Lecture Notes in Computer Science Vol. 94, pp. 408-414, (Springer-Verlag, Berlin-Heidelberg-New York, 1980).

Walker, D. E. (ed.) [1978] Understanding Spoken Language (Elsevier North-Holland, New York, New York, 1978).