AN OBJECT-ORIENTED PARSER FOR TEXT UNDERSTANDING

Brian Phillips

Texas Instruments Inc.
P.O. Box 226015, MS 238
Dallas, TX 75266, USA

ABSTRACT

The parser is part of a text understanding system in which structural ambiguity is a major problem. All components of the system use a message-passing control structure. A general advantage of this form of control is that it allows the flexible integration of diverse knowledge sources. The parser transmits sub-sentential constituents for semantic interpretation. A pseudo-parallel version of the left-corner parsing algorithm with top-down filtering is used. As blind transmission would send spurious constituents, a delay mechanism is used to queue constituents until all alternative analyses of a segment have been completed.

I INTRODUCTION

The parser is a component in a system that constructs a knowledge base from textual input. Only the parser is discussed here; Phillips and Hendler (1982) gives a plan of the whole system.

Our corpus is descriptions of Texas Instruments* patents; a one-sentence fragment is shown in Figure 1.

> A modulator comprising two transistors each having collector, emitter and base electrodes, means for applying a direct voltage across said emitter electrodes, a center-tapped source of alternating signals connected between said base electrodes, said collector electrodes being connected together and to the center tap of said source.

Figure 1: A patent description

The phrase the "means for applying ..." can be attached either to "a modulator comprising ..." or to "two transistors each having ..." The explosive effect of structural ambiguity makes it essential that such ambiguities be resolved quickly. Text understanding systems that do not integrate the use of linguistic and domain knowledge have limited potential to handle such problems.

The parser has an ability to dispatch various sub-sentential phrases for semantic interpretation as they are formed, thus permitting the elimination of meaningless parse paths. Also note that the example "sentence" has the form of an NP. The nature of the system permits representation to be constructed in parallel with syntactic analysis; thus meaning is extractable in spite of grammatical incompleteness. Further, conceptual predictions can be used to guide parsing.

II DESIGN FEATURES

A. Accessing semantics

Integrating syntax and semantics may be achieved by use of a common data structure as in Semantic Grammars (Hendrix, 1977). However, retaining the autonomy of components enables work in theoretical linguistics and knowledge representation, for example, to be directly utilized. Cascaded ATNs (Woods, 1981) allow semantic critiquing of syntactic constructs. We would like to have a bidirectional flow of information: Halliday and Hasan (1975) claim that cohesion is greater within paragraphs than across them. Consequently we envisage a system that is more predictive when within a paragraph and more bottom-up near paragraph boundaries. Semantically driven systems (Schank, 1975) are less likely to perform well when trying to understand structurally complex texts.

Our system integrates the use of descriptively autonomous components with an object-oriented, message-passing control structure (Hewitt, 1976).

B. Forming constituents

Only noun phrases, verb groups, and clauses are transmitted by the parser. The first build entity concepts in knowledge, the second retrieve the case structures of event concepts, and the last are equivalents of completed case frames.

A parser that transmits phrases as they are formed will send many spurious constituents. For example, in the text of Figure 1, there are "two", "emitter and base electrodes", etc. Lookahead is used in Parsifal (Marcus, 1979) for deterministic parsing, but many of the problems found in the patents are not addressed.

Dispatching is controlled by counting the number of ongoing alternative analyses and delaying action until they have all terminated. Then those constituents that are on still-viable parse paths are transmitted.

C. Depth- or breadth-first?

The system has to know simultaneously the state of all alternative parses that start in any word position. A depth-first system cannot know if yet-to-be-tried paths will yield another analysis but a breadth-first system can be cogniscent of concurrent alternatives. A (pseudo-) parallel control structure is thus the appropriate environment for the delay mechanism.

III THE OBJECT-ORIENTED PARSER

A. The grammar: "Local Grammar"

The grammar has a context-free phrase structure component augmented with constraints and percolation rules for passing feature values to the parent category (Saenz, 1982), see Figure 2.

```
PHRASE STRUCTURE:
  np = (det) adj* (nmod) n

BLOCKING:
  (when (equal (dtype 1) article)
        (not-both (gap 3) (gap 4))
  (when (equal (number 4) sing)
        (exist 1)))

PERCOLATION:
  (number (number 4))
  (gap (gap 4))
```

Figure 2: A Local Grammar rule

The grammar allows null realizations for nouns and noun phrases(*). It also includes the clause rule:

clause = (comp) np verb-group

without any explicit objects. The dictionary entry for a verb form includes its type: transitive, ditransitive, etc. This feature is percolated up to the verb-group and the rule completed appropriately.

Dictionary entries also contain the information used to translate syntactic relations onto meaning relations in the spirit of lexical-functional grammar (Bresnan and Kaplan, 1982).

B. The parsing algorithm

The algorithm is based on the left-corner algorithm with a reachability matrix (Griffiths and Petrick, 1965), which Slocum (1982) has shown to be efficient for long sentences, as are found in the patents. It is modified to a pseudo-parallel format for reasons mentioned earlier.

The system is implemented using the "flavor" system in ZLISP (Weinreb and Moon, 1981). A "constituent" flavor creates an object that is associated with a rule of the grammar. Each constituent attempts to instantiate its rule and has methods for doing this.

```
NAME:          CONST-21
CATEGORY:      CLAUSE
GOALS-LIST:    ((CLAUSE . NONE))
PART-PARSE:    ((2 CONST-2
                  ((NP (NUMBER SING))))
                (3 CONST-17
                  ((VERB-GROUP (NUMBER SING)
                      (VTYPE TRANS) ... ))))
RULE-TAIL:     ((NP))
BLOCKERS:      ((EXIST 4))
PERCOLATERS:   ((GAP (GAP 2)))

ALTERNATE:     NIL
CONTINUE:      (CONST-27)

ACTIVE:        (CONST-23)
COUNT:         2
QUEUE:         ((CONST-26 . CONST-27))
LEVELLERS:     (CONST-26 CONST-25 ... )
```

Figure 3: A constituent object

Figure 3 shows the principal variables of a constituent. The first group describe the rule-state and the GOALS-LIST variable shows the constituent towards which the constituent is growing and is used by the reachability matrix. The parser merges common parts of parse paths; thus there may be multiple categories on GOALS-LIST. The rest of the variables will be explained later.

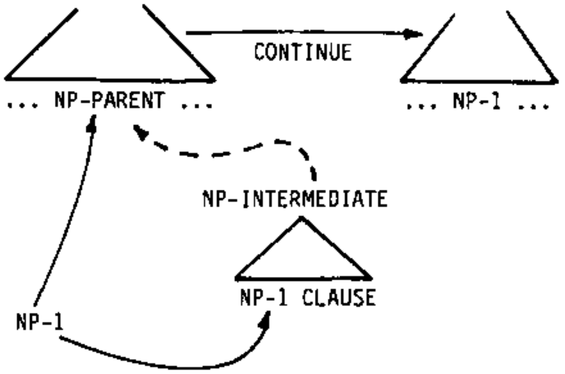

Figure 4: Upward attachment

Active constituent., from a list maintained by a scheduler, are sent the input word in order that they can advance their rule-state. If the next rule-segment is a terminal category an immediate match can be attempted. For non-terminal segments, subparse constituents(**) are initiated. When an object successfully instantiates its grammar rule it will be linked to

* If the full range of ellipsis were incorporated into the grammar, it may be so unconstrained as to approach worthlessness. This indicates an area where prediction could be useful in guiding a less permissive grammar.

** The filtered left-corner algorithm selects rules that are "reachable" from the parent category and have the category-to-be-matched as the first symbol of their RHS.

higher level constituents. If the completed constituent has the category of the goal slot, it can fill that slot, NP-1 into NP-PARENT in Figure 4. Otherwise, following the left corner algorithm, it will create intermediate constituents. If there are recursive rules, then both actions take place, as is shown in Figure 4, with NP-INTERMEDIATE being the intermediate constituent.

When a subparse fills the parent slot, the parent's constituent makes a copy of itself, with the subparse inserted, to continue the analysis. A version of the constituent has to be left to capture other subparses, e.g. NP-INTERMEDIATE in Figure 4, that finish later. The subparse constituent and its copied context are recorded in QUEUE (Figure 3) of the original parent. The pristine constituent retains pointers to its copies in the CONTINUE variable (Figures 3, 4).

Optional rule elements will cause constituents to be set up to follow the alternatives. The ALTERNATE variable (see Figure 3) points to a constituent that splits off in this manner.

ACTIVE and COUNT (Figure 3) record subparses. COUNT is the number of structures that could fill the parent slot. This number will grow when optional elements cause subparses to split. It can also grow if intermediate constituents are created when an subordinate constituent is completed. The count will diminish when a subparse fails, or is attached to a parent slot. When COUNT becomes 0 the QUEUEd constituents are examined and only those on still valid parse paths are transmitted.

If self-embedding occurs, we do not want to wait until the top-level constituent is completed, e.g. appraisal of NP-1 should not await the completion of NP-INTERMEDIATE, Figure 4. This implies that the delay mechanisms should be sensitive to levels of self-embedding. The LEVELLERS variables (Figure 3) is used for this. The initial level contains only items initiated directly from a non-terminal category. Counting is only affected by constituents on the current level; self-embedded constituents are marked as being on the next level. When the first-level objects have all been accounted for, the count is reset with the next level's constituents and the process of attaching and elimination is iterated.

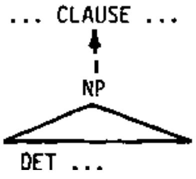

Figure 5: The role of ephemeral parents

COUNT only records structures that will fill the non-terminal slot in the parent. However, within those structures transmittable constituents, the NP in Figure 5, may appear. For them ephemeral parents having the required counting capabilities, are used.

Ambiguous attachment of constituents cannot immediately be resolved as a spurious constituents could be involved; it will be treated when the attachee is passed for interpretation.

IV CONCLUSION

The features of the parser should allow it to perform efficiently in the text understanding environment. However, following the findings of Slocum (1982), we need to be wary that implementation overhead does not outweigh theoretical advantages.

REFERENCES

Bresnan, J., and Kaplan, R. Introduction: Grammars as mental representations of language. In J. Bresnan (Ed.), The Mental Representation of Grammatical Relations. Cambridge, MA: MIT Press, 1982":

Griffiths, T.V., and Petrick, S.R. On the relative efficiencies of context-free grammar recognizers. CACM 8:5 (1965) 289-300.

Halliday, M., and Hasan, R. Cohesion in English. London: Longmans, 1975.

Hendrix, G.G. Human engineering for applied natural language processing. In Proc. IJCAI-77. Cambridge, MA, August, 1977, pp. 183-191.

Hewitt, C. "Viewing control structures as patterns of passing messages," AI Memo 410, MIT AI Laboratory, Cambridge, December 1976.

Marcus, M.P. A Theory of Syntactic Recognition. Cambridge: MIT Press, 1979.

Phillips, B., and Hendler, J.A. A message-passing control structure for text understanding. In Proc. COLING-82. Prague, Czechoslovakia, July, 1982 pp. 307-312.

Saenz, R. "Local grammar," Unpublished paper, Department of Linguistics, University of Massachusetts, Amherst, June 1982.

Schank, R.C. Conceptual Information Processing. New York: American Elsevier, 1975.

Slocum, J. "A practical comparison of parsing strategies for machine translation and other natural language purposes," Technical report NL-41, Department of Computer Science, University of Texas, Austin, August 1981.

Weinreb, D., and Moon, D. Lisp Machine Manual. Cambridge: MIT AI Laboratory, 1981

Woods, W.A. Cascaded ATN grammars. AJCL 6:1 (1980) 1-12.