If The Parser Fails*

Ralph M. Weischedel
University of Delaware
and
John E. Black**
W. L. Gore & Associates, Inc.

The unforgiving nature of natural language components when someone uses an unexpected input has recently been a concern of several projects. For instance, Carbonell (1979) discusses inferring the meaning of new words. Hendrix, et.al. (1978) describe a system that provides a means for naive users to define personalized paraphrases and that lists the items expected next at a point where the parser blocks. Weischedel, et.al. (1978) show how to relax both syntactic and semantic constraints such that some classes of ungrammatical or semantically inappropriate input are understood. Kwasny and Sondheimer (1979) present techniques for understanding several classes of syntactically ill-formed input. Codd, et.al. (1978) and Lebowitz (1979) present alternatives to top-down, left-to-right parsers as a means of dealing with some of these problems.

This paper presents heuristics for responding to inputs that cannot be parsed even using the techniques referenced in the last paragraph for relaxing syntactic and semantic constraints. The paper concentrates on the results of an experiment testing our heuristics.

We assume only that the parser is written in the ATN formalism. In this method, the parser writer must assign a sequence of condition-action pairs for each state of the ATN. If no parse can be found, the condition-action pairs of the last state of the path that progressed furthest through the input string are used to generate a message about the nature of the problem, the interpretation being followed, and what was expected next. The conditions may refer to any ATN register, the input string, or any computation upon them (even semantic ones). The actions can include any computation (even restarting the parse after altering the unparsed portion) and can generate any responses to the user.

These heuristics were tested on a grammar which uses only syntactic information. We constructed test data such that one sentence would block at each of the 39 states of the ATN where blockage could occur. In only 3 of the 39 cases did the parser continue beyond the point that was the true source of the parse failing.

From the tests, it was clear that the heuristics frequently pinpointed the exact cause of the block. However, the response did not always convey that precision to the user due to the technical nature of the grammatical cause of the blockage. Even though the heuristics correctly selected one state in the overwhelming majority of cases, frequently there were several possible causes for blocking at a given state.

Another aspect of our analysis was the computational and developmental costs for adding these heuristics to a parser. Clearly, only a small fraction of the parsing time and memory usage is needed to record the longest partial parse and generate messages for the last state on it. Significant effort is required of the grammar writer to devise the condition-action pairs. However, such analysis of the grammar certainly adds to the programmer's understanding of the grammar, and the condition-action pairs provide significant documentation of the grammar. Only one page of program code and nine pages of constant character strings for use in messages were added.

From the experiment we conclude the following:

1. The heuristics are powerful for small natural language front ends to an application domain.

2. The heuristics should also be quite effective in a compiler, where parsing is far more deterministic.

3. The heuristics will be more effective in a semantic grammar or in a parser which frequently interacts with a semantic component to guide it.

We will be adding condition-action pairs to the states of the RUS parser (Bobrow, 1978) and will add relaxation techniques for both syntactic and semantic constraints as described in Weischedel, et.al. (1978) and Kwasny and Sondheimer (1979). The purpose is to test the effectiveness of paraphrasing partial semantic interpretations as a means of explaining the interpretation being followed. Furthermore, Bobrow (1978) indicates that semantic guidance makes the RUS parser significancly more deterministic; we wish to test the effect of this on the ability of our heuristics to pinpoint the nature of a block.

References

Bobrow, Robert S., "The RUS System," in Research in Natural Language Understanding, B. L. Webber and R. Bobrow (eds.), BBN Report No. 3878, Bolt Beranek and Newman, Inc., Cambridge, MA, 1978.

Carbonell, Jaime G., "Toward a Self-Extending Parser," in Proceedings of the 17th Annual Meeting of the Association for Computational Linguistics, San Diego, August, 1979, 3-7.

Codd, E. F., R. S. Arnold, J-M. Cadiou, C. L. Chang and N. Roussopoulis, "RENDEZVOUS Version 1: An Experimental-Language Query Formulation System for Casual Users of Relational Data Bases," IBM Research Report RJ 2144, San Jose, CA, January, 1978.

Hendrix, Gary G., Earl D. Sacerdoti, Daniel Sagalowicz, and Jonathan Slocum, "Developing a Natural Language Interface to Complex Data," ACM Transactions on Database Systems, 3, 2, (1978), 105-147.

Kwasny, Stan C. and Norman K. Sondheimer, "Ungrammaticality and Extragrammaticality in Natural Language Understanding Systems," in Proceedings of the 17th Annual Meeting of the Association for Computational Linguistics, San Diego, August, 1979, 19-23.

Lebowitz, Michael, "Reading with a Purpose," in Proceedings of the 17th Annual Meeting of the Association for Computational Linguistics, San Diego, August, 1979, 59-63.

Weischedel, Ralph M., Wilfried M. Voge, and Mark James, "An Artificial Intelligence Approach to Language Instruction," Artificial Intelligence, 10, (1978), 225-240.

*This work was supported by the University of Delaware Research Foundation, Inc.

**This work was performed while John Black was with the Dept. of Computer & Information Sciences, University of Delaware.