

A RULE-BASED APPROACH TO ILL-FORMED INPUT

Norman K. Sondheimer

Ralph M. Weischedel

Sperry Univac
Blue Bell, PA, USA

University of Delaware
Newark, DE, USA

SUMMARY

Though natural language understanding systems have improved markedly in recent years, they have only begun to consider a major problem of truly natural input: ill-formedness. Quite often natural language input is ill-formed in the sense of being misspelled, ungrammatical, or not entirely meaningful. A requirement for any successful natural language interface must be that the system either intelligently guesses at a user's intent, requests direct clarification, or at the very least, accurately identifies the ill-formedness. This paper presents a proposal for the proper treatment of ill-formed input. Our conjecture is that ill-formedness should be treated as rule-based. Violation of the rules of normal processing should be used to signal ill-formedness. Meta-rules modifying the rules of normal processing should be used for error identification and recovery. These meta-rules correspond to types of errors. Evidence for this conjecture is presented as well as some open questions.

1. Introduction

Natural Language interfaces have improved markedly in recent years and have even begun to enter the commercial marketplace, e.g., the ROBOT system of Artificial Intelligence Corporation (Harris, 1978). These systems promise to make major improvements in the ease-of-use of data base management and other computer systems. However, they have only begun to consider the problems of truly natural input. The emphasis has been, and continues to be, on the understanding of well-formed inputs. True natural language input is often ill-formed in the absolute sense of being filled with misspellings, mistypings, mispunctuations, tense and number errors, word order problems, run-on sentences, sentence fragments, extraneous forms, meaningless sentences, impossible requests, etc. In addition, natural input is ill-formed in the relative sense of containing requests that are beyond the limits of either the computer system or the natural language interface. The frequent occurrence of these phenomena has been pointed out by both friend and foe of natural language interfaces, see for example, Malhotra (1975),

Montgomery (1972), and Shneiderman (1978).

Most systems deal with a few of these types of ill-formedness. Experience (Harris, 1977b and Hendrix, et al., 1978) has shown that users can adapt to the limitations of the system's well-formed, anticipated input. Yet, we feel that presuming on such user adaptation eliminates one of the most powerful motivations for English input: namely, enabling infrequent users to access their data without an intermediary person and without extensive practice. Even for the person who frequently uses such a system, if it cannot explain why it misunderstands an input, the system will be exasperating at times.

Therefore, we totally agree with Wilks (1976) in his statement that "Understanding requires, at the very least, ... some attempt to interpret, rather than merely reject, what seems to be ill-formed utterances." A requirement for any natural language interface must be that, when faced with ill-formed input, the system either intelligently guesses at a user's intent, requests direct clarification, or at the very least, accurately identifies, the ill-formedness.

Researchers including ourselves have worked on various aspects of ill-formedness. Out of our work, and that of others, we have produced a conjecture on the treatment of ill-formed input to natural language interfaces. That conjecture is in essence that ill-formedness should be treated as "rule-based". First, natural language interfaces should process all input as presumably well-formed until the rules of normal processing are violated. At that point, error handling procedures based on meta-rules relating ill-formed input to well-formed structures through the modification of the violated normal rules should be employed. These meta-rules correspond to types of errors.

The rest of the paper argues for this rule-based approach. Section 2 characterizes both the types of ill-formed input, and the types of possible approaches to them. Section 3 explains our proposal. Section 4 motivates the proposal through analysis of its effect on the

development and operation of natural language interfaces, the use of evidence from other disciplines that consider ill-formedness in natural and artificial languages, and, most importantly, evidence from work on natural language understanding systems. Section 5 discusses some open problems in light of the proposal.

2. Problem and Solution Spaces

This section introduces the problem of interpreting ill-formed input. First, an analysis is given of the types of ill-formed input. Then we present the range of approaches for allowing for such input. In the next section we use this set to isolate our own conjecture.

2.1 A view of Ill-formed Input

Ill-formedness can be divided into two sets. The first defines what we call absolute ill-formedness. We will call an utterance absolutely ill-formed if it cannot convey the speaker's intended message unless the typical listener gives it an abnormal interpretation. The definition unfortunately appeals to subjective evaluations; these are known to differ widely (Ross, 1979). But it seems to include the majority of typical cases and exclude the majority of types of good English sentences.

The second set defines relative ill-formedness. This is ill-formedness with respect to the normal processing rules of the formal computing system including the natural language interface and the underlying application system. The set of ill-formed inputs for an interface will be defined as the union of these two sets for that interface.

The set of ill-formed input captured by this definition can also be seen through the four typical phases of interpretation in natural language interfaces: lexical, syntactic, semantic and pragmatic processing. In lexical processing, (processing individual words), absolute ill-formedness can come from misspelling and mistyping; relative ill-formedness can arise from unexpected words. In syntactic processing, absolute ill-formedness is seen in subject-verb agreement, word order errors, fragmentary queries, run-on sentences, etc; relative ill-formedness is seen in grammatical combinations of words that exceed the interface's grammar.

Semantic processing can be defined as the interpretation of the input in isolation. Knowledge of the task domain can be applied, but the context of input with respect to previous interactions and the state of the underlying computing system are only considered in pragmatic processing. Absolute ill-formedness in semantics includes omitting needed information and violation of selectional restrictions. Absolute ill-formedness in pragmatics includes breaking the rules of conversation, as when answering a

question with a question, having presuppositions of the questioner fail, and failing to make clear the reference of a pronoun. Relative ill-formedness in both cases is usually a matter of "overshoot", requesting capabilities or information not covered by the system in its current state.

2.2 Possible Approaches to Interpretation

A large set of approaches deal with ill-formed input. The set's size is seen through the choices available in three of the major phases in a system's life and use: systems development, error identification and error recovery. In this section, we will go through the basic options in these phases, giving a quick survey of systems exhibiting the options.

2.2.1 Development Phase

The basic decision during a natural language interface's development is the degree of differentiation between ill-formed and well-formed input processing. Where no difference is maintained either ill-formed inputs can be included in the regular components, (e.g., putting unallowable words in dictionaries to detect some queries that cannot be answered (Codd et al., 1978) and sentence fragments in the grammar (Burton and Brown, 1977)), or the components can be written to ignore many well-formedness constraints (Shapiro and Kwasny, 1975; Waltz, 1978; Lebowitz, 1979), when the task does not depend on them.

When differences are recognized, a much larger range of choices is possible. For example, new classes of rules could be added to the existing components (Kwasny and Sondheimer, 1979). New components could be added to the system as well, using the same form as normal rules, (Harris, 1977).

2.2.2 Error Identification Phase

Discovering how an input is ill-formed will be called error identification, even though not all ill-formedness is an error. During this phase, the type of computation undertaken can vary. If ill-formed input is not distinguished from well-formed, then no error identification is done.

When it is done, one approach is to make no effort to analyze ill-formed portions, e.g., unidentified words are often skipped and processing resumed (Burton and Brown, 1977; Codd et al., 1978; Kwasny and Sondheimer, 1979). A second strategy is to totally ignore the source of the failure in interpretation and attempt error identification (and also recovery) by completely independent computations on input, e.g., the separate grammar for ill-formed sentences in ROBOT (Harris, 1977). Using a third approach, many systems attempt to clarify the nature of

the ill-formedness by directly considering the source of the failure (Weischedel et al., 1978; Weischedel and Black, 1980; Kwasny and Sondheimer, 1979).

2.2.3 Error Recovery Phase

The error recovery phase covers all computation following identification of an error until normal processing is resumed. One choice is to communicate to the user the error description and have him reenter a corrected request (Weischedel et al., 1978 and Codd et al. 1978).

Another basic choice is automatic recovery. One can design a process not relying on the rules for well-formed input; for instance, Harris (1977) uses a grammar for ill-formed input and Waltz (1978) has a specialist routine for completing fragmentary sentences. Alternatively one can relate the ill-formedness to normal processing; Hendrix, et al. (1978), Biermann and Ballard (1978), Weischedel (1977), and Kwasny and Sondheimer (1979) have taken this approach for some classes of ill-formedness.

It is difficult to give an independent picture of error recovery. Depending on the choices in system development and error identification, a system can have anything from no idea as to the source of the problem, to moderately informative insights such as knowing it does not recognize some words, to precise knowledge of the problem such as being aware of the lack of subject-verb number agreement. Further, the error itself may dictate the limits of recoverability. Some errors may be too crucial to allow recovery. For example, when a presupposition of the user's input is not true, no recovery seems reasonable other than stating that the particular presupposition is not true.

3. A Rule-Based Approach to Ill-formed Input

Out of the options available for preparing for and responding to ill-formed input, we propose one, in particular. This section begins with a short statement of our proposal and continues by clarifying and motivating it. Evidence for it from other work is then presented.

3.1 Statement

In essence, we believe that ill-formedness should be treated as rule-based. We see two kinds of rules: first, rules used in normal processing and second, meta-rules which are only employed to interpret ill-formed input. With respect to the first, we feel that their violation should be used to detect ill-formed input. With respect to the second, we feel that they should be meta-rules applying to the rules of the first sort in order to relate the structure of ill-formed input to that of well-formed structures. This would be done by showing how the well-formedness rules could be modified to

accept the ill-formed input with as complete a structure as possible. They will indicate a general type of error user's make.

In terms of the three phases discussed in the last section, acceptance of our conjecture would lead to separate development of components for handling well-formed and ill-formed inputs. Considering syntactic processing as an example, a normative grammar would be written to interpret grammatically well-formed sentences. Separately, meta-rules would be developed for grammatically ill-formed sentences.

Error identification would include analysis of failures in normal processing rules using the rules defining ill-formedness. For example, an error identification component would find the cause of a blockage in parsing by considering the failed grammar rules and the meta-rules that show how these normative rules could fail. In light of these, the normative rules could be modified automatically via the meta-rules in order to see if the input could be accepted.

Finally, whenever error recovery was feasible, it would use the ill-formedness rules to guide the modification of the rules of normal processing in order to continue processing the ill-formed input. For example, a failed semantic restriction test can be relaxed by a meta-rule and processing continued. Note that this often introduces uncertainty in that the constraint often carries semantic information, hence complete understanding is not guaranteed by our proposal.

3.2 Example

Consider subject-verb number agreement as in Weischedel (1977). Presumably any natural language interface for well-structured input would have tests to check for this, since it reflects semantic information, e.g., verb number differentiates between the meanings of "Flying planes is dangerous" and "Flying planes are dangerous." However, number agreement errors are known to occur. We would capture this by adding a meta-rule allowing the agreement test to be ignored. This would, of course, be done at the cost of not identifying the intended sense. According to our proposal, an input, such as "The boy run fast", would be treated as a potentially well-formed until the grammar failed to interpret it. When the example fails to parse, we would attempt identification of the input based on the failure of the agreement test and the meta-rule. Then recovery would be attempted by removing the test and proceeding without knowing whether singular or plural was intended.

Of course a system could at this point request user supplied clarification, or it could decide to abort the processing. However, our goal is to provide the ability to automatically

interpret as much as possible.

3.3 Assumptions

Underlying our belief in the viability of this approach to ill-formedness are some assumptions that limit the problem. Most important is the assumption of a cooperative user. Observation of cooperative users has shown that they tend to keep their requests linguistically simple and tailored to what they feel are the system's limits (Woods, 1973; Malhotra, 1975; Damerau, 1979). At the same time, users have been shown to be able to communicate effectively through limited machine interfaces (Kelly and Chapanis, 1977). This allows us to ignore many of the more difficult ill-formedness phenomena. An uncooperative user could "break" any system. For example, a user is reported to have asked a well-known system "What the h-ll is going on here?". No system should be expected to handle such input.

Overshoot is a related phenomenon. Overshoot often arises with users unfamiliar with the capabilities of the computer system underlying the natural language interface (Woods, 1973; Shneiderman, 1978; Tennant, 1979). In order to allow for any overshoot we must be able to depend on our understanding of the user's knowledge. We therefore assume that the user has at least basic familiarity with the purpose and power of the underlying system.

Finally, we assume that the natural language interface for normal sentences is well-structured in the sense of handling like sentences similarly and unlike sentences dissimilarly, and in the sense of having a decomposition of processing into explainable and defensible phases. In programming languages, it is the case that grammars and parsers can be written to identify and recover from errors (Aho and Johnson, 1974). This ought to be the case with natural language interfaces. We are willing to defend our conjecture independent of any one structuring as long as the interface for well-formed input we are augmenting is built on consistent, explainable lines.

4. Supporting evidence

We will now consider evidence supporting our proposal.

4.1 Pragmatic Motivation

There are a number of reasons to prefer this solution, independent of the empirical evidence that we will present shortly. Basically, this approach will ease systems development and processing. This is true first because of the ability to design the normative processing system independent of the error identification and recovery methods. Second, not invoking ill-formedness processing until normal processing

fails avoids unnecessary runtime costs for well-formed sentences, which are the normal type of input. Third, describing ill-formedness through meta-rules that relate to normative rules will avoid duplication of aspects of normative processing and allow general statements covering classes of ill-formedness.

4.2 External Supporting Evidence

There is support for our proposal from many other areas where ill-formedness in natural or artificial languages is considered. Most relevant are the efforts of linguists. When they have considered ill-formedness it has been common for them to propose the type of meta-rules we propose. For example, Chomsky (1964) relates failures to abide by different aspects of his grammar model to different classes of ill-formedness through relaxation of well-formedness constraints. Linguists also try to spot patterns in utterances containing errors in order to motivate rules for normal processing (Fronkin, 1973).

A pattern of rule-based treatment of ill-formedness can be seen elsewhere. In information retrieval, index terms are processed as if they were correctly presented, until failure starts recovery methods based on rules which change the conditions for acceptance (Damerau, 1964). In programming languages, similar processing is seen with typographic errors and with syntactic problems such as incorrect numbers of parentheses (Teitelman, 1969; Morgan, 1970; Aho and Johnson, 1974). Trapping based on normative constraints and error recovery (at least in notifying the user) is seen in the maintenance of data base integrity (Wilson and Salazar, 1979). Finally, speech understanding systems, whose ill-formedness problems are related to noisy signals, often work from an initial assumption that a clear interpretation can be found for the input. When this fails, they take what they have found and attempt to recover by applying normative rules in a less rigorous way in order to identify the ill-formed segments (Bates, 1976; Miller, 1974).

4.3 Support from Natural Language Interface Efforts

To our knowledge, our general approach to ill-formedness has not been propounded elsewhere. However, work fitting within the paradigm has been applied to a number of isolated ill-formedness problems. In addition, one important technique which has been employed for ill-formedness appears to be modifiable so as to fit within our approach. The success of these efforts stands as support for our approach. In this section, they will be briefly surveyed.

4.3.1 Lexical

A lexicon may be thought of as a computa-

tional model of dictionary information. According to our approach, processing of lexical ill-formedness would be developed separately from the preparation of the processing of normal lexical entries (i.e. dictionary entries). Once the rules for processing well-formed inputs fail to recognize a lexical entry, error identification would begin based on the failed rules and rules which showed how lexical entries could be ill-formed. At the end of this identification phase, a guess or guesses as to the identity of a lexical entry would be available for the system to attempt recovery. This paradigm for processing can be seen in a number of systems in attempts to treat both absolute and relative lexical ill-formedness.

The LIFER system is prepared to deal with misspelled and mistyped words through a method fitting within our model (Hendrix et al., 1978). The developer of a question-answering system using LIFER prepares only a dictionary of well-formed words. If a sentence contains a word that is not in the dictionary, the LIFER parser will fail and start error identification. LIFER first chooses as the putative failed rule the one associated with the partial interpretation that has proceeded furthest. From that rule, LIFER identifies the part of speech the word should belong to and applies a mistyping and misspelling algorithm based on such meta-rules as "expect letters to be duplicated" or "expect letters to be reversed" to modify the normal dictionary look up rules and to match the ill-formed input to all well-formed members of the desired part of speech. If one is found, normal processing resumes.

Examples related to our approach can also be seen in methods that deal with relative ill-formedness. For example, Granger's (1977) FOUL-UP program proceeds through input until it finds an unknown word. Based on its expectations for the input derived from parsing and its model of semantic content, it attempts recovery by assigning a partial interpretation to the input.

Somewhat similar processing can be seen in dealing with typographic errors (Biermann and Ballard, 1978), learning new names (Codd et al., 1978), and learning new words (Carbonell, 1979, and Miller, 1975).

4.3.2 Syntax

With syntactic processing, our paradigm calls for separate development of a grammar for well-formedness, identification of errors based on the failure to parse, and error recovery based on manipulation of the grammar. This is most clearly seen in our own work. Weischedel (1977) was the first to suggest several different techniques for dealing with syntactically ill-formed input. One technique allows grammar writers to insert rules to enable selective re-

laxation of restrictions in the grammar so that certain ungrammatical sentences may be assigned as much structure as possible. For example, his method would allow the number-agreement test to be relaxed as was discussed. Weischedel's method was tested in a natural language understanding system for intelligent tutoring of students learning a foreign language (Weischedel et al., 1978). A second technique suggested by Weischedel (1977) is the assignment of meanings to the states of an ATN grammar. These assignments were used to guide error identification for the end-user when interpretation of a sentence blocked at a state. The assignments could be quite general including operational procedures and could attempt complex deductions of the source of the error. Weischedel and Black (1980) report the results of testing the method on a parser for English.

Kwasny and Sondheimer (1979) extend Weischedel's first method to allow for successively less stringent constraints. In addition, they propose a relaxation method using hierarchical structuring of syntactic categories, based on a suggestion in Chomsky (1964). If the normal rules fail to accept a sentence and the failed rule is looking for a part of speech which is a member of a hierarchy, then relaxation proceeds by substituting the next more general class in the hierarchy for the unsatisfied part of speech.

Perhaps the most powerful technique of treating syntactic ill-formedness, as Hayes and Reddy (1979) and Hayes and Mouradian (1980) point out, is including patterns for ill-formed input. Kwasny and Sondheimer (1979) generalize this technique by allowing even more dramatic relaxation of the grammar through patterns that allow the input to be matched against the grammar in a relaxed way, either by skipping words in the input, or by skipping the application of rules. This is most useful for assigning structure to sentence fragments. Importantly, it also applies to many types of conjunction including the problematic case of gapping cases. This technique differs from the paradigm suggested here because of its method of error identification and recovery. When an input is not recognized by the grammar, processing switches to an entirely separate set of arcs in an ATN grammar, essentially another grammar, which are used to assign structure to the ill-formed input. However, experience with the method suggests that the arcs used in this separate grammar could in general be found in the normative grammar. If this is always the case, then the separate grammar could be eliminated. Also, error identification could proceed by considering the failure of the normative rules; error recovery could proceed by relaxing the conditions on the application of the rules to the input string.

4.3.3 Semantics and Pragmatics

Similar kinds of relaxation efforts can be seen in semantic processing. One feature of the preference semantics system of Wilks (1975) is the ability to relax certain semantic constraints. With respect to error identification, Heidorn (1972) dealt with incomplete semantic entities by requesting users to supply missing information based on failures to translate from the internal semantic structures to external computer programs. A somewhat similar process is seen in work by Chang (1978) on the RENDEZ-VOUS system where failure to parse a query leads to a request for clarification from the user.

With respect to pragmatic errors, Weischedel (1977) introduced a technique which uses presupposition to find certain incorrect uses of words. Joshi and Weischedel (1977) and Weischedel (1979) show that since presuppositions can be computed by a parser and its lexicon they are a class of assumptions inherent in the user input; therefore they can be checked for discrepancies with the system's world knowledge. This work was used and extended by Kaplan (1979) in error identification and recovery in those situations where a user's database query would normally yield only an empty set, i.e. an answer of none. Janas (1979) applied similar techniques to assist the user in the same situations.

Many of these techniques can be applied to problems of relative ill-formedness. For example, techniques that are being applied in the development of JETS specifically to capture relatively ill-formed sentences will fit within our paradigm (Finin et al., 1979).

We find the number of techniques that fit within the model we suggest encouraging.

5. Conclusion

Our hypothesis is that both absolute and relative ill-formedness should be treated as rule-based. Rules for well-formed input should be employed first. The detailed way in which the rules of well-formed input are violated signal which meta-rule(s) to use to relate the structure of the ill-formed input to a well-formed one. The meta-rules show how well-formed rules should be modified to interpret ill-formed input as completely as possible.

There are at least three ways to proceed in order to strengthen that hypothesis:

- 1) Reformulating the popular technique of explicitly encoding ill-formed patterns of an ATN within the methodology,

- 2) Developing strategies for additional classes of ill-formedness:

- a) merged thoughts or run-on sen-

tences. An example is "Give me a list of the supplier's list."

- b) wrong word choice. An example is "Computer the standard deviation..." instead of "Compute the standard deviation..." This could not be treated as a spelling error if both "compute" and "computer" were in the lexicon. Hence, it would have to be treated as incorrect word choice.

- c) "expansion" ellipsis. Expansion ellipsis is a kind of fragmentary input no system has processed before. An example would be a response of "On employee name" to a question, "Should the list be printed in alphabetical order?"

- d) violation of semantic constraints. An ill-formed input such as "Have we ordered supplier 34?" violates semantic constraints, since a supplier is not something that can be ordered. We plan to develop techniques that will recognize this semantic violation and hypothesize that "Have we ordered from supplier 34?" was intended.

- 3) Improving ill-formedness handling by parallel processing of lexical, semantic, syntactic, and pragmatic components:

- a) interaction of semantics and syntax for explaining the cause of misunderstanding when no interpretation is possible,

- b) pragmatic and semantic overshoot. An example of overshoot is asking, "What are the average weights of all rock samples?" when the system has no such weights. This could not be detected by dictionary lookup if the data base has weights of atomic elements and has data on rock samples, just not their weights. We intend to develop strategies to detect overshoot and respond appropriately; for the example, an appropriate response is "The system has no weights of rock samples."

We are engaged in a research program involving work on these three problems. All of the rules and meta-rules that we have already developed or are developing will be tested in one of two systems. One is an English front end to data base systems; this research-oriented natural language processor is under development in the Software Research Department of Sperry Univac. The second is a question-answering system (with English input) is being constructed at the University of Delaware.

We believe that sophisticated understanding and response to ill-formed input is the missing ingredient in making natural language interaction truly natural.

REFERENCES

Aho, A. V. and S. C. Johnson, "LR Parsing", Computing Surveys, 6, 2, (1974), 99-124.

Bates, Madeleine, "Syntax in Automatic Speech Understanding", AJCL, Microfiche 45, (1976).

- Biermann, A. W. and B. W. Ballard, Toward Natural Language Computation, CS-1978-11, Durham, North Carolina: Department of Computer Science, Duke University, December, 1978.
- Burton, Richard R. and John Seely Brown, Semantic Grammar: A Technique for Constructing Natural Language Interfaces to Instructional Systems, BBN Report No. 3587, Cambridge: Bolt Beranek and Newman Inc., May 1977.
- 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.
- Chang, C. L., Finding Missing Joins for Incomplete Queries in Relational Data Bases, RJ2145 (29408), San Jose: IBM Research Laboratory, February, 1978.
- Chomsky, Noam, "Degree of Grammaticalness", in The Structure of Language: Readings in the Philosophy of Language, eds., J. A. Fodor and J. J. Katz, Englewood Cliffs, New Jersey: Prentice-Hall, 1964, 384-389.
- Codd, E. F., R. S. Arnold, J-M. Cadiou, C. U. Chang and N. Roussopoulos, RENDEZVOUS Version 1: An Experimental English-Language Query Formulation System for Casual Users of Relational Data Bases, IBM Research Report RJ2144, San Jose, California, January, 1978.
- Damerau, Fred J., "A Technique for Computer Detection and Correction of Spelling Errors", Communications of the ACM, 7, 3, (1964), 171-176.
- Damerau, Fred J., The Transformational Question Answering (TQA) System Operational Statistics - 1978, IBM Research Report RC 7739, Yorktown Heights, New York, June, 1979.
- Finin, Tim, Bradley Goodman, and Harry Tennant, "JETS: Achieving Completeness through Coverage and Closure", in Proceedings of the Sixth International Joint Conference on Artificial Intelligence, Tokyo, August, 1979, 275-281.
- Fromkin, Victoria A., ed., Speech Errors as Linguistic Evidence, Janua Linguarum, Series maior 77, The Hague: Mouton, 1973.
- Granger, Richard H., Jr., "FOUL-UP: A Program that Figures Out Meanings of Words from Context", Proceedings of the 5th International Joint Conference on Artificial Intelligence, Cambridge, Massachusetts, August, 1977, 172-178.
- Harris, Larry R., ROBOT: A High Performance Natural Language Interface for Data Base Query, Technical Report TR 77-1, Dartmouth College, Department of Mathematics, February, 1977.
- Harris, Larry R., "The ROBOT System: Natural Language Processing Applied to Data Base Query", in Proceedings 1978 Annual Conference, Association for Computing Machinery, Washington, D.C., December, 1978, 165-172.
- Harris, L. R., "User Oriented data base query with the ROBOT natural language query system," Int. Journal of Man-Machine Studies, 9, (1977b), pp. 697-713.
- Hayes, P. and R. Reddy, An Anatomy of Graceful Interaction in Spoken and Written Man-Machine Communication, Pittsburgh: Department of Computer Science, Carnegie-Mellon University, August, 1979.
- Hayes, P. and G. Mouradian, "Flexible Parsing", in Proceedings of the 18th Annual Meeting of the Association for Computational Linguistics and Parasession on Topics on Interactive Discourse, Philadelphia, June, 1980, 97-103.
- Heidorn, George E., Natural Language Inputs to a Simulation Programming System, NPS-55HD72101A, Monterey, CA: Naval Postgraduate School, October, 1972.
- 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.
- Janas, Jurgen M., "How to not say 'Nil' - Improving Answers to Failing Queries in Data Base Systems", in Proceedings of the Sixth International Joint Conference on Artificial Intelligence, Tokyo, August, 1979, 429-434.
- Joshi, Aravind K. and Ralph M. Weischedel, "Computation of a Subclass of Inferences: Presupposition and Entailment," American Journal of Computational Linguistics, 4, Microfiche 63, 1977.
- Kaplan, Samuel Jerrold, Cooperative Responses from a Portable Natural Language Data Base Query System, unpublished Ph.D. Dissertation, The University of Pennsylvania, 1979.
- Kelly, M. J., and A. Chapanis, "Limited Vocabulary Natural Language Dialogue", Int. Journal of Man-Machine Studies, 9, (1977), 479-501.
- 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.

- Malhotra, Ashok, Design Criteria for a Knowledge-Based English Language System for Management: An Experimental Analysis, MAC TR 146, Cambridge, Mass: Project MAC, Massachusetts Institute of Technology, February, 1975.
- Miller, Perry L., "A Locally-Organized Parser for Spoken Input", Communications of the ACM, 17, 11, (1974), 621-630.
- Miller, Perry L., "An Adaptive Natural Language System that Listens, Asks, and Learns", in Advance Papers of the Fourth International Joint Conference on Artificial Intelligence, Tbilisi, Georgia, USSR, September 3-8, 1975, 406-413.
- Montgomery, Christine A., "Is Natural Language an Unnatural Query Language?", in Proceedings of the ACM Annual Conference, New York, 1972, 1075.
- Morgan, Howard L., "Spelling Correction in Systems Programs", Communications of the ACM, 13, 2, (1970), 90-94.
- Ross, John Robert, "Where's English" in Individual Differences in Language Ability and Language Behavior, eds., Charles J. Fillmore, Daniel Kempler, and William S-Y. Wang, New York: Academic Press, 1979, 127-163.
- Shapiro, Stuart C. and S. C. Kwasny, "Interactive Consulting Via Natural Language", Communications of the ACM, 18, 8, (1975), 459-462.
- Shneiderman, Ben, "Improving the Human Factors Aspects of Database Interactions", ACM-TODS, 3, 4, (1978), 417-439.
- Teitelman, W., "Toward a Programming Laboratory", in Proceedings: International Joint Conference on Artificial Intelligence, Washington, D.C., May, 1969.
- Tennant, Harry, "Experience with the Evaluation of Natural Language Question Answerers", in Proceedings of the Sixth International Joint Conference on Artificial Intelligence, Tokyo, August, 1979, 874-876.
- Waltz, David L., "An English Language Question Answering System for a Large Relational Database", Communications of the ACM, 21, 7, (1978), 526-539.
- Weischedel, Ralph M., Please Re-Phrase, TR #77/1, Department of Statistics and Computer Science, Newark: University of Delaware, 1977.
- Weischedel, Ralph M., "A New Semantic Computation While Parsing: Presupposition and Entailment", Syntax and Semantics, Volume 11: Presupposition, eds., Choon-Kyu Oh and Dineen, New York: Academic Press, 1979.
- Weischedel, Ralph M. and John Black, "Responding Intelligently to Unparsable Inputs", The American Journal of Computational Linguistics, 6, 2, (1980), 97-109.
- Weischedel, Ralph M., Wilfried M. Voge, and Mark James, "An Artificial Intelligence Approach to Language Instruction", Artificial Intelligence, 10, (1978), 225-240.
- Wilks, Y. A., "A Preferential Pattern-Seeking Semantics for Natural Language Inference", Artificial Intelligence, 6, (1975), 53-74.
- Wilks, Yorick, "Natural Language Understanding Systems Within The AI Paradigm - A Survey and Some Comparisons", AJCL, Microfiche 40, (1976).
- Wilson, Gerald A. and Sandra B. Salazar, "A System for Interactive Error Detection", in Proceedings Fifth International Conference on Very Large Data Bases, Rio de Janeiro, October, 1979, 33-51.
- Woods, William A., "Progress in Natural Language Understanding - An Application to Lunar Geology", in AFIPS Conference Proceedings, 42, (1973), 441-450.