# Evaluating Natural Language Systems: A Sourcebook Approach *

Walter READ
Alex QUILICI
John REEVES
Michael DYER
*Artificial Intelligence Laboratory*
*Computer Science Department*
*University of California, Los Angeles, CA, 90024*

Eva BAKER
*Center for the Study of Evaluation School of Education*
*University of California, Los Angeles, CA, 90024*

**Abstract**

This paper reports progress in development of evaluation methodologies for natural language systems. Without a common classification of the problems in natural language understanding authors have no way to specify clearly what their systems do, potential users have no way to compare different systems and researchers have no way to judge the advantages or disadvantages of different approaches to developing systems.

## Introduction.

Recent years have seen a proliferation of natural language systems. These include both applied systems such as database front-ends, expert system interfaces and on-line help systems and research systems developed to test particular theories of language processing. Each system comes with a set of claims about what types of problems the system can "handle". But what does "handles ellipsis" or "resolves anaphoric reference" actually mean? All and any such cases? Certain types? And what classification of 'types' of ellipsis is the author using? Without a common classification of the problems in natural language understanding authors have no way to specify clearly what their systems do, potential users have no way to compare different systems and researchers have no way to judge the advantages or disadvantages of different approaches to developing systems. While these problems have been noted over the last 10 years (Woods, 1977; Tennant, 1979), research developing specific criteria for evaluation of natural language systems has appeared only recently.

This paper reports progress in development of evaluation methodologies for natural language systems. This work is part of the Artificial Intelligence Measurement System (AIMS) project of the Center for the Study of Evaluation at UCLA. The AIMS project is developing evaluation criteria for expert systems, vision systems and natural language systems.

## Previous Work on Natural Language Evaluation.

Woods (1977) discussed a number of dimensions along which progress in development of natural language systems can be measured. In particular, he considered approaches via a "taxonomy of linguistic phenomena" covered, the convenience and perspicuity of the model used and the time used in processing. As Woods points out, the difficulty of a taxonomic approach is that the taxonomy will always be incomplete. Any particular phenomenon will have many subclasses and it often turns out that the published examples cover only a small part of the problem. A system might claim "handles pronoun reference" but the examples only cover parallel constructions. To make such a taxonomy useful we have to identify as many subclasses as possible. On the positive side, if we can build such a taxonomy, it will allow authors to state clearly just what phenomena they are making claims about. It could serve not only as a description of what has been achieved but as a guide to what still needs to be done.

Woods provides a useful discussion of the difficulties involved in each of these approaches but offers no specific evaluative criteria. He draws attention to the great effort involved in doing evaluation by any of these methods and to the importance of a "detailed case-by-case analysis". Our present work is an implementation and extension of some of these ideas.

Tennant and others (Tennant 1979; Finin, Goodman & Tennant, 1979) make a distinction between conceptual coverage and linguistic coverage of a natural language system and argue that systems have to be measured on each of these dimensions. Conceptual coverage refers to the range of concepts handled by the system and linguistic coverage to the range of language used to discuss the concepts. Tennant suggests a possible experimental separation between conceptual and linguistic coverage.

The distinction these authors make is important and useful, in part for emphasizing the significance of the knowledge base for usability of a natural language system. But the examples that Tennant gives for conceptual completeness — presupposition, reference to discourse objects — seem to be

*This work reported here is part of the Artificial Intelligence Measurement Systems (AIMS) Project, which is supported in part by ONR contract number N00014-86-K-0395.

part of a continuum with topics like ellipsis and anaphora, which are more clearly linguistic. For this reason we don't draw a sharp distinction here. We prefer to look at the broadest possible range of language use. Insofar as recognizing presuppositions depends on the structure of the knowledge base, we note that in the examples. In any case, the question of evaluating the linguistic coverage is still open.

Bara and Guida (1984) give a general overview of issues in evaluation of natural language systems. They emphasize the importance is measuring competence, what the system is capable of doing, over performance, what users actually do with the system. We agree with the emphasis. But how do we measure competence?

Guida and Mauri (1984, 1986) present the most formal and detailed approach to evaluation of natural language systems. They consider a natural language system as a function from inputs to (sets of) outputs. Assuming a measure of error (closeness of the output to the correct output) and a measure of the importance of each input, they evaluate the system by the sum of the errors weighted by the importance of the input. It is assumed that the user can assign these measures in some reasonable way. They give some suggestions for this assignment and work out a small example in detail.

The advantage of a careful, formal analysis is that it focuses attention on the key role of the 'importance' and 'error' measures. In practice, the importance measure has to be given over categories of input. The difficulty is determining what these categories are for a *natural* language. A system that handled five types of ellipsis but not the type the user most needs would be of little use. If the user has a description of the varieties of issues involved, he can define his specific needs and give his own weights to the different categories.

## The Sourcebook Project

The natural language part of the AIMS project has two parts. The first task is to develop methods for describing the coverage of natural language systems. To this end, we are building a database of 'exemplars' of representative problems in natural language understanding, mostly from the computational linguistics literature. Each exemplar includes a piece of text (sentence, dialogue fragment, etc.) a description of the conceptual issue represented, a detailed discussion of the problems in understanding the text and a reference to a more extensive discussion in the literature. (See appendix A for examples.) The Sourcebook consists of a large set of these exemplars and a conceptual taxonomy of the types of issues represented in the database. The exemplars are indexed by source in the literature and by conceptual class of the issue so that the user can readily access the relevant examples. The Sourcebook provides a structured representation of the coverage that can be expected of a natural language system.

The second task of our group is to develop methods for a 'process evaluation' of natural language systems. A process evaluation includes questions of efficiency, perspicuity and conceptual coverage in the sense of Tennant. We are interested in the learnability of a system, in how well the model is documented, in how easily the system can be extended, etc. Generally, we are interested in how the system actually works, including the user interface. The criteria we develop will be applied to representative existing systems. In this paper we focus on the Sourcebook.

## Why a Sourcebook?

In developing evaluative criteria for linguistic coverage we had several goals we wanted to achieve. First, the criteria used should be applicable over the broadest possible range of systems and still provide comparability of the systems. The criteria should be relevant to even very innovative approaches. In fact, the criteria should let the developers of the system describe exactly what is innovative about the system. Second, the criteria should be independent of implementation issues including programming language. A complete analysis of a particular system would of course include implementation details. But it should be possible to describe the coverage independent of such details. Only in this way do we have a basis for claiming an advantage for new implementations or representations. Third, the system shouldn't just rate the system on a pass/fail count. It should outline areas of competence so that implementers and researchers can see where further work is needed within their system or their paradigm. They should be able to say "this approach handles types 1, 2 and 3 of ellipsis but not types 4 and 5 yet" rather than "this approach handles ellipsis". Fourth, the criteria used should be comprehensible to the general user and to researchers outside computational linguistics. For one thing, as Tennant noted, users are less deterred by, say, syntactic limitations than by limitations in the system's concepts, discourse ability, ability to understand the user's goals, etc. We need to present the issues in such a way that the user can make judgments about the importance of different components of the evaluation. This means presenting the issues in terms of the general principles involved and giving concrete examples. This approach also allows us to bring in information from areas like psychology, sociology, law and literary analysis and enables researchers in those areas to contribute to the evaluation. A fifth point is more a negative point. We don't expect to be able to judge any system by one or even a few numbers. Our goal is to find a way to describe and to compare the coverage of systems.

One method often used in computer science to test programs is a test suite and these have been used for natural language evaluation. Test suites have the advantage of simplicity and precision. Hewlett-Packard presented one such suite covering a variety of tests of English syntax at the 1987 Association for Computational Linguistics meeting. But this approach is very limited. Although a parser passed one example of a "Bach-Peters sentence (periphery)", it might fail on another very similar sentence which is conceptually different. (This test suite doesn't measure how well the system *understands* what's going on.) The categories are those derived from a particular syntactic theory, rather than categories that users work with. The test suite tests only a very

limited range of linguistic phenomena and the test is simply pass/fail. And when a sentence fails to pass, it's not always clear why without looking at the implementation. For the reasons mentioned here, we looked for a more generally useful method than test suites.

Rather than start with a particular theory of language, we began with a search of the computational linguistics literature. While no-one would claim that computational linguistics has discovered, let alone solved, every problem in language use, twenty-five years of research has covered a broad range of problems. Looking at language use computationally focuses attention on phenomena that are often neglected in more theoretical analyses. Building systems intended to read real text or interact with real users raises complex problems of interaction of linguistic phenomena. The exemplars are mostly taken from the literature although we have added examples to fill in gaps where we felt the published examples were incomplete. Because many of the published cases involved particular systems, the examples are often discussed in the literature in relation to that system. In the exemplars, we analyze the example in terms of the general issue represented. Then the exemplars are grouped into categories of related problems. This generates the hierarchical classification of the issues. We don't start with an a priori theory for this classification but rather look for patterns in the exemplars. (A summary of the first two layers of the hierarchical classification is in Appendix 2.)

By drawing examples from the full range of the literature, including not only successful examples but unsuccessful ones, the Sourcebook gives a broad view of linguistic phenomena. Although published examples are often about implementations, we have focused on examples that illustrate more general issues. The classification of the examples maps the overall topology of the issues and describes both areas covered and areas not covered. Finally, by defining the issues through specific examples and conceptual classification, rather than implementation details or linguistic theories, the Sourcebook is accessible to non-specialists in computational linguistics.

In the hierarchical classification, groups I, II and III roughly match stages of development in natural language systems. They correspond to simple database query systems (I), database systems capable of extended interaction (II) and systems where knowledge flow between user and system goes both ways (III). Type III systems will be needed for, e.g., intelligent interfaces to expert systems. Progress on problems in areas I, II and III can be considered as describing *first*, *second* and *third* generation natural language systems, respectively.

### Continuing and Future Work

We are continuing to add exemplars to the Sourcebook and are elaborating the classification scheme. We will be making the Sourcebook available to other researchers for comment and analysis.

We have several hundred exemplars and we estimate that we have covered 10 per cent of the relevant literature (journals, proceedings volumes, dissertations, major textbooks) in computational linguistics, artificial intelligence and cognitive science. Our intention is to be as exhaustive as possible. Which leaves us with a very ambitious project.

We are also continuing work on the process evaluation methodologies.

## Appendix 1: Sample Exemplars

### Exemplar 1

(1) I heard an earthquake singing in the shower.
(2) I heard an earthquake sing in the shower. (Wilks, 1986, p. 199)

### Topic

Case ambiguity.

### Discussion

In (1), we know the speaker is the one singing in the shower. How? Because we know that earthquakes don't sing. So it is likely that there is a missing "while" and the speaker heard an earthquake while singing in the shower. However, that reasoning fails on (2). In that sentence, the earthquake is singing, not the person in the shower. A selectional restriction that says earthquakes don't sing will work in understanding (1) but fail for (2). How is the correct actor for actions like singing determined?

### References

Yorick Wilks. (1986). An Intelligent Analyzer and Understander of English. In Barbara J. Grosz, Karen Sparck Jones, and Bonnie Lynn Webber (Eds.), *Readings in Natural Language Processing.* Morgan Kaufman. Page 199.

### Exemplar 2

User: Add a dual disk to the order.
System: A dual ported disk. What storage capacity? (Carbonell & Hayes, 1983, p. 133)

### Topic

Intersentential Ellipsis — Echo

### Discussion

The response by the system is a form of elaboration ellipsis. The system intends to confirm the missing information and gather more needed information without interrupting the conversational flow. In each case, the utterance must be recognized as referring to the topic introduced by the user. This kind of cooperative dialogue is very common when the user believes that he is dealing with someone who understands natural language. We often assume that "understanding language" means understanding the user's goals and sharing common assumptions.

### Reference

Carbonell, James G. & Hayes, Philip J. (1983). Recovery strategies for parsing extragrammatical language. *American Journal of Computational Linguistics, 9*, 123-146.

### Exemplar 3

Jim Fixx had a heart attack while jogging.

### Topic

Recognizing interesting information — situational irony.

### Discussion

Syntactically and semantically this sentence is straightforward. However, to a reader who knows Jim Fixx as an author of books promoting jogging for health the information is very interesting. The interest comes from the irony of the situation. How is the irony, the point of the sentence, recognized? Extracting the irony requires accessing the relevant beliefs of the characters and recognizing violations of beliefs.

### Reference

Peter Norvig. (1983). Six Problems for Story Understanders. *Proceedings of The National Conference on Artificial Intelligence*, 284-287.
Dyer, M. G., Flowers, M. & Reeves, J. F. (in press). Recognizing Situational Irony: A Computer Model of Irony Recognition and Narrative Understanding. *Advances in Computing and the Humanities, 1(1).*

## Exemplar 4

(1) I want to meet the chairman of Chrysler.
(2) I want to be the chairman of Chrysler.
(3) I want to be the Newton of AI.

## Topic

Definite reference - referential vs attributive.

## Discussion

Resolving definite references requires that the system distinguish between referential and attributive uses. In (1), 'the chairman of Chrysler' refers to the current holder of that position, presently Lee Iacocca. But in (2) the speaker doesn't want to be Iacocca but rather to hold the job Iacocca holds. In (1) 'the chairman of Chrysler' is said to be referential because it refers to a specific object. In (2) it is said to be attributive because it describes a characteristic or set of characteristics. In (3) the use is metaphorical, referring to the historical role that Newton played in physics rather than any particular job Newton held. For example, it doesn't not mean that the speaker wants to be the AI equivalent of the director of the mint in England. Recognizing the reference in these cases requires that the system be able to process several levels of abstraction and, especially for (3), to access world knowledge. (Cf. Allen, p. 355.)

## Reference

Allen, J. F. (1987). *Natural Language Understanding.* Menlo Park, CA: Benjamin/Cummings.

### Exemplar 5

(1) The next day after we sold our car, the buyer returned and wanted his money back. (Allen, 1987, p. 346)
(2) The day after we sold our house, the escrow company went bankrupt.
(3) The day after we sold our house, they put in a traffic light at the corner.

### Topic

Anaphoric reference - roles.

### Discussion

In (1) the 'buyer' refers back to a figure in one of the roles in the 'selling a car' event. The system must search not only the direct possible antecedents (the 'selling') but must also consider aspects of the selling to resolve the 'buyer' reference. In (1), there is nothing specific to 'car' about resolving the reference. But in (2), finding the reference of 'the escrow company' involves looking past the general "buying" script and searching through aspects of selling specific to selling houses. This might require extensive local knowledge of the typical ways in which houses are bought and sold in this area. There is a general problem here with controlling the amount of search while still looking deeply enough. In (3), the system has to go from the house to the location of the house to the street that runs past the house to the corner at a nearby intersection of the street to understand the reference.

### Reference

Allen, J. F. (1987). *Natural Language Understanding.* Menlo Park, CA: Benjamin/Cummings.

## Appendix 2: Brief View of the Evolving Classification

I. Single-utterance issues.

- A. Identification of syntactic units.
- B. Ambiguity.
  - i. Lexical ambiguity.
  - ii. Case ambiguity.
- C. Modifier attachment.
- D. Reference.
- E. Metaphor and novel language.
- F. Other

II. Connected-utterance issues.

- A. Anaphora.
- B. Ellipsis.
  - i. Intersentential ellipsis.
    - a. Intersentential ellipsis — echo.
- C. Integrating complex information.
- D. Reasoning, argumentation, story understanding.
  - i. Interest.
    - a. Irony.
- E. Other.

III. True-dialogue issues.

- A. Recognizing user goals and plans.
- B. Using and modifying models of the user's knowledge.
- C. Recognizing logical presuppositions.
- D. Speech acts.
- E. Meta-linguistic discourse.
- F. Other.

IV. Generation.

V. Ill-formed input.

## Bibliography


Bara, B. G. & Guida, G. (1984). Competence and Performance in the Design of Natural Language Systems. In B. G. Bara & G. Guida (Eds.), *Computational Models of Natural Language Processing* (pp. 1-7). Amsterdam: North-Holland.

Finin, T., Goodman, B. & Tennant, H. (1979). JETS: Achieving Completeness through Coverage and Closure. *Proceedings of the Sixth International Joint Conference on Artificial Intelligence*, 275-281.

Guida, G. & Mauri, G. (1984). A Formal Basis for Performance Evaluation of Natural Language Understanding Systems. *Computational Linguistics, 10*, 15-30.

Guida, G. & Mauri, G. (1986). Evaluation of Natural Language Processing Systems: Issues and Approaches. *Proceedings of the IEEE, 74*, 1026-1035.

Tennant, H. (1979). Experience with the Evaluation of Natural Language Question Answerers. *Proceedings of the Sixth International Joint Conference on Artificial Intelligence*, 874-876.

Woods, W. A., (1977). A Personal View of Natural Language Understanding. *SIGART Newsletter*, 17-20.