# DESIGN DIMENSIONS FOR NON-NORMATIVE UNDERSTANDING SYSTEMS

Robert J. Bobrow
Madeleine Bates
Bolt Beranek and Newman Inc.
10 Moulton Street
Cambridge, Massachusetts 02238

## 1. Introduction

This position paper is not based upon direct experience with the design and implementation of a "non-normative" natural language system, but rather draws upon our work[1] on cascade [11] architectures for understanding systems in which syntactic, semantic and discourse processes cooperate to determine the "best" interpretation of an utterance in a given discourse context. The RUS and PSI-KLONE systems [1, 2, 3, 4, 5], which embody some of these principles, provide a strong framework for the development of non-normative systems as illustrated by the work of Sondheimer and Weischedel [8, 9, 10]and others.

Here we pose a number of questions in order to clarify the theoretical and practical issues involved in building "non-normative" natural language systems. We give brief indications of the range of plausible answers, in order to characterize the space of decisions that must be made in designing such a system. The first questions cover what is intended by the ill-defined term "non-normative system", beyond the important but vague desire for a "friendly and flexible" computer system. The remaining questions cover several of the architectural issues involved in building such a system, including the categories of knowledge to be represented in the system, the static modularization of these knowledge sources, and the dynamic information and control flow among these modules.

## 2. System performance goals

**What are the overall performance objectives of the system?**

Marcus has argued [7] that the "well-formedness" constraints on natural language make it possible to parse utterances with minimal (or no) search.[2] The work we have done on the RUS system has convinced us that this is true and that cascading semantic interpretation with syntactic analysis can further improve the efficiency of the overall system. The question naturally arises as to whether the performance characteristics of this model must be abandoned when the input does not satisfy the well-formedness constraints imposed by a competence model of language. We believe that it is possible to design natural language systems that can handle well-formed input efficiently and ill-formed input effectively.

The way the system is to deal with ill-formed input depends in a strong way on how much the system is expected to do with well-formed input. Ad hoc data base retrieval systems (a currently hot topic) pose different constraints than systems that are expected to enter into a substantial dialogue with the user. When the behavior of the system is severely limited even given perfect input, the space of plausible inputs is also limited and the search for a reasonable interpretation for ill-formed input can be made substantially easier by asking the user a few well-chosen questions. In the dbms retrieval domain, even partially processed input can be used to suggest what information the user is interested in, and provide the basis for a useful clarification dialogue.

**What is the system expected to do with ill-formed input?**

The system may be expected to understand the input but not provide direct feedback on errors (e.g. by independently deciding on the (most plausible) interpretation of the input, or by questioning the user about possible alternative interpretations). Alternatively, the system might provide feedback about the probable source of its difficulty, e.g. by pointing out the portion of the input which it could not handle (if it can be localized), or by characterizing the type of error that occurred and describing general ways of avoiding such errors in the future.

## 3. Architectural issues

In order to design a fault-tolerant language processing system, it is important to have a model for the component processes of the system, how they interact in handling well-formed input, and how each process is affected by the different types of constraint violations occurring in ill-formed input.

**What categories of knowledge are needed to understand well-formed input, and how are they used?**

Typically, a natural language understanding system makes use of lexical and morphological knowledge (to categorize the syntactic and semantic properties of input items), syntactic knowledge, semantic knowledge, and knowledge of discourse phenomena (here we include issues of ellipsis, anaphora and focus, as well as plan recognition ("why did he say this to me now?") and rhetorical structure). Of course, saying that these categories of knowledge are represented does not imply anything about the static (representational) or dynamic (process interaction) modularization of the resulting system.

We will assume that the overall system consists of a set of component modules. One common decomposition has each category of knowledge embodied in a separate component of the NLU system, although it is possible to fuse the knowledge of several categories into a single process. Given this assumption, we must then ask what control and information flow can be imposed on the interaction of the modules to achieve the overall performance goals imposed on the system.

In analyzing how violations of constraints affect the operation of various components, it is useful to distinguish clearly between the information used within a component to compute its output, and the structure and content of the information which it passes on to other components. It is also important to determine how critically the operation of the receiving component depends on the presence, absence or internal inconsistency of various features of the inter-component information flow.

As an example, we will consider the interaction between a syntactic component (parser) and a semantic interpretation component. Typically, the semantic interpretation process is componential, building up the interpretation of a phrase in a lawful way from the interpretations of its constituents. Thus a primary goal for the parser is to determine the (syntactically) acceptable groupings of words and constituents (a constituent structure tree, perhaps augmented by the equivalent of traces to tie together components). Unless such groupings can be made, there is nothing for the semantic interpreter and subsequent components to operate on. Some syntactic features are used only within the parser to determine the acceptability of possible constituent groupings, and are not passed to the semantic component (e.g. some verbs take clause complements, and require the verb in the complement to be subjunctive, infinitive, etc.).

The normal output of the parser may also specify other properties of the input not immediately available from the lexical/morphological analysis of individual words, such as the syntactic number of noun phrases, and the case structure of clauses. Additionally, the parser may indicate the functional equivalent of "traces", showing how certain constituents play multiple roles within a structure, appearing as functional constituents of more than one separated phrase. From the point of view of semantics, however, the grouping operation is of primary importance, since it is difficult to reconstruct the intended grouping without making use of both local and global syntactic constraints. The other results of the parsing process are less essential. Thus, for example, the case structure of clauses is often highly constrained by the semantic features of the verb and the constituent noun phrases, and it is possible to reconstruct it even with minimal syntactic guidance (e.g. "throw" "the ball" "the boy").

**How can each component fill its role in the overall system when the constraints and assumptions that underlie its design are violated by ill-formed input?**

The distinction between the information used within a component from the information which that component is required to provide to other components is critical in designing processing strategies for each component that allow it to fulfill its primary output goals when its input violates one or more well-formedness constraints. Often more than one source of information or constraint may be available to determine the output of a component, and it is possible to produce well-formed output based on the partial or conflicting internal information provided by ill-formed input. For example, in systems with feedback between components, it is possible for that feedback to make up for lack of information or violation of constraints in the input, as when semantic coherence between subject and verb is sufficient to override the violation of the syntactic number agreement constraint. When the integrity of the output of a component can be maintained in the face of ill-formed input, other components can be totally shielded from the effects of that input.

A clear specification of the interface language between components makes it possible to have recovery procedures that radically restructure or totally replace one component without affecting the operation of other components. In general, the problem to be solved by a non-normative language understander can be viewed as one of finding a "sufficiently good explanation" for an utterance in the given context.[3] A number of approaches to this problem can be distinguished. One approach attempts to characterize the class of error producing mechanisms (such as word transposition, mistyping of letters, morphological errors, resumptive pronouns, etc.). Given such a characterization, recognition criteria for different classes of errors, and procedures to invert the error process, an "explanation" for an ill-formed utterance could be generated in the form of an intended well-formed utterance and a sequence of error transformations. The system would then try to understand the hypothesized well-formed utterance. While some "spelling corrector" algorithms use this approach, we know of no attempt to apply it to the full range of syntactic, semantic and pragmatic errors. We believe that some strategies of this sort might prove useful as components in a larger error-correcting system.

A more thoroughly explored set of strategies for non-normative processing is based on the concept of "constraint relaxation". If a component can find no characterization of the utterance because it violates one or more constraints, then it is necessary to _relax_ such constraints. A number of strategies have been proposed for relaxing well-formedness constraints on input to permit components to derive well-structured output for both well-formed and ill-formed input:

1. extend the notion of well-formed input to include (the common cases of) ill-formed input (e.g. make the grammar handle ill-formed input explicitly);
2. allow certain specific constraints to be overridden when no legal operation succeeds;
3. provide a process that can diagnose failures and flexibly override constraints.

Somehow the "goodness" of an explanation must be related to the number and type of constraints which must be relaxed to allow that explanation. How good an explanation must be before it is accepted is a matter of design choice. Must it simply be "good enough" (above some threshold), or must it be guaranteed to be "the best possible" explanation? If it must be "the best possible", then one can either generate all possible explanations and compare them, or use some strategy like the shortfall algorithm [12] that guarantees the first explanation produced will be optimal.

While space prohibits discussion of the advantages and disadvantages of each of these strategies, we would like to present a number of design dimensions along which they might be usefully compared. We believe that choices on these dimensions (made implicitly or explicitly) have a substantial effect on both the practical performance and theoretical interest of the resulting strategies. These dimensions are exemplified by the following questions:

o Does the component have an explicit internal competence model that is clearly separated from its performance strategies?[4]

o What information is used to determine which constraints to attempt to relax? Is the decision purely local (based on the constraint and the words in the immediate vicinity of the failure) or can the overall properties of the utterance and/or the discourse context enter into the decision?

o When is relaxation tried? How are various alternatives scheduled? Is it possible, for example, that a "parse" including the relaxation of a syntactic constraint may be produced before a parse that involves no such relaxation?

o Does the technique permit incremental feedback between components, and is such feedback used in determining which constraints to relax?[5]

**Non-syntactic ill-formedness**

While the overall framework mentioned above raises questions about errors that affect components other than syntax, the discussion centers primarily on syntactic ill-formedness. In this we follow the trend in the field. Perhaps because syntax is the most clearly understood component, we have a better idea as to how it can go wrong, while our models for semantic interpretation and discourse processes are much less complete. Alternatively, it might be supposed that the parsing process as generally performed is the most fragile of the components, susceptible to disruption by the slightest violation of syntactic constraints. It may be that more robust parsing strategies can be found.

Without stating how the semantic component might relax its constraints, we might still point out the parallel between constraint violation in syntax and such semantic phenomena as metaphor, personification and metonymy. We believe that, as in the syntactic case, it will be useful to distinguish between the internal operation of the semantic interpreter and the interface between it and discourse level processes. It should also be possible to make use of feedback from the discourse component to overcome violations of semantic constraints. In the context of a waiter talking to a cook about a customer complaint, the sentence "The hamburger is getting awfully impatient." should be understood.

## 4. Conclusions

We believe that it will be possible to design robust systems without giving up many valuable features of those systems which already work on well-formed input. In particular, we believe it will be possible to build such systems on the basis of competence models for various linguistic components, which degrade gracefully and without the use of ad hoc techniques such as pattern matching.

One critical resource that is needed is a widely available, reasonably large corpus of "ill-formed input", exhibiting the variety of problems which must be faced by practical systems. This corpus should be sub-divided by modality, since it is known that spoken and typewritten interactions have different characteristics. The collections that we know of are either limited in modality (e.g. the work on speech errors by Fromkin [6]) or are not widely available (e.g. unpublished material collected by Tony Kroch). It would also be valuable if this material were analyzed in terms of possible generative mechanisms, to provide needed evidence for error recovery strategies based on inversion of error generation processes.

Finally, we believe that many error recovery problems can be solved by using constraints from one knowledge category to reduce the overall sensitivity of the system to errors in another category. To this end, work is clearly needed in the area of control structures and cooperative process architectures that allow both pipelining and feedback among components with vastly different internal knowledge bases.

---

[1]The preparation of this paper was supported by the Advanced Research Projects Agency of the Department of Defense, and monitored by the Office of Naval Research under contract N00014-77-C-0378.

[2]The parser designed by G. Ginsparg also has similar search characteristics, given grammatical input.

[3]What constitutes "sufficiently good" depends, of course, on the overall goals of the system.

[4]In almost any case, we believe, the information available at the interface between components should be expressed primarily in terms of some competence model.

## REFERENCES

1. Bates, M., Bobrow, R. J. and Webber, B.L. Tools for Syntactic and Semantic Interpretation. BBN Report 4785, Bolt Beranek and Newman Inc., 1981.

2. Bates, M. and Bobrow, R. J. The RUS Parsing System. Bolt Beranek and Newman Inc., forthcoming.

3. Bobrow, R. J. The RUS System. BBN Report 3878, Bolt Beranek and Newman Inc., 1978.

4. Bobrow, R. J. & Webber, B. L. PSI-KLONE - Parsing and Semantic Interpretation in the BBN Natural Language Understanding System. CSCSI/CSEIO Annual Conference, CSCSI/CSEIO, 1980.

5. Bobrow, R. J. & Webber, B. L. Knowledge Representation for Syntactic/Semantic Processing. Proceedings of The First Annual National Conference on Artificial Intelligence, American Association for Artificial Intelligence, 1980.

6. Fromkin, Victoria A.. Janua Linguarum, Series maior. Volume 77: Speech Errors as Linguistic Evidence. Mouton, The Hague, 1973.

7. Marcus, M.. A Theory of Syntactic Recognition for Natural Language. MIT Press, 1980.

8. Sondheimer, N.K. and Weischedel, R.M. A Rule-Based Approach to Ill-Formed Input. Proc. 8th Int'l Conf. on Computational Linguistics, Tokyo, Japan, October, 1980, pp. 46-54.

9. Weischedel, Ralph M. and Black, John E. "If The Parser Fails." Proceedings of the 18th Annual Meeting of the ACL (June 1980).

10. Weischedel, Ralph and Sondheimer, Norman. A Framework for Processing Ill-Formed Input. Department of Computer and Information Sciences, University of Delaware, October, 1981.

11. Woods, W. A. "Cascaded ATN Grammars." Amer. J. Computational Linguistics 6, 1 (Jan.-Mar. 1980).

12. Woods, W. A. "Optimal Search Strategies for Speech Understanding Control." Artificial Intelligence 18, 3 (June 1982).