[Proceedings of the *Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages*, Colgate University, Hamilton, New York, August 14-16, 1985]

Reflections on the Knowledge Needed to Process Ill-Formed Language[1]

Ralph M. Weischedel
Lance A. Ramshaw
Bolt Beranek and Newman Inc.

Abstract

This paper reflects about the kinds of morphological, syntactic, semantic, and pragmatic knowledge needed to process ill-formed input. We conclude that an excellent start on processing ill—formed input has been exemplified in a number of concrete implementations, but that a substantial amount of fundamental work must still be done if our systems are to understand language robustly to the degree that humans do. Furthermore, we conclude that studying ill—formed language offers important perspectives on the knowledge and architecture needed to correctly understand natural languages.

1 Introduction

In the past five years there has been considerable interest in processing ill-formed input. That resulted in several implementations processing various classes of ill-formedness. These include EPISTLE [9], NOMAD [7], EXCALIBUR [3], and a system based on meta-rules [11, 14]. All deal with one or more classes of syntactic ill-formedness, and some work [7, 14] even discusses certain classes of semantic ill—formedness. Various techniques have been used for processing input in the face of ill-formedness.

- o employing only syntactic constraints [9],
- o employing predominantly semantic constraints [3],
- o employing both syntactic and semantic constraints [7, 11, 14].

[1] This work has been partially supported by a grant from the National Science Foundation, IST-8419162.

Rather than describing another implementation, in this paper we present a few reflections on the limitations of these systems by showing the kinds of knowledge that are necessary to have a fairly complete, robust understanding of natural language. Section 2 discusses morphology and phonetics. Section 3 discusses the role of syntax. In Sections 4 and 6, we discuss the limitations of semantics and of pragmatic knowledge, respectively. Section 5 presents the prospects that arise if pragmatic knowledge can be effectively incorporated, and Section 7 presents the potential of combining sources of knowledge. Section 8 concludes.

2 Unexplored Morphology and Phonetics

Spelling errors are prime candidates for applying morphological and phonetic knowledge Of course, most systems deal with unrecognized items in the input by having a model of typical typographical errors and the correct form. In describing heuristics for analyzing typographical errors when an unrecognized symbol occurs in the input, we tend to describe such heuristics as "misspelling correction." However, there are several kinds of knowledge regarding true misspellings, which to our knowledge, have not been encoded in systems heretofore. For instance, there are probably many specific patterns of typical spelling errors of native speakers of English, the rule "i before e except after c and in words like *neighbor* and *weigh"* reflects one pattern of misspelling that is not the case of simply transposing two letters when typing. Furthermore, there are problems that occur due to errors in spelling phonetically, such as *ph* and *f*. There are "spelling errors" that seem related to remembering or transcribing phonetic representations, e.g., *Kane* vs. *Caine* or the phone message that one of my supervisors received, *Please call terminal main for insulation. Thank you.* for *Please call terminal man for installation. Thank you.*

Misspellings or typographical errors that produce a symbol which is recognized as a word have generally been untouched, though [13] is an exception. Those kinds of spelling errors or typographical errors are far more challenging since there is no overt sign in the input regarding what is wrong. The system must mistrust the symbols actually present and determine which one to revise. Studies of examples like that may suggest both new kinds of knowledge that should be incorporated in natural language understanders and also new architectures for interactions among the kinds of knowledge.

An example of how pragmatic knowledge could influence recovery from spelling errors is given in Section 5.

3 Role of Syntax

It has at times seemed almost fashionable to try to build natural language processors employing as little syntax as possible in the analysis phase. The attractiveness of that approach is the apparent simplicity that comes with ignoring a difficult component in the system and the fact that language can be understood by humans in spite of very poor syntactic form. One need only look at transcribed spoken language or at headlines to see that syntactic rules are frequently violated. However, the argument that humans can understand language in spite of poor use of syntax does not argue that people do not have a syntactic model, nor that it is inapplicable in human or machine processing.

On the contrary, there is quite strong evidence that humans have a very clear model of the syntax of their natural language. Humans can edit written language exhibiting poor syntax and can edit correct grammatical structures to create a new form which is deemed to be clearer to the reader. Furthermore, syntactic constraints, like morphological, semantic, and pragmatic constraints, all may be employed to determine what interpretation is intended. For instance, in *How many glumpfs are in it?,* syntax suggests that *glumpfs* is a noun.

Ignoring syntactic constraints can lead to seeing ambiguity where there is none. Consider subject-verb agreement, which appears to be a very minor syntactic constraint in English and which is rather frequently violated; one case study [6] found that 2.3 % of natural language queries to a data base violated the subject-verb agreement constraint. Nevertheless, the constraint is a part of the language, and if ignored, would mean that a system or person could not distinguish between the following two forms:

1. *List all assets of the company that was bought by XYZ Corp.*
2. *List all assets of the company that were bought by XYZ Corp.*

As a consequence, we conclude that all syntactic constraints should be employed in natural language understanding systems, otherwise, forms that are perfectly clear to humans will not be correctly understood by machines. Ill-formed input tends to highlight the need for using all constraints including syntactic constraints, for an ill-formed input, one or more of the well-formedness constraints of correct language is violated. Relaxing or ignoring a potential constraint opens up the search space of possible interpretations, thereby making it even more critical to use all other constraints to prune the search and minimize the combinatorics.

4 Limitations of Semantics

It is commonly agreed that for most applications semantic constraints such as selection restrictions must be applied in order to determine what is intended by an ill-formed input. Exceptions to this include some applications where precise understanding is unnecessary, such as some checks for grammaticality and style; see [4, 9].

However, the limitations of such semantic constraints are not that well known. Suppose we are building a student advisement system which knows about courses, majors, degree requirements, registration instructors, and policies. If the input is *Can I FROB Sociology 101?,* then there are several verbs which might be well-known to the system having the case frame *person* <verb> *course.* These include *add, drop, transfer, pass, take* and *fail.* All satisfy the syntactic context as well. If the syntactic context is less constraining, then the number of alternatives is even larger. In *Is it a FROB?,* if *it* is known to refer to a course then *FROB* could be any adjective or noun which applies to a course.

Furthermore, it appears that even in the most highly constrained of applications, the same phenomenon occurs. For instance, natural language access to computer mail systems would seem to be one of the most highly constrained environments, see [5, 8]. Entities in that domain are limited to messages, addresses, persons, sites, times, etc. Yet, if one says, *FROB the message to Jones,* almost any operation in the mail domain is a possibility, since *to Jones* could be dative or *the message to Jones* could be a reference. If both interpretations for *to Jones* are possible, then *send, resend, forward, delete, move,* etc. are all possible.

The depth of the problem becomes even more clear when one realizes that selection restrictions or semantic constraints themselves are not sacred. For instance, in our experience with the RUS parser, the most frequent reason that a sentence fails to parse is not due to syntactic limitations but rather is due to limitations in the set of selection restrictions encoded or to limitations in their use. Even when it is possible to have as general a model of selection restrictions and semantics as we have now for grammar, there are still clear cases where selection restrictions are violated, such as metonymy, synecdoche, and metaphor, or where their constraining effect is weakened via relatively neutral noun phrases, such as pronouns, *stuff, thing,* and *gift.*

There still seems to be a need for clear engineering and linguistic criteria for specifying appropriate grain size on semantic classes (or semantic features) and how to recognize from a given noun phrase what semantic class is associated with the noun phrase. For instance, suppose our grain size dictates that we should have a semantic class such as weapon so that we could specify that the logical object of *fire* is a weapon. Then one has to determine whether it makes sense to determine whether *toy gun* and *fake gun* should be considered in the class of weapons.

5 Prospects from Pragmatics

What is needed is a model of the intent of the user based on pragmatic knowledge, including at least the plans and goals of the user. Some work that has begun to apply richer pragmatic models appears in [2, 7]

There are several kinds of knowledge that are pragmatic in nature and that could potentially be brought to bear on understanding natural language, whether well-formed or ill-formed. For this paper we focus primarily on knowledge about the plans and goals of a speaker in the framework first proposed by Allen [1] and followed up in [2, 10, 12]. Therefore, we can assume we have available a tree representing the goals and subgoals that a user may have in mind to accomplish. A particular path in that tree represents the stack of pending goals that the user may have.

Such a context can add further constraint on the search space in a natural language processor. If the speaker has just previously said, *I did not preregister for enough credits,* then that context might be adequate to set up a tree of subgoals indicating that *FROB* in *Can I FROB Sociology 101?* is likely to be synonymous with *add*.

The kind of knowledge and reasoning presumed in the example above is well beyond what anyone has accomplished thus far. However, in [2], a model of user plans and goals has been applied to two problems related to ill-formedness: contextual ellipsis and pragmatic overshoot[2] . Using rich pragmatic models is clearly a most important direction for future work, for it implies an ability to follow the intent of the user through a dialog at a level of detail that is useful not only for understanding ill-formed input but also for reducing ambiguity and providing appropriate responses.

[2] By *pragmatic overshoot,* one means requests that do not make sense in the underlying application system, such as requesting the rental fee of condominiums.

6 Limitations of Pragmatics

Though the kind of analysis in the previous section is very promising, the combinatorics still can be quite large. For instance, if the top of the stack is a goal such as fulfilling a distribution requirement, the request of the individual need not be an immediate descendant of that goal in some goal tree. In *I need to fulfill the Group 1 requirement. Is there FROB in History 101?,* presumably the subgoal of fulfilling the distribution requirement is to take a certain number of courses, to take a course, there must be adequate openings in that course at the time of registration. Consequently, the possible predicates and entities that need to be examined May appear more than one generation below the current goal. Similarly, the input might relate to a node deeper in the stack (higher in the tree) or one of its descendants. An example is

Student: *I need to fulfill the Group 1 requirement.*
Does Dr. Arnold teach any sections of PolySci 101?
System: <no>
Student: *Is History 101 FROB?'*

Here it seems plausible that the student has examined a lower level goal regarding Dr. Arnold's course offerings and is returning to the higher level goal of fulfilling the Group 1 requirement. In that context, *FROB* could be the equivalent of *appropriate.*

Pragmatic context may be able to at least limit the number of alternatives sufficiently to warrant reporting those alternatives to the user for his/her selection. Suppose we have the following input: *Last semester I was unable to get into CS 105. How many FROBS are in it?* If the system can recognize that the student is likely to want to know if there are spaces available in the course, then it could know the student is asking either for the number of students already registered or the number of spaces available and present him/her with the alternative. (Of course if the system is that smart, it may be able to in fact determine that the number of open spaces would supersede the answer of how many spaces are already taken.)

Therefore, given a tree of subgoals representing a strategy to achieve a goal and the path in that tree representing the goals inherent in the last input, virtually any node in the tree or any new descendants of its leaves may arise as a goal of a new ill-formed input. The number of alternative subgoals which could contribute to correctly interpreting an ill-formed input is therefore large, though not all alternatives are necessarily equally likely.

7 The Need for Combining Knowledge Sources

With an ill-formedness the input cannot be understood due to problems with the input or deficiencies in the understanding system, the reason could be any of:

- an error in an input symbol,
- inadequate lexical information,
- ungrammaticality,
- inadequate grammar,
- a semantic error,
- a figure of speech,
- incomplete selection restrictions,
- overly restrictive case frame constraints,
- non-felicitous input, or
- incomplete dialog models.

In the face of all the alternatives for what might prevent the system from understanding the input, all the knowledge and constraints available must be applied to determine what is intended.

Consider an input containing an unknown word *FROBBED* in the form *Is History 101 FROBBED?* Case frame constraints give little indication of what *FROBBED* might mean, since a very large number of predicates apply to courses. Syntax helps us little, since *FROBBED* could be either a noun, a proper noun, an adjective, or a past participle. Additional knowledge can limit the alternatives however; if the word were capitalized, one could assume it is a proper noun. Noticing the *ed* ending and repeated final consonant, the system could propose using the morphological information that *FROB* is a verb whose past participle is *FROBBED.* Pragmatic information can further limit the alternatives. For instance, if the input occurred in the context of *I need to take another Group 1 course,* the system could look at predicates associated with registration for the course such as being filled, the schedule of its being offered, etc. Using all the constraints together, the system could have a ranked ordering of the alternatives it believes likely and suggest them to the user; one would be *Is History 101 filled?.*

In a second example, phonetic knowledge and pragmatic knowledge play a crucial role. Though we have focused on models of user plans and goals as a kind of pragmatic knowledge, other kinds would also be very useful. For instance suppose the system knows of no Professor Chaminski. The request, *Does Dr. Chaminski teach any section of History 101?,* would then not make sense. Phonetic similarity might suggest two alternatives for Chaminski: *Charinski* and *Kasinski.* If the system knows that Charinski is in the EE Department and that Kasinski is in the History Department, then it might be able to reason that the student probably means Kasinski. The system could then answer appropriately, *Dr. Kasinski is teaching History 311 and History 620.*

8 Conclusions

Our first conclusion is that all forms of knowledge that may be used as constraints are potentially critical to understanding an ill-formed input. Though we have focused primarily in this paper on examples regarding unknown words, the same principles seem to hold for the broad class of ill-formed inputs.

Our second conclusion is that work resulting in current implementations certainly has taken initial steps that should markedly improve the robustness and user-friendliness of applied natural language processors. Nevertheless, systems that hope to approach the performance of humans in understanding ill-formed language must incorporate far more knowledge than simply syntax and semantics[3].

Third, the problems of ill-formedness appear to offer an important opportunity for studying knowledge sources and architectures for understanding natural language. Ill-formed input requires relaxing the rules that normally constrain search or requires doubting the symbols received as input. This suggests using all sources of knowledge, thereby exposing issues which might not surface so readily in studies of well-formed input in limited domains, where redundancy in the input, domain, and context may let one get by with fewer knowledge sources and simpler architectures.

[3] At present, the approaches of [7, 14] have paid most attention to such extensions.

References

[l] James F. Allen and C.R. Perrault. -
Analyzing Intention in Utterances.
Artificial Intelligence 15(3), December, 1980.

[2] Mary Sandra Carberry.
Pragmatic Modeling in Information System Interfaces.
PhD thesis, University of Delaware, August, 1985.

[3] Jaime G. Carbonell and Philip J. Hayes.
Recovery Strategies for Parsing Extragrammatical Language.
American Journal of Computational Linguistics 9(3-4): 123-146, 1983.

[4] L.L. Cherry, W. Vesterman.
Writing Tools - The STYLE and DICTION Programs.
Technical Report 9, Computing Science, Bell Laboratories, Murray Hill, NJ, 1980.

[5] T Kaczmarek, W. Mark, and N. Sondheimer.
The Consul/CUE Interface: An Integrated Interactive Environment.
In *Proceedings of CHI '83 Human Factors in Computing Systems,* pages 98-102. ACM, December, 1983.

[6] C.M Eastman and D.S. McLean.
On the Need for Parsing Ill-Formed Input.
American Journal of Computational Linguistics 7(4):257, October-December, 1981.

[7] Richard H. Granger.
The NOMAD System: Expectation-Based Detection and Correction of Errors during Understanding of Syntactically and Semantically Ill-Formed Text.
American Journal of Computational Linguistics 9(3-4): 188-198, 1983.

[8] P.J. Hayes and R. Reddy.
An Anatomy of Graceful Interaction in Man-Machine Communication.
Technical Report, Carnegie-Mellon University, 1979.

[9] K. Jensen, G.E Heidorn, L.A. Miller, and Y Ravin,
Parse Filling and Prose Fixing: Getting a Hold on Ill-Formedness.
American Journal of Computational Linguistics 9(3-4): 147-160, 1983.

[10] Diane J. Litman and James F. Allen.
A Plan Recognition Model for Clarification Subdialogues.
In *Proceedings of Coling84,* pages 302-311. Association for Computational Linguistics. July, 1984.

[11] Amir M. Razi.
An Empirical Study of Robust Natural Language Processing.
PhD thesis, University of Delaware, June, 1985.

[12] Candace L. Sidner.
Plan Parsing for Intended Response Recognition in Discourse.
Computational Intelligence 1(1):1 -10, February, 1985.

[13] David James Trawick.
Robust Sentence Analysis and Habitability.
PhD thesis, California Institute of Technology, February, 1983.

[14] Weischedel, Ralph M. and Sondheimer, Norman K.
Meta-rules as a Basis for Processing Ill-Formed Output.
American Journal of Computational Linguistics 9(3-4): 161 - 177, 1983.