Scruffy Text Understanding: Design and Implementation of 'Tolerant' Understanders

Richard H. Granger

Artificial Intelligence Project
Computer Science Department
University of California
Irvine, California 92717

ABSTRACT

Most large text-understanding systems have been designed under the assumption that the input text will be in reasonably "neat" form, e.g., newspaper stories and other edited texts. However, a great deal of natural language text, e.g., memos, rough dratts, conversation transcripts, etc., have features that differ significantly from "neat" texts, posing special problems for readers, such as misspelled words, missing words, poor syntactic construction, missing periods, etc. Our solution to these problems is to make use of expectations, based both on knowledge of surface English and on world knowledge of the situation being described. These syntactic and semantic expectations can be used to figure out unknown words from context, constrain the possible word-senses of words with multiple meanings (ambiguity), fill in missing words (ellipsis), and resolve referents (anaphora). This method of using expectations to aid the understanding of "scruffy" texts has been incorporated into a working computer program called NOMAD, which understands scruffy texts in the domain of Navy messages.

1.0 Introduction

Consider the following (scribbled) message, left by a computer science professor on a colleague's desk:

[1] Met w/chrmn agreed on changes to prposl nxt mtg 3 Feb.

A good deal of informal text such as everyday messages like the one above are very ill-formed grammatically and contain misspellings, ad hoc abbreviations and lack of important punctuation such as periods between sentences. Yet people seem to easily understand such messages, and in fact most people would probably understand the above message just as readily as they would a more "well-formed" version:

"I met with the chairman, and we agreed on what changes had to be made to the proposal. Our next meeting will be on Feb. 3."

This research was supported in part by the Naval Ocean Systems Center under contract N-00123-81-C-1078.

No extra information seems to be conveyed by this longer version, and message-writers appear to take advantage of this fact by writing extremely terse messages such as [1], apparently counting on readers' ability to analyze them in spite of their messiness.

If "scruffy" messages such as this one were only intended for a readership of one, there wouldn't be a real problem. However, this informal type of 'memo' message is commonly used for information transfer in many businesses, universities, government offices, etc. An extreme case of such an organization is the Navy, which every hour receives many thousands of short messages, each of which must be encoded into computer-readable form for entry into a database. Currently, these messages come in in very scruffy form, and a growing number of man-hours is spent on the encoding-by-hand process. Hence there is an obvious benefit to partially automating this encoding process. The problem is that most existing text-understanding systems (e.g. ELI [Riesbeck and Schank 76], SAM [Cullingford 77], FRUMP [DeJong 79], IPP [Lebowitz 80]) would fail to successfully analyze ill-formed texts like [1], because they have been designed under the assumption that they will receive 'neater' input, e.g., edited input such as is found in newspapers or books.

This paper briefly outlines some of the properties of texts like [1], that allow readers to understand it in spite of its scruffiness; and some of the knowledge and mechanisms that seem to underlie readers' ability to understand such texts. A text-processing system called NOMAD is discussed which makes use of the theories described here to process scruffy text in the domain of everyday Navy messages. NOMAD's operation is based on the use of expectations during understanding, based both on knowledge of surface English and on world knowledge of the situation being described. These syntactic and semantic expectations can be used to aid naturally in the solution of a wide range of problems that arise in understanding both "scruffy" texts and pre-edited texts, such as figuring out unknown words from context, constraining the possible word-senses of words with multiple meanings (ambiguity), filling in missing words (ellipsis), and resolving unknown referents (anaphora).

2.0 Background: Tolerant text processing

2.1 FOUL-UP figured out unknown words from context

The FOUL-UP program (Figuring Out Unknown Lexemes in the Understanding Process) [Granger 1977] was the first program that could figure out meanings of unknown words encountered during text understanding. FOUL-UP was an attempt to model the corresponding human ability commonly known as "figuring out a word from context". FOUL-UP worked with the SAM system [Cullingford 1977], using the expectations generated by scripts [Schank and Abelson 1977] to restrict the possible meanings of a word, based on what object or action would have occurred in that position according to the script for the story.

For instance, consider the following excerpt from a newspaper report of a car accident:

[2] Friday, a car swerved off Route 69. The vehicle struck an embankment.

The word "embankment" was unknown to the SAM system, but it had encoded predictions about certain attributes of the expected conceptual object of the PROPEL action (the object that the vehicle struck); namely, that it would be a physical object, and would function as an "obstruction" in the vehicle-accident script. (In addition, the conceptual analyzer (ELI - [Riesbeck and Schank 1976]) had the expectation that the word in that sentence position would be a noun.)

Hence, when the unknown word was encountered, FOUL-UP would make use of those expected attributes to construct a memory entry for the word "embankment", indicating that it was a noun, a physical object, and an "obstruction" in vehicle-accident situations. It would then create a dictionary definition that the system would use from then on whenever the word was encountered in this context.

2.2 Blame Assignment in the NOMAD system

But even if the SAM system had known the word "embankment", it would not have been able to handle a less edited version of the story, such as this:

[3] Vehcle acc Rt69; car strck embankment; drivr dead one psngr inj; ser dmg to car full rpt frthcmng.

While human readers would have little difficulty understanding this text, no existing computer programs could do so.

The scope of this problem is wide; examples of texts that present "scruffy" difficulties to readers are completely unedited texts, such as messages composed in a hurry, with little or no re-writing, rough drafts, memos, transcripts of conversations, etc. Such texts may contain missing words, ad hoc abbreviations of words, poor syntax, confusing order of presentation of ideas, misspellings, lack of punctuation, etc. Even edited texts such as newspaper stories often contain misspellings, words unknown to the reader, and ambiguities; and even apparently very simple texts may contain alternative possible interpretations, which can cause a reader to construct erroneous initial inferences that must later be corrected (see [Granger 1980,1981]).

The following sections describe the NOMAD system, which incorporates FOUL-UP's abilities as well as significantly extended abilities to use syntactic and semantic expectations to resolve these difficulties, in the context of Naval messages.

3.0 How NOMAD Recognizes and Corrects Errors

3.1 Introduction

NOMAD incorporates ideas from, and builds on, earlier work on conceptual analysis (e.g., [Riesbeck and Schank 1976], [Birnbaum and Selfridge 1979]); situation and intention inference (e.g., [Cullingford 1977], [Wilensky 1978]); and English generation (e.g. [Goldman 1973], [McGuire 1980]). What differentiates NOMAD significantly from its predecessors are its error recognition and error correction abilities, which enable it to read texts more complex than those that can be handled by other text understanding systems.

We have so far identified the following five types of problems that occur often in scruffy unedited texts. Each problem is illustrated by an example from the domain of Navy messages. The next section will then describe how NOMAD deals with each type of error.

1. Unknown words (e.g., Enemy 'scudded' bombs at us. -- the verb is unknown to the system);

2. Missing subject, object, etc. of sentences. (e.g., Sighted enemy ship. Fired -- the actor who fired is not explicitly stated);

3. Missing sentence and clause boundaries. (e.g., Locked on opened fire. -- two actions, aiming and firing);

4. Missing situational (scripty) events. (e.g., Midway lost contact on Kashin. -- no previous contact mentioned);

5. Ambiguous word usage. (e.g., Returned bombs to Kashin. -- "returned" in the sense of retaliation after a previous attack, or "returned" in the sense of "peaceably delivered to"?)

When these problems arise in a message, NOMAD must first recognize what the problem is (which is often difficult to do), and then attempt to correct the error. These two processes are briefly described in the following sections.

3.2 Recognizing and correcting errors

For each of the above examples of problems encountered, NOMAD's method of recognizing and correcting the problem are described here, along with actual English input and output from NOMAD.

1. INPUT:
ENEMY SCUDDED BOMBS AT US.

Problem: Unknown word. The unknown word "scudded" is trivial to recognize, since it is the only word without a dictionary entry. Once it has been recognized, NOMAD checks it to see if it could be (a) a misspelling, (b) an abbreviation or (c) a regular verb-tense of some known word.
Solution: Use expectations to figure out word meaning from context. When the spelling checkers fail, a FOUL-UP mechanisms is called which uses knowledge of what actions can be done by an enemy actor, to a weapon object, directed at us. It infers that the action is probably a propel. Again, this is only an educated guess by the system, and may have to be corrected later on the basis of future information.

NOMAD OUTPUT:
An enemy ship fired bombs at our ship.

2. INPUT:
MIDWAY SIGHTED ENEMY. FIRED.

Problem: Missing subject and objects. 'Fired' builds a PROPEL, and expects a subject and objects to play the conceptual roles of ACTOR (who did the PROPELing), OBJECT (what got PROPELed) and RECIPIENT (who got PROPELed at). However, no surface subjects or objects are presented here.
Solution: Use expectations to fill in conceptual cases. NOMAD uses situational expectations from the known typical sequence of events in an "ATTACK" (which consists of a movement (PTRANS), a sighting (ATTEND) and firing (PROPEL)). Those expectations say (among other things) that the actor and recipient of the PROPEL will be the same as the actor and direction of the ATTEND, and that the OBJECT that got PROPELed will be some kind of projectile, which is not further specified here.

NOMAD OUTPUT:
We sighted an enemy ship. We fired at the ship.

3. INPUT:
LOCKED ON OPENED FIRE.

Problem: Missing sentence boundaries. NOMAD has no expectations for a new verb ("opened") to appear immediately after the completed clause "locked on". It tries but fails to connect "opened" to the phrase "locked on".
Solution: Assume the syntactic expectations failed because a clause boundary was not adequately marked in the message; assume such a boundary is there. NOMAD assumes that there may have been an intended sentence separation before "opened", since no expectations can account for the word in this sentence position. Hence, NOMAD saves "locked on" as one sentence, and continues to process the rest of the text as a new sentence.

NOMAD OUTPUT:
We aimed at an unknown object. We fired at the object.

4. INPUT:
LOST CONTACT ON ENEMY SHIP.

Problem: Missing event in event sequence. NOMAD's knowledge of the "Tracking" situation cannot understand a ship losing contact until some contact has been gained.
Solution: Use situational expectations to infer missing events. NOMAD assumes that the message implies the previous event of gaining contact with the enemy ship, based on the known sequence of events in the "Tracking" situation.

NOMAD OUTPUT:
We sighted an enemy ship. Then we lost radar or visual contact with the ship.

5. INPUT:
RETURNED BOMBS TO ENEMY SHIP.

Problem: Ambiguous interpretation of action. NOMAD cannot tell whether the action here is "returning" fire to the enemy, i.e., firing back at them (after they presumably had fired at us), or peaceably delivering bombs, with no firing implied.
Solution: Use expectations of probable goals of actors. NOMAD first interprets the sentence as "peaceably delivering" some bombs to the ship. However, NOMAD contains the knowledge that enemies do not give weapons, information, personnel, etc., to each other. Hence it attempts to find an alternative interpretation of the sentence, in this case finding the "returned fire" interpretation, which does not violate any of NOMAD's knowledge about goals. It then infers, as in the above example, that the enemy ship must have previously fired on us.

NOMAD OUTPUT:
An unknown enemy ship fired on us. Then we fired bombs at them.

4.0 Conclusions

The ability to understand text is dependent on the ability to understand what is being described in the text. Hence, a reader of, say, English text must have applicable knowledge of both the situations that may be described in texts (e.g., actions, states, sequences of events, goals, methods of achieving goals, etc.) and the the surface structures that appear in the language, i.e., the relations between the surface order of appearance of words and phrases, and their corresponding meaning structures.

The process of text understanding is the combined application of these knowledge sources as a reader proceeds through a text. This fact becomes clearest when we investigate the understanding of texts that present particular problems to a reader. Human understanding is inherently tolerant; people are naturally able to ignore many types of errors, omissions, poor constructions, etc., and get straight to the meaning of the text.

Our theories have tried to take this ability into account by including knowledge and mechanisms of error noticing and correcting as implicit parts of our process models of language understanding. The NOMAD system is the latest in a line of 'tolerant' language understanders, beginning with FOUL-UP, all based on the use of knowledge of syntax, semantics and pragmatics at all stages of the understanding process to cope with errors.

5.0 References

Birnbaum, L. and Selfridge, M. 1980. Conceptual Analysis of Natural Language, in R. Schank and C. Riesbeck, eds., Inside Computer Understanding. Lawrence Erlbaum Associates, Hillsdale, N.J.

Cullingford, R. 1977. Controlling Inferences in Story Understanding. Proceedings of the Fifth International Joint Conference on Artificial Intelligence (IJCAI), Cambridge, Mass.

DeJong, G. 1979. Skimming Stories in Real Time: An Experiment in Integrated Understanding. Ph.D. Thesis, Yale Computer Science Dept.

Goldman, N. 1973. The generation of English sentences from a deep conceptual base. Ph.D. Thesis, Stanford University.

Granger, R. 1977. FOUL-UP: A program that figures out meanings of words from context. Proceedings of the Fifth IJCAI, Cambridge, Mass.

Granger, R.H. 1980. When expectation fails: Towards a self-correcting inference system. In Proceedings of the First National Conference on Artificial Intelligence, Stanford University.

Granger, R.H. 1981. Directing and re-directing inference pursuit: Extra-textual influences on text interpretation. In Proceedings of the Seventh International Joint Conference on Artificial Intelligence (IJCAI), Vancouver, British Columbia.

Lebowitz, M. 1981. Generalization and Memory in an Integrated Understanding System. Computer Science Research Report 186, Yale University.

McGuire, R. 1980. Political Primaries and Words of Pain. Unpublished ms., Dept. of Computer Science, Yale University.

Riesbeck, C. and Schank, R. 1976. Comprehension by computer: Expectation-based analysis of sentences in context. Computer Science Research Report 78, Yale University.

Schank, R.C., and Abelson, R. 1977 Scripts, Plans, Goals and Understanding. Lawrence Erlbaum Associates, Hillsdale, N.J.

Wilensky, R. 1978. Understanding Goal-Based Stories. Computer Science Technical Report 140, Yale University.