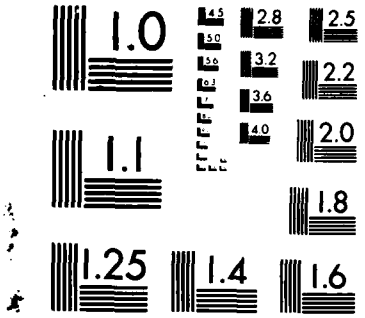
AD-A173 891 PROTEUS (PROTOTYPE TEXT UNDERSTANDING SYSTEM) AND PUNDIT (PROLOG UNDERSTA..(U) NEW YORK UNIV NY COURANT INST OF MATHEMATICAL SCIENCES R GRISHMAN ET AL. APR 86 1/1

UNCLASSIFIED N00014-85-K-0163 F/G 9/2 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

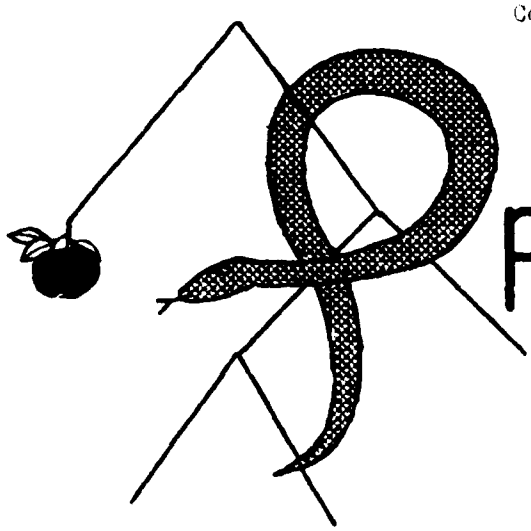
AD-A173 891

PROTEUS and PUNDIT: Research in Text Understanding

Ralph Grishman and Lynette Hirschman

PROTEUS Project Memorandum #1
April 1986

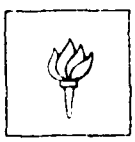
Contract N00014-85-K-0163



PROTEUS PROJECT

DTIC FILE COPY

NOV 3 1986



Department of Computer Science
Courant Institute of Mathematical Sciences
New York University

86

(12)

PROTEUS and PUNDIT: Research in Text Understanding

Ralph Grishman and Lynette Hirschman

PROTEUS Project Memorandum #1
April 1986

QUALITY INSPECTED

N0014-85-K-0163

DTIC
ELECTE
NOV 5 1986
A

Accession For
NTIS GRA&I
DTIC TAB
Unannounced
Justification
By
Distribution/
Availability Codes
Dist | Avail and/or Special
A-1

To appear in Computational Linguistics

Approved for public release;
distribution is unlimited.

PROTEUS and PUNDIT: RESEARCH IN TEXT UNDERSTANDING

at
the Department of Computer Science, New York University
and
System Development Corporation -- A Burroughs Company

prepared by
Ralph Grishman
(New York University)
and
Lynette Hirschman
(System Development Corporation)

1. Introduction

We are engaged in the development of systems capable of analyzing short narrative messages dealing with a limited domain and extracting the information contained in the narrative. These systems are initially being applied to messages describing equipment failure. This work is a joint effort of New York University and the System Development Corp. for the DARPA Strategic Computing Program. Our aim is to create a system reliable enough for use in an operational environment. This is a formidable task, both because the texts are unedited (and so contain various errors) and because the complexity of any real domain precludes us from assembling a "complete" collection of the relationships and domain knowledge relevant to understanding texts in the domain.

A number of laboratory prototypes have been developed for the analysis of short narratives. None of the systems we know about, however, is reliable enough for use in an operational environment (the possible exceptions are expectation-driven systems, which simply ignore anything deviating from these built-in expectations). Typical success rates reported are that 75-80% of sentences are correctly analyzed, and that many erroneous analyses pass the system undetected; this is not acceptable for most applications. We see the central task of the work to be described below as the construction of a substantially more reliable system for narrative analysis.

Our basic approach to increasing reliability will be to bring to bear on the analysis task as many different types of constraints as possible. These include constraints related to syntax, semantics, domain knowledge, and discourse structure. In order to be able to capture the detailed knowledge about the domain that is needed for correct message analysis, we are initially limiting ourselves to messages about one particular piece of equipment (the "starting air compressor"); if we are successful in this narrow domain, we intend to gradually broaden our system.

The risk with having a rich set of constraints is that many of the sentences will violate one constraint or another. These violations may arise from problems in the messages or in the knowledge base. On the one hand, the messages frequently contain typographical or grammatical errors (in addition to the systematic use of fragments, which can be accounted for by our grammar). On the other hand, it is unlikely that we will be able to build a "complete" model of domain knowledge; gaps in the knowledge base will lead to constraint violations for some sentences. To cope with these violations, we intend to develop a "forgiving" or flexible analyzer which will find a best analysis (one violating the fewest constraints) if no "perfect" analysis is possible. One aspect of this is the use of syntactic and semantic information on an equal footing in assembling an analysis, so that

neither a syntactic nor a semantic error would, by itself, block an analysis.

2. Application

This work is work is a component of the Fleet Command Center Battle Management Program (FCCBMP), which is part of the Strategic Computing Program. The FCCBMP has two natural language components: one for interactive natural language access, the other for message processing. The interactive component -- which is to provide access to a data base and multiple expert systems -- is being integrated by Bolt Beranek and Newman. The message processing component is being integrated as a joint effort of New York University and the System Development Corporation.

Much of the information received by the Fleet Command Center is in the form of messages. Some of these messages have a substantial natural language component. Consequently, natural language analysis is required if the information in these messages is to be recorded in a data base in a form usable by other programs. The specific class of messages which we are studying are CASREPs, which are reports of equipment failures on board ships. These messages contain a brief narrative, typically 3 to 10 sentences in length, describing the symptoms, diagnosis, and possibly the attempts at repair of the failure. A typical narrative is shown in Figure 1. The problems we face in analyzing these messages are similar to those in analyzing short messages and reports in other technical domains, and we therefore expect that the solutions we develop will be widely applicable.

3. Project organization

This work is a joint research effort of New York University and the System Development Corporation. NYU has principal responsibility for development of the domain knowledge base; SDC has principal responsibility for development of the flexible parser. The division of the other tasks is noted in the detailed component descriptions below. We will also be integrating work on the knowledge base being done by SRI, which is a component technology developer for the FCCBMP natural language work.

The work by NYU is being done in LISP (primarily in COMMON LISP), as is most of the Strategic Computing research. SDC is doing its development in PROLOG because Prolog provides a powerful framework for writing grammars; it also provides the inference engine necessary for knowledge structuring and reasoning about the discourse structures in text processing. This division will permit us to make some valuable comparisons between the LISP and PROLOG development environments, and of the resulting systems.

The system being developed in LISP by NYU is called PROTEUS (PROtotype TExt Understanding System) (Grishman *et al.*, 1986); the SDC system is called PUNDIT (Prolog UNDerstander of Integrated Text) (Palmer *et al.* 1986). Notwithstanding the difference in implementation languages, we have tried to maintain a high level of compatability between the two systems. We use essentially the same grammar and have agreed on common representations for the output of the syntactic analyzer (the regularized syntactic structure) and the output of the semantic analyzer. This commonality makes is possible assign primary responsibility for the design of a component to one group, and then to take the design developed for one system and port it to the other in a straightforward way.

We are currently developing baseline systems which incorporate substantial domain knowledge but use a traditional sequential processing organization. When these systems are complete, we will begin experimenting with flexible parsing algorithms. The systems currently being developed (Figure 2) process input in the following stages: lexical look-up, parsing, syntactic regularization, semantic analysis, integration with the domain knowledge representation, and discourse analysis. These components, and other tasks which are part of our research program, are described individually below.

4. System Components

4.1. Lexicon (SDC + NYU)

The lexicon consists of a modified version of the lexicon of the NYU Linguistic String Project, with words classified as to part of speech and subclassified for various grammatical properties (e.g., verbs and adjectives are subclassified for their complement types).

4.2. Lexical acquisition (SDC)

The message vocabulary is large and will grow steadily as the system is modified to handle a wider range of equipment; several measures are planned to manage the growth of the lexicon. An interactive lexical entry program has been developed to facilitate adding words to the dictionary. Special constructions such as dates, times, and part numbers are processed using a small definite clause grammar defining special shapes. Future plans include addition of a component to use morphological analysis and selectional patterns to aid in classification of new lexical items.

4.3. Syntax analysis (NYU + SDC)

4.3.1. Grammar

The syntactic component uses a grammar of BNF definitions with associated restrictions that enforce context-sensitive constraints on the parse. This grammar is generally modelled after that developed by the NYU Linguistic String Project (Sager 1981). The grammar has been expanded to cover the fragmentary constructions and complex noun phrases characteristic of the Navy message domain. A wide range of conjunction types is parsed by a set of conjunction rules which are automatically generated by metarules (Hirschman, in press). To serve as an interface between the syntactic and semantic components, an additional set of rules produces a normalized intermediate representation of the syntax.

4.3.2. Top-Down Parsers

Two top-down parsers have been implemented using the common grammar just described. In each case, the analyzer applies the BNF definitions and their associated constraints to produce explicit surface structure parses of the input; the analyzer also invokes the regularization rules which produce the normalized intermediate representation.

In the NYU (LISP-based) system the basic algorithm is a chart parser, which provides goal- directed analysis along with the recording (for possible re-use) of all intermediate goals tried. The context sensitive constraints are expressed in a version of Restriction Language (Sager 1975) which is compiled into LISP. The SDC (PROLOG-based) system uses a top-down left-to-right Prolog implementation of a version of the restriction grammar (Hirschman and Puder 1982).

4.4. Flexible Analyzer (SDC)

A major research focus for SDC during the first two years will be to produce a flexible analyzer that integrates application of syntactic and semantic constraints. The flexible analyzer will focus more quickly on the correct analysis and will have recovery strategies to prevent syntactic analysis from becoming a bottleneck for subsequent processing.

4.5. Semantic Analysis

The task of the semantic analyzer is to transform the regularized syntactic analysis into a "logical form", which involves identifiers of specific components within the equipment,

predicates describing states and events involving the equipment, and higher-order predicates capturing the syntactically-expressed time and causal relations. Roughly speaking, the clauses from the syntactic analysis map into states and events, while the noun phrases map into particular objects (there are several exceptions, including nominalizations and adjectives of state, such as "broken valve"). Accordingly, the semantic analysis is divided into two major parts, clause semantics and noun phrases semantics. In addition to these two main parts, a time analysis component captures the time information which can be extracted from the input.

4.5.1. Clause semantics (SDC)

Semantic analysis of clauses is performed by Inference Driven Semantic Analysis (Palmer 1985), which analyzes verbs into their component meanings and fills their semantic roles, producing a semantic representation in predicate form. This representation includes information normally found in a case-frame representation, but is more detailed. The task of filling in the semantic roles is used to integrate the noun phrase analysis (described in the next section) with the clausal semantic analysis. In particular, the selection restriction information on the roles can be used to reject inappropriate referents for noun phrases.

The semantics also provides a filtering function, by checking selectional constraints on verbs and their arguments. The selectional constraints draw on domain knowledge for type and component information, as well as for information about possible relationships between objects in the domain. This function is currently used to accept or reject a completed parse. The goal for the flexible analyzer is to apply selectional filtering compositionally to partial syntactic analyses to rule out semantically unacceptable phrases as soon as they are generated in the parse.

4.5.2. Noun phrase semantics (SDC + NYU)

A noun phrase resolution component determines the reference of noun phrases, drawing on two sources: a detailed equipment model, and cumulative information regarding referents in previous sentences. SDC has concentrated on the role of prior discourse, and has developed a procedure which handles a wide variety of noun phrase types, including pronouns and missing noun phrases, using a focusing algorithm based on surface syntactic structure (Dahl, submitted for publication). NYU, as part of its work on the domain model, has developed a procedure which can identify a component in the model from any of the noun phrases which can name that component (Ksiezyk and Grishman 1986). After further development, these procedures will be integrated into a comprehensive noun phrase semantic analyzer.

4.5.3. Time analysis (SDC)

SDC has started to develop a module to process time information. Sources of time information include verb tense, adverbial time expressions, prepositional phrases, co-ordinate and subordinate conjunctions. These are all mapped into a small set of predicates expressing partial information about the time relations among the states and events in the message.

4.6. Domain model (NYU)

The domain model captures the detailed information about the general class of equipment, and about the specific pieces of equipment involved in the messages, which is needed in order to fully understand the messages. The model integrates part/whole information, type/instance links, and functional information about the various components (Ksiezyk and Grishman 1986).

The knowledge base performs several functions: It provides the domain-specific constraints needed for the semantics to select the correct arguments for a predicate, to correctly attach modifiers to noun phrases, and for noun phrase semantics to identify the correct referent for a phrase. It provides the prototype information structures which are instantiated in order to record the information in a particular message. It provides the information on equipment structure and function which is used by the discourse rules in establishing probable causal links between the sentences. And finally, associated with the components in the knowledge base are procedures for graphically displaying the status of the equipment as the message is interpreted.

These functions are performed by a large network of frames implemented using the Symbolics Zetalisp flavors system.

4.7. Discourse analysis (NYU)

The semantic analyzer generates separate semantic representations for the individual sentences of the message. For many applications it is important to establish the (normally implicit) intersentential relationships between the setences. This is performed by a set of inference rules which (using the domain model) identify plausible causal and enabling relationships among the sentences. These relationships, once established, can serve to resolve some semantic ambiguities. They can also supplement the time information extracted during semantic analysis and thus clarify the temporal relation among the sentences.

4.8. Diagnostics (NYU)

The diagnostic procedures are intended to localize the cause of failure of the analysis and provide meaningful feedback when some domain-specific constraint has been violated. We are initially concentrating on violations of local (selectional) constraints, and have built a small component for diagnosing such violations and suggesting acceptable sentence forms; later work will study more global discourse constraints.

REFERENCES

Dahl, Deborah A. (submitted for publication). Reference Resolution in the PUNDIT Text Understanding System.

Grishman, Ralph, Tomasz Ksiezyk, and Ngo Thanh Nhan (1986). Model-based Analysis of Messages about Equipment. PROTEUS Project Memorandum #2, Computer Science Department, New York University.

Hirschman, Lynette and Karl Puder (1982). Restriction Grammar in Prolog. *Proc. of the First International Conf. on Logic Programming, Marseilles.*

Hirschman, Lynette. (in press). Constraints on Noun Phrase Conjunction: A Domain Independent Mechanism. *Journal of Logic Programming.*

Ksiezyk, Tomasz, and Ralph Grishman (1986). An Equipment Model and its Role in the Interpretation of Nominal Compounds. PROTEUS Project Memorandum #3, Computer Science Department, New York University.

Palmer, Martha S. (1985) Driving Semantics for a Limited Domain. Ph.D. thesis. University of Edinburgh.

Palmer, Martha, Deborah Dahl, Rebecca Schiffman, Lynette Hirschman, Marcia Linebarger, and John Dowding. (1986) Recovering Implicit Information. To appear

in *Proc. 24th Annl. Meeting Assn. Computational Linguistics*.

Sager, Naomi and Ralph Grishman (1975). The Restriction Language for Computer Grammars of Natural Language. *Comm. of the ACM*, vol. 18, pp. 390-400.

Sager, Naomi (1981). *Natural Language Information Processing: A Computer Grammar of English and its Applications*. Addison-Wesley, Reading, MA.

A Sample CASREP

about a SAC (Starting Air Compressor)

DURING NORMAL START CYCLE OF 1A GAS TURBINE, APPROX 90 SEC AFTER CLUTCH ENGAGEMENT, LOW LUBE OIL AND FAIL TO ENGAGE ALARM WERE RECEIVED ON THE ACC. (ALL CONDITIONS WERE NORMAL INITIALLY). SAC WAS REMOVED AND METAL CHUNKS FOUND IN OIL PAN. LUBE OIL PUMP WAS REMOVED AND WAS FOUND TO BE SEIZED. DRIVEN GEAR WAS SHEARED ON PUMP SHAFT.

Figure 1

PROTEUS/PUNDIT SYSTEM STRUCTURE

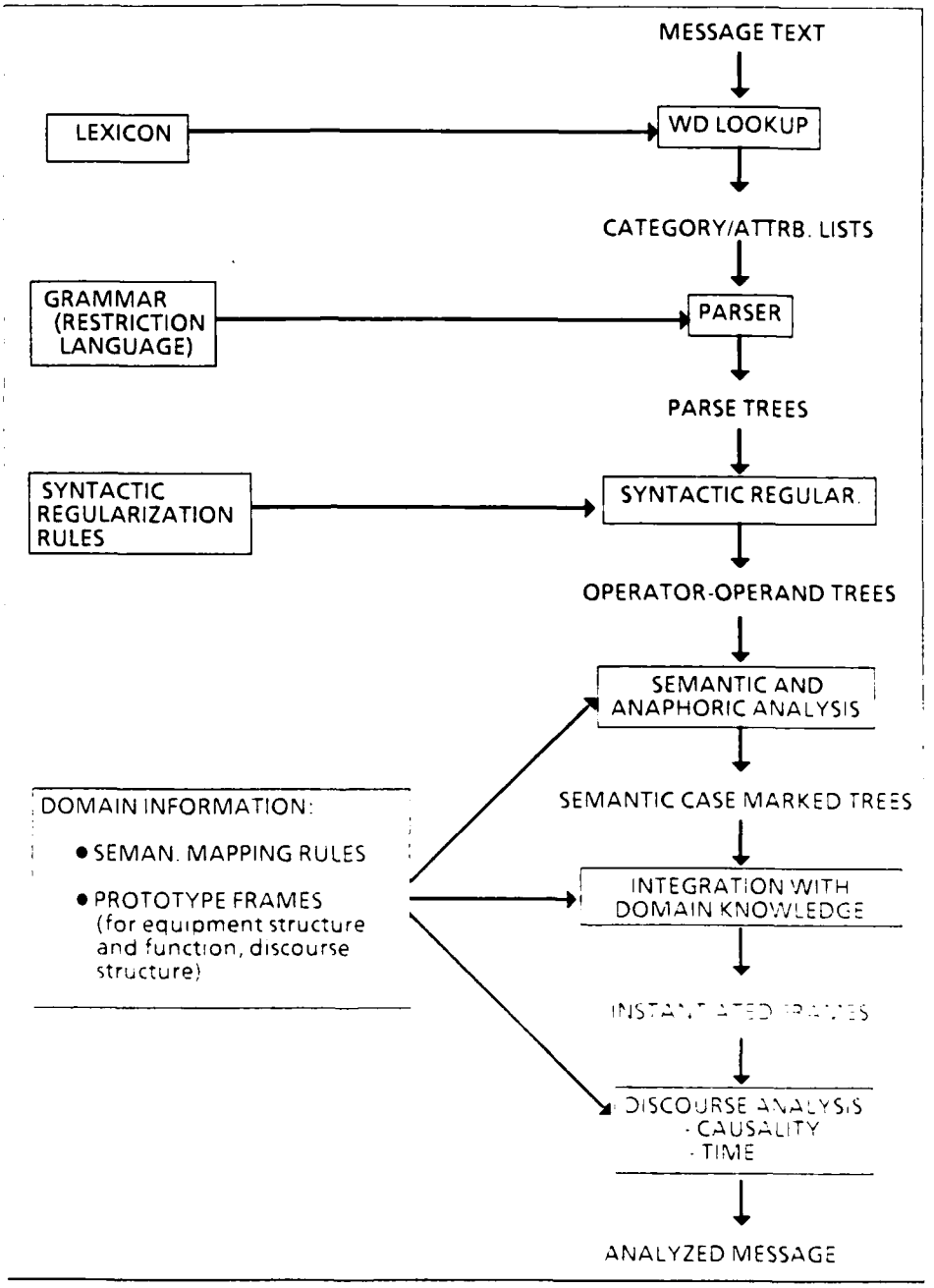


Figure 2

END

12-86

DTIC