A PRODUCTION RULE SYSTEM FOR MESSAGE SUMMARIZATION

*Elaine Marsh and Henry Hamburger**

Navy Center for Applied Research
in Artificial Intelligence
Naval Research Laboratory - Code 7510
Washington, D.C. 20375
*Also at George Mason University

Ralph Grishman

Department of
Computer Science
New York University
New York, New York 10012

ABSTRACT

In summarizing a message, it is necessary to access knowledge about linguistic relations, subject matter knowledge about the domain of discourse, and knowledge about the user's goals for the summary. This paper investigates the feasibility of integrating these knowledge sources by using computational linguistic and expert system techniques to generate one-line summaries from the narrative content of a class of Navy messages. For deriving a knowledge representation of the narrative content, we have adapted an approach developed by Sager *et al.* at New York University. This approach, called *information formatting,* uses an explicit grammar of English and a classification of the semantic relationships within the domain to derive a tabular representation of the information in a message narrative. A production system, written in OPS5, then interprets the information in the table and automatically generates a summary line. The use of a production rule system provides insight into the mechanisms of summarization. A comparison of computer-generated summaries with those obtained manually showed good agreement, indicating that it is possible to automatically process message narrative and generate appropriate, and ultimately useful, summaries.

INTRODUCTION

Behavior modeled in expert systems has generally been held distinct from that modeled in natural language understanding systems. Attempts at practical expert systems have been directed toward design [McDermott 1980], diagnosis [Shortliffe 1976], and interpretation [Buchanan 1978], among others. Practical systems for natural language understanding have concentrated largely on database interfaces [Grosz 1983, Ginsparg 1983, Grishman 1983] and database creation [Sager 1978]. In this paper we investigate the feasibility of integrating techniques from computational linguistics and expert system technology to summarize a set of Navy equipment failure messages called CASREPs (casualty reports). A natural language analysis procedure automatically generates a tabular representation of the information contained in message narrative. A production rule system then interprets the tabular representation and identifies a clause that is appropriate as a message summary. We have chosen to use a production system for a natural language application because it facilitates understanding and modification of the system. More important for research purposes, a production system makes the operations involved in summarization explicit and, thus, can provide insight into the general problem of summarization.

Summarization can be approached at several different levels. Typically, strategies for summarization have taken a single-level approach. Summaries of stories have been derived at the high level of conceptual representation. Structural features of a graph reveal the central concepts of a story [Lehnert 1980]. Goal-directed summaries have also been investigated in some detail [Fum 1982]. We, on the other hand, have taken a multi-level approach, incorporating several sources of knowledge in the linguistic analysis and production rule system. This permits us to investigate not only the requirements of individual knowledge sources, but also their interactions.

NATURAL LANGUAGE PROCESSING

Each CASREP message contains a set of structured (i.e. *pro forma*) fields and a narrative describing the equipment failures. These narratives typically consist of two to twelve sentences and sentence fragments.

The central task of narrative analysis is the extraction and representation of information contained in narrative portions of a message. This task is difficult because the structure of the information, and often much of the information itself, is implicit in the narrative. Several formalisms, such as *scripts* and *frames*, have been developed to describe such information and have been used in text analysis [Schank 1977; Montgomery 1983]. We are using an approach called *information formatting* that was developed at New York University for the representation of the information in medical narratives [Sager 1978, Hirschman 1982]. In simple terms, an information format is a large table, with one column for each type of information that can occur in a class of texts and one row for each sentence or clause in the text. It is derived through a distributional analysis of sample texts. The narrative is automatically transformed into a series of entries in the information format table. This procedure involves three stages of processing: (1) parsing, (2) syntactic regularization, and (3) mapping into the information format.

First the text sentences are parsed using a top-down parser and the broad-coverage Linguistic String Project English grammar [Sager 1981] extended to handle the sentence fragments and special sublanguage constructions (e.g. date expressions, such as *NLT 292300 Z SEP 82*) that appear in these messages. The grammar consists of a set of context-free definitions augmented by grammatical restrictions. It also uses a Navy sublanguage lexicon that classifies words according to their major parts of speech (e.g. noun, verb, adjective), as well as their special subfield classes (e.g. PART, FUNCTION, SIGNAL), and certain English syntactic subclasses. The parsing procedure identifies the grammatical relations that hold among parts of the sentence, principally subject-verb-object relations and modifier-host relations.

The syntactic regularization component utilizes the same machinery as the parser, augmented by standard transformational operations. The principal function of the regularization component is to reduce the variety of syntactic structures and word forms to be processed, without altering the information content of the sentences, thereby simplifying the subsequent mapping into the information format. Regularization includes: (1) standardization into subject-verb-object word order, e.g. passive to active; (2) expansion of conjoined phrases into conjoined assertions; (3) reduction of words to "canonical form" plus information marker(s); (4) filling in of certain omitted or reduced forms of information.

The third stage of processing moves the phrases in the syntactically regularized parse trees into their appropriate format columns. It involves two steps: (1) identifying connectives and (2) mapping into the information format. A connective word indicates a causal, conjunctional, or time relation between the two clauses it connects. The connective is mapped into the CONNective column of the format table; arguments of the connective are mapped into separate format rows, and their words are mapped into the appropriate format columns. The mapping process is controlled in a large part by the sublanguage (semantic) word classes associated with each word in the lexicon. In general, the formatting procedure is straightforward because most word classes are in a one-to-one correspondence with a particular format column.

The production system for message summarization operates on the information format that is generated for each message.

PRODUCTION RULE SYSTEM FOR SUMMARIZATION

We have implemented prototype knowledge bases for two application areas: dissemination and summary generation [Marsh 1984]. While the dissemination application relies on information obtained from both *pro forma* and narrative data sets of a message, summary generation is based entirely on information contained in narrative portions of the messages. Such summaries, which up to now have been generated by hand, are used to detect patterns of failure for particular types of equipment. This failure information is crucial to decision-makers who procure equipment for new and existing ships. Typically, the manually derived summary consists of a single clause, extracted from the sentences of text. Only rarely is a summary generated from material not explicitly stated in the narrative. The single line summary results in a five- to ten-fold reduction of material. Clearly, the sharp reduction in reading material can ease the decision-making process, provided that the key information from the report regularly finds its way into the summary.

Our current system consists of a set of productions, implemented in a Lisp-based version of the OPS5 production system programming language. OPS5 permits the assignment of attributes and numerical values, or scores, to the working memory elements, and our system takes advantage of this. Productions operate on an initial database of working memory elements that includes data from the the information formats and identify the crucial clause that will be used for the summary. Criteria for production rules are based on the manual summarization that is currently performed.

Several types of knowledge are required for message summarization. Knowledge of the possible relationships is reflected in the initial choice of what fields are available in the format system devised for the domain. This is represented by the columns of each message's information format table. Additional domain knowledge and knowledge of the nature of the application are embodied in the production rules of the expert system.

Each production rule incorporates one of three different types of knowledge necessary for summarization. The first type reflects an understanding of the subject matter of the equipment failure reports. These production rules assign semantic attributes or categories to working memory elements by explicitly specifying these words in a list in the rule. For example, the working memory element containing the word *inhibit* is assigned a category IMPAIR. Elements indicating a bad status (e.g. broken, corroded, failure, malfunction, etc.) have the category BAD assigned and so on. Other category assignment rules are concerned with level of generality, flagging equipment failures at the assembly level, and not at the more detailed part or more general system level, since assemblies are most important to the summary.

Other production rules are based on general principles of summarization, and these rules are typically inferencing rules. These identify causal relationships among working memory elements and may add information to the data base in the form of new working elements. We will see an example of this type below.

Finally, the end use that will be made of the summaries is also a guiding factor in some of the productions. To guide future equipment specification and procurement, one must know not only what went wrong and how often, but also why. Format rows that contain such information are identified as being more important by having the score of the row boosted. For example, causality is important to the summaries. Once a causal relationship is identified, the row specifying the 'cause' has its score boosted. Taken together, the productions are attentive to such matters as malfunction, causality, investigative action, uncertainty, and level of generality. In addition, the system has rules excluding from summaries format rows containing very general statements. For instance, universal quantification and mention of the top level in a *part-of* tree betray a clause that is too general for a summary line.

Summarization proceeds in three stages: (i) inferencing, (ii) scoring the format rows for their importance, and (iii) selection of the appropriate format row as the summary. First, inferences are drawn by a set of production rules. For example, the presence of one of the words in the IMPAIR category triggers an inferencing rule. If part1 impairs part2, we can infer that part1 causes part2 to be bad, and we can also infer that part1 is bad. A set of production rules, summarized as rules (1) and (2) below, operate on the format lines to draw such inferences. The production rule in (1) infers that the second argument (part2) of CONN is bad.

(1) if both (a) CONN contains an 'impair' word
    and (b) the STATUS column of the 2nd argument of
        CONN [the connective] is empty
    then both (c) fill the STATUS column of the
        2nd argument with 'bad'
        and (d) assign the word in CONN
        the attribute 'cause'.

For example, in Table 1, the connective word *inhibit* has been mapped by the formatting procedure into the CONN column, connecting two format rows, its first argument, *APC-PPC circuit*, a PART, and its second, *PA driver*, also a PART. Both rows have the PART column of the format filled.

CONN	PART	STATUS
	APC-PPC circuit	
inhibit		
	PA driver	

Table 1: Simplified information format for the sentence: *APC-PPC is inhibiting PA driver*

By a previous production rule, the *inhibit* has been categorized in the class of impairment verbs. Rule (1) replaces impairment by a format version of "cause to be bad." Specifically, the verb *inhibit* in the CONN column gets assigned the attribute 'cause'. Since the STATUS column of the second argument is empty, *bad* is inserted into that STATUS column. Thus, it is inferred that the PA driver is bad because it has been impaired.

Another production rule, summarized as (2), infers that the STATUS column of the first argument (part1) of CONN is also 'bad' and inserts *bad* into the STATUS column since it has caused something else to be bad.

(2) if both (a) CONN has the attribute 'cause'
and (b) the STATUS of the first argument of CONN is empty
and (c) the STATUS of the second argument of CONN is 'bad'
then (d) insert 'bad' into the empty STATUS column.

In our example Table 1, 'inhibit' in the CONNective column has been assigned the attribute 'cause', and the STATUS of *APC-PPC circuit* is empty. The STATUS of the *PA driver* contains 'bad', by rule (1). So 'bad' is inserted into the STATUS column of the first argument, yielding *APC-PPC circuit bad*.

The second stage of the summarization system rates the format rows for their importance to the summary. When it comes time to score the various formats to determines the most appropriate one for the summary, since "bad" is a member of the class of words signifying malfunction, it will cause both arguments of *inhibit* to be promoted in importance. An additional scoring increment will accrue to the first argument but not the second because it is a cause rather than an effect. Another rule increments a format row referring to an assembly (a mid-level component), since such a format is more revealing than a format containing a statement about about a whole unit or an individual part (such as a transistor). For example, *circuit*, the head of the PART phrase of the first argument is identified as belonging to a class of components at the assembly level. As a result, the score of the row containing *APC-PPC circuit bad* is incremented again.

The third and final stage of summarization is to select the format row or rows with the highest rating. As a result of the various production rule actions, the winning format row is "PART: *APC-PPC circuit*; STATUS: *bad*." While other format rows may also have positive scores, only the row with the highest score is selected. The system does not preclude selecting several format rows if they have equally high scores.

IMPLEMENTATION

The LSP parser is implemented in about 15,000 lines of Fortran 77 code. The parser runs on a DEC VAX 11/780 under the UNIX and VMS operating systems and requires 2 megabytes of virtual memory when executing, of which two-thirds is list space for holding the grammar, dictionary entries, etc. The English grammar, regularization component, and information formatting components are written in Restriction Language, a special language developed for writing natural language grammars [Sager 1975]. The dissemination and summary generation applications programs are written using the OPS5 production system. In total, there are 63 production rules in the applications programs.

EXPERIMENTAL RESULTS

The purpose of this experiment was to test the feasibility of automatically summarizing narrative text in Navy equipment failure messages using techniques of computational linguistics and artificial intelligence. Computer-generated results were compared to those obtained by manual summarization procedures to evaluate the performance of the system. The manual summaries were prepared independently of our experiment by experts who routinely summarize such messages.

Since both the natural language processing components and the applications programs were under development while this experiment was being carried out, 12 casualty reports were used for debugging the programs. Subsequently, 12 other reports were used for the computer-human comparison. For an appropriate summary line to be generated, it is necessary that 100% of the sentences in a text be processed correctly by the natural language procedures. The natural language analysis procedures processed 100% of the sentences contained in the documents; this percentage includes 9 sentences (25%) that were paraphrased and rerun because they were not correctly processed on their first run. Paraphrasing these sentences brought the total number of sentences from 36 to 38. The sentences were paraphrased to expedite processing since the major purpose of running the messages was to investigate methods of summarization and not the performance of the natural language processing system. 70 format lines were generated from 38 sentences in 12 messages.

The computer-generated results of the summarization program compare favorably to those obtained manually. Figure 1 shows a comparison of the two sets of results for the 12 test documents. The discrepancies between the computer-generated results and the manual results are summarized in Figure 2.

Doc.	Machine *# format rows*	Manual *# sentences*	Agreement *Machine/Manual*
1.	1	1	1/1
2.	1	1	1/1
3.	1	1	1/1
4.	1	1	0/1
5.	1	2	1/2
6.	2	1	1/1
7.	1	1	1/1
8.	2	1	1/1
9.	1	1	0/1
10.	1	2	1/2
11.	1	1	1/1
12.	1	2	1/2
	14	15	10/15

Fig. 1: Comparison of machine and manual summary results

#	Discrepancy	Doc.
1	word not included in category list	4
1	second manual summary not about bad-status	5
1	second manual summary not in narrative text	10
2	different summaries generated	9,12

Fig. 2: Analysis of machine and manual summary results

Agreement between machine and manual summaries is obtained when the text contained in the format row selected by the automatic procedure agrees with the text in the manually generated sentences. The discrepancies in the *Agreement* column of Figure 1, as specified in Figure 2, can be categorized as follows. One (message 4) is the result of a failure to enter a word on a category list in the production rule system. As a result, the word was not categorized as a BAD-STATUS, and the score of its format row was not correspondingly boosted. Two errors (messages 5 and 10) were due to the program selecting one format line, although manual generation produced two sentences. In the first case (message 5), the additional text in the manual summary did not concern a description of a bad status. Rather it was a description of a good function status (i.e. *Drive shaft was found to rotate freely.*). In message 10, the extra manual summary consisted entirely of text that was not contained in the message narrative. Our system does not automatically generate text, nor could it have made the inferences necessary to do so. In both these cases, however, the line that the program selected agreed with one of the manual summaries.

The most significant discrepancies (a total of 2) were caused by the system selecting more specific causal information than was indicated in the manual summary. In message 9, which contains the sentence *Loss of lube oil pressure when start air compressor engaged for operation is due to wiped bearing* the manual summary line generated was *Loss of LO pressure*, while the system selected the more specific information that indicated the cause of the casualty, i.e. *wiped bearing*. Similarly, in message 12, the system selected the line *low output air pressure* from the assertion *low output air pressure resulting in slow gas turbine starts* since it indicated a cause. The program did not identify the second part of the manual summary because its score was not as high as that of the cause *low output air pressure*. However, its score was the second highest for that message. This suggests that it may be more appropriate to select all the summary lines in some kind of score window rather than only those lines that have the highest score.

In two cases (messages 6 and 8), the system generated two summary texts, although the manual summary consisted of only one sentence. Two summary lines were selected because both had equally high scores. Nonetheless, one of the two summaries was also the manual summary.

In conclusion, the summarization system was able to identify the same summary line as the manual summary 10/15 times (66.6%). For 10 out of 12 messages, the summarization system selected at least one of the same summary lines as the manual generation produced. For two messages, the system was not able to match the manual summary, in one case, because the crucial status word was not in the appropriate list in the production rule system and, in a second case, because the automatic procedure identified the more specific causal agent.

CONCLUSION

The results of our work are quite promising and represent a successful first step towards demonstrating the feasibility of integrating computational linguistic and expert system techniques. We recognize that much remains to be done before we have an operational system. Our work up to now has pointed to several areas that require further development.

Refinement of the semantic representation. Our current information format was developed from a limited corpus of 38 messages, including those in the test set. Even within that corpus not all types of information have been captured - for example, modes of operation, relations between parts and signals, and relations and actions involving more than one part. Some of this information has been incorporated into the expert system. For example, part-assembly-system information has been encoded as a categorization rule. However it is clear that enrichment of our semantic representation is a high priority. We are considering the use of some external knowledge sources to obtain this information. One possibility is to access machine-readable listings of Navy equipment.

Intersentential processing. Our current implementation does almost no intersentential processing. This has proved marginally adequate for our current applications, but clearly needs to be remedied in the long run. One aspect of this processing is the capture of information that is implicit in the text. This includes missing arguments (subject and objects of verbs) and anaphors (e.g. pronouns) that can be reconstructed from prior discourse (earlier format lines); such processing is part of the information formatting procedure for medical records [Hirschman 1981]. It should also include reconstruction of some of the implicit causal connections The reconstruction of the connections will require substantial domain knowledge, of equipment-part and equipment-function relations, as well as "scriptal" knowledge of typical event sequences (e.g. failure - diagnosis - repair).

ACKNOWLEDGMENTS

This research was supported by the Office of Naval Research and the Office of Naval Technology PE-62721N. The authors gratefully acknowledge the efforts of Judith Froscher and Joan Bachenko in processing the first set of messages and providing the specifications for our dissemination system.

REFERENCES

[Buchanan 1978] Buchanan, B.G., Feigenbaum, E.A. DENDRAL and Meta-DENDRAL: their applications dimension, *Artificial Intelligence* **11**, 5-24.

[Fum 1982] Fum, D., Guida, G., Tasso, C. Forward and backward reasoning in automatic abstracting. *COLING 82* (J. Horecky (ed)). North Holland Publishing Company.

[Ginsparg 1983] Ginsparg, J.M. A robust portable natural language data base interface. *Proceedings of the Conference on Applied Natural Language Processing*, ACL.

[Grosz 1983] Grosz, B.J. TEAM, a transportable natural language interface system.*Proceedings of the Conference on Applied Natural Language Processing*, ACL.

[Grishman 1983] Grishman, R., Hirschman, L., Friedman, C. Isolating domain dependencies in natural language interfaces.*Proceedings of the Conference on Applied Natural Language Processing*, ACL.

[Hirschman 1982] Hirschman, L., Sager, N. Automatic information formatting of a medical sublanguage. In *Sublanguage: Studies of Language in Restricted Domains* (Kitttedge and Lehrberger, eds). Walter de Gruyter, Berlin.

[Hirschman 1981] Hirschman, L., Story, G., Marsh, E., Lyman, M., Sager, N. An experiment in automated health care evaluation from narrative medical records. *Computers and Biomedical Research* **14**, 447-463.

[Lehnert 1980] Lehnert, W. Narrative text summarization. *AAAI-80 Proceedings*: 337-339.

[Marsh 1984] Marsh, E., Froscher, J., Grishman, R., Hamburger, H., Bachenko, J. Automatic processing of Navy message narrative. NCARAI Internal Report.

[McDermott 1980] McDermott, J. R1: A rule-based configurer of computer systems. Carnegie-Mellon University, Department of Computer Science Report CMU-CS-80-119, April 1980.

[Montgomery 1983] Montgomery, C. Distinguishing fact from opinion and events from meta-events. *Proceedings of Conference on Applied Natural Language Processing*, 55-61, Assn. for Computational Linguistics.

[Sager 1975] Sager, N., Grishman, R. The Restriction Language for computer grammars of natural language. *Communications of the ACM*,**18**, 390.

[Sager 1978] Sager, N. Natural language information formatting: The automatic conversion of texts to a structured data base. In *Advances in Computers***17** (M.C. Yovits, ed), Academic Press.

[Sager 1981] Sager, N. *Natural Language Information Processing.* Addison-Wesley.

[Schank 1977] Schank, R., Abelson, R. *Scripts, Plans, Goals, and Understanding.* Lawrence Erlbaum Associates.

[Shortliffe 1976] Shortliffe, E.H. *Computer-based medical consultations: MYCIN.* American Elsevier.