USING A MATCHER TO MAKE AN EXPERT CONSULTATION SYSTEM BEHAVE INTELLIGENTLY*

Rene' Reboh
Artificial Intelligence Center
SRI International, Menlo Park, CA 94025

ABSTRACT

This paper describes how even a simple matcher, if it can detect simple relationships between statements in the knowledge base, will support many features that will make a knowledge-based consultation system appear to behave more intelligently. We describe three features that are useful during the knowledge acquisition phase (involving the building and testing of the knowledge base), and four features that are of assistance during a consultation. Although these features are described in terms of the Prospector environment [2], it will be clear to the reader how these features can be transferred to other environments.

I INTRODUCTION

Partial-matching (also referred to as interference-matching, correspondence-mapping, ...) touches many issues of representation and efficiency in many AI systems [3]. Its role in Prospector is significant because it is involved in many aspects of consultation and knowledge acquisition. Given any two statements S1 and S2 of the knowledge base, the Semantic Network Matcher of Prospector determines which of the following situations applies:

S1 and S2 are identical (S1 = S2)
S1 is a restriction of S2 ($S1 \subset S2$),
(or S2 is a generalization of S1 ($S2 \supset S1$))
S1 and S2 are disjoint statements ($S1 \cap S2 = \phi$)
S1 overlaps S2 (otherwise)

For instance, suppose the knowledge base contains the following statements:

S1: "rhyolite is present"
S2: "a rhyolite plug is present"
S3: "an igneous intrusive is present"
S4: "rhyolite or dacite is present"
S5: "pyrite is present"

As these statements are being added to the knowledge base, the Matcher will conclude that:

S2 is a restriction of S1,
S2 is a restriction of S3
(rhyolite is an igneous rock and a plug is a special kind of intrusive),
S1 and S3 overlap (rhyolite is an igneous rock, but need not be an intrusive),
S1 is a restriction of S4,
S2 is a restriction of S4
(transitivity from the first and third conclusion),
S3 and S4 overlap,
S5 is disjoint from S1, S2, S3 and S4.

A detailed description of how the Matcher operates in the Prospector environment can be found in [6]. We mention briefly here that the Matcher views each statement as a set of constraints corresponding to a set of assertions about the existence of physical entities or processes and their attributes. In Prospector, partitioned semantic networks [5] are used to represent statements in the knowledge base whereby these assertions are expressed in terms of relations and entries in taxonomies of the domain of application (in this case geology).

Let us examine some of the features of a knowledge-based system that can be supported by such a Matcher.

---

* This work was supported by the Office of Resource Analysis of the U. S. Geological Survey under Contract No. 14-08-0001-15985. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author, and do not necessarily reflect the views of the U. S. G. S.

II USE OF THE MATCHER IN KNOWLEDGE ACQUISITION

A. Aid in Maintaining Probabilistic Consistency of the Inference Network

The knowledge bases of many expert systems are organized as explicit or implicit networks of statements connected by rules or logical constructs. Because such networks provide the framework for judgmental reasoning, various numerical values, such as probabilities or certainties, are often maintained in them. A major concern of expert systems is the difficulty of keeping the knowledge base error free and consistent in form and content as it grows.

Let us examine how the Matcher assists Prospector in maintaining probabilistic consistency in the case where S1 is the most recently entered statement, and S2, which already exists in the knowledge base, is a restriction of S1.

(a) Because S2 is a restriction of S1, the probability of S2 can never exceed that of S1. In particular, if the prior probabilities supplied by the domain specialist (DS) do not satisfy this constraint, the Matcher will detect the violation and a correction will be required. Thus, before a consultation begins, we can assume that $P(S2) < P(S1)$.

(b) Unfortunately, even though all the probabilistic constraints are initially satisfied, the probability changes that follow from the use of inference rules may not maintain them. For example, if S1 and S2 are the hypotheses (right-hand sides) of two rules E1 --> S1 and E2 --> S2, and if the evidence (left-hand side) E1 is sufficiently unfavorable for S1, we may have $P(S1|E1) < P(S2)$. Similarly, if the evidence E2 is sufficiently favorable for S2, we may have $P(S1) < P(S2|E2)$.

In essence, the problem is that when the DS provided the rule saying that the evidence E1 is unfavorable for S1 (rhyolite), he overlooked the fact that E1 is also unfavorable for S2 (a rhyolite plug), and did not supply a rule of the form E1 --> S2. Similarly, when he supplied the rule saying that the evidence E2 is favorable for a rhyolite plug, he overlooked the fact that E2 is also favorable for rhyolite, and did not supply a rule of the form E2 --> S1. Indeed, the DS should not be asked to supply rules for such obvious deductions.

The Matcher helps to detect these situations. It is the responsibility of the consultation system to take the appropriate actions to maintain the probabilistic consistency in the inference networks. In [1] it is shown how Prospector uses results from the Matcher to create additional rules ensuring that the probabilistic constraints will be maintained at run time when inference rules are applied.

B. Aid in Designing the Semantic Representation

Because statements in the knowledge base may be arbitrarily complex, their semantic encoding is often entered manually during the knowledge acquisition phase. During a consultation, however, the user is allowed to volunteer information to the system, and a parser [4] is used to create the semantic representation corresponding to the volunteered statements. The kinds of statements that can be translated by the parser depend upon taxonomy contents and an external grammar. Whether the semantic representation of statements is entered manually or constructed by a parser, the knowledge engineer needs to determine if the resulting representation is adequate. He must ensure that it reflects the intentions of the DS in all situations that could occur during of a consultation. Statements can be combined to form larger statements or broken into smaller units, and their semantic representation need not always be elaborate. Which representation is finally chosen depends upon what other statements are in the knowledge base and how they are related, as well as what the DS thinks is the most appropriate for each particular situation. Because the Matcher can be used to analyze how statements are related, it can assist in choosing an appropriate representation. In particular, no elaborate semantic representation may be needed for a statement (or a portion of a statement) that is unrelated to any other statement in the knowledge base. Because such a statement is unlikely to have a major effect on the consultation, a simple text-string representation would be adequate for most purposes.

In addition to determining if a restriction/generalization relation exists between two statements, the Semantic Network Matcher in Prospector can identify corresponding elements of the statements and point out the nature of their differences. This feature has been exploited to some extent in the knowledge acquisition module of Prospector [2] where it was used to choose a representation for a statement from possible alternative representations. For instance, a conjunction "X and Y" can be encoded either as a single statement or as two statements, "X" and "Y," connected by a logical AND in the inference network. The first alternative is chosen if a statement already exists in the knowledge base that is equal to or is a restriction of "X and Y," or if "X and Y" is not related to any existing statement. The second alternative is chosen otherwise. We believe this approch can be generalized, and that an automatic procedure using the Matcher can be devised to assist in the uniform, and perhaps optimal, encoding of all statements in the knowledge base.

C. As a Search Feature--Accessing by Contents

Development and testing of a knowledge base typically extend over long periods. The knowledge engineer cannot be expected to remember all the statements (or any labels assigned to them) that he

or another knowledge engineer developing the same knowledge base has already entered. The Matcher can be used as a search feature allowing the knowledge engineer to access statements by specifying a partial description of their (semantic) contents. In effect, the Matcher-search (a command of the knowledge acquisition system) will allow the knowledge engineer to say something like:

"Now I want to work on that portion of the inference network that deals with sulfide mineralization."

The search-by-content feature is accomplished by matching the partial description specified by the knowledge engineer with statements currently in the knowledge base.

III USE OF THE MATCHER IN CONSULTATION

A. As a Tool for Maintaining Consistency of the User's Answers:

1. Discovering Inconsistencies

If a user is allowed to volunteer information to an expert system, logical inconsistencies in the input could result. For example, suppose the user volunteers the two statements S1 and S2 concerning rhyolite and a rhyolite plug. Because S2 is a restriction of S1, if his statements imply that P(S2) > P(S1), he will be reminded that his answers are contradictory. This is the case, for instance, if the user says that "there are igneous rocks" with some degree of certainty, but later volunteers that "there is a rhyolite plug" with a higher degree of certainty. The contradictions occurring in a consultation often involve several levels of inference and long chains of restriction/generalization links, which sometimes have embarrassed our expert users while being impressed by Prospector's ability to detect the inconsistencies.

2. Changing Answers

A significant advantage of the Bayesian method used in Prospector for computing probabilities is the ease with which answers to questions can be changed without having to repeat all previous calculations. Basically, all that is required in changing an answer to a question about any evidence E is to change the probability for E and to propagate the results through the inference network.

The possibility of violating the restriction/generalization probabilistic constraints causes the only complication in this process. However, by keeping a record of how statements are related as computed by the Matcher , the answer-changing program knows that it may also have to change the probabilities of some of the related statements in order to maintain consistency. For instance, if the inference network contains the two rules S1 --> H1 and S2 --> H2, and the user gives a negative answer to a question about S1, the probability of H1 will be updated (in accordance with the rule strengths associated with the first rule). In addition, because S2 is a restriction of S1, the probability of H2 must also be updated (in accordance with the rule strengths of the second rule) as if a negative answer had been given for S2 as well. When the user then changes his answer for S1, the probabilities of both H1 and H2 will be automatically updated and propagated through the inference network.

By changing an answer, the user may contradict some of his earlier assertions, and changing these assertions may give rise to still further contradictions. This can confuse the user, but poses no problem for the answer-changing program, which is recursive and will make sure no contradictions are introduced before resuming a consultation.

B. Use of the Matcher as a Dialog Management Tool

1. Mixed Initiative, Volunteering Information

Prospector can work in either of two modes -- the consequent mode or the antecedent mode. In the consequent mode Prospector attempts to establish (or rule out) an hypothesis, and it traces backward through the inference network and searches for appropriate evidences to ask about. In the antecedent mode, the inference network is used in a forward direction to propagate the consequences of input information. Prospector is a mixed initiative system whereby the user has the option of taking control any time during the consultation session to inform Prospector about facts he believes to be relevant. The Matcher makes this possible by relating the volunteered information to the current knowledge base in the same fashion as it did for the knowledge acquisition phase.

2. Control Strategy and Goal Selection

The information volunteered by the user is often relevant to several hypotheses; In Prospector, a simple scoring criterion is used to select the goal hypothesis. Among other things, this criterion takes into account the volunteered statements (whose effect on the hypotheses may be encouraging or discouraging) that are linked to each hypothesis as recorded by the Matcher.

3. Interaction Psychology

Before the user is asked about the evidence E selected by the control strategy, Prospector

reminds him about any facts it thinks relevant. The information needed to recognize these facts are the links relating E to other statements in the knowledge base computed by the Matcher and recorded at some earlier phase of the consultation or during knowledge acquisition. How these facts are presented to the user depends upon the current "state" of the statements involved. The state of a space is determined by its certainty value and how that certainty was established --whether it was inferred by using rules, volunteered by the user, or inferred through restriction/generalization links through the Matcher. Depending upon the actual situation, one of several standard phrases is displayed before the question is asked, and an appropriate phrase is selected to ask the question.

The following are some of the standard phrases used in a Prospector consultation:

- You told me about ...
- You suspected ...
- I know you doubted ...
- Your statements imply ...
- I know there is reason to doubt ...
- I have reason to suspect ...
- I have reason to doubt ...

Thus, the program might preface a question about a piece of evidence E by saying: "I have reason to doubt E. What is your degree of belief about that?"

Clearly, these stock phrases are simple attempts to inform the user about the implications of his earlier statements. Although they have no effect on the function of Prospector and are not necessary in any logical sense, they enhance communication between the user and the consultation system and often serve to make the logical processes of the consultation system more evident.

The Matcher has been an important tool for the design of the interaction environment in all phases of development and use of the Prospector knowledge-based system. It is particularly important in the "psychology" of man-machine interaction in consultation systems that the user does not feel ignored and that the dialogs are not totally dictated by the system. Whenever possible, the user should be shown evidence that the system listens to him, understands what he says, and sometimes can even use the information he supplied!

IV CONCLUSION

By providing means to relate the statements in the knowledge base to each other, the semantic network Matcher in Prospector has been an important instrument in supporting many of the features that constitute the AI contents of the system. We believe that the approach is a general one, and can enhance the intelligent behaviour of any knowledge-based system.

REFERENCES

[1] Duda, R.O., P.E. Hart, N.J. Nilsson, R. Reboh, J. Slocum and G.L. Sutherland, "Development of a Computer-Based Consultant for Mineral Exploration," Annual Report, SRI Projects 5821 and 6415, SRI International, Menlo Park, California (October 1977).

[2] Duda, R.O., P.E. Hart, K. Konolige, R. Reboh, "A Computer-Based Consultant for Mineral Exploration," Final Report SRI Project 6415 SRI International, Menlo Park, California (September 1979).

[3] Hayes-Roth, F., "The Role Of Partial and Best Matches in Knowledge Systems," in Pattern-Directed Inference Systems, D.A-Waterman and F. Hayes-Roth, eds., pp. 557-574 (Academic Press, New York, 1978).

[4] Hendrix, G.G., "LIFER: A Natural Language Interface Facility," SIGART Newsletter, No. 61, pp 25-26 (February 1977).

[5] Hendrix, G.G., "Encoding Knowledge in Partitioned Networks," in Associative Networks - The Representation and Use of Knowledge in Computers, N. V. Findler, ed., Academic Press, New York (1979).

[6] Reboh, R., "A Knowledge Acquisition Environment for Expert Consultation Systems," Ph.D. Dissertation, Department of Mathematics, Linkoping University, Sweden (to appear 1980).

[7] Waterman, D.A. and F. Hayes-Roth, eds., Pattern-Directed Inference Systems (Academic Press, New York, 1978).