AD-A095 425 RAND CORP SANTA MONICA CA F/G 5/2
MATCHING AND ABSTRACTION IN KNOWLEDGE SYSTEMS,(U)
JAN 80 F HAYES-ROTH
UNCLASSIFIED RAND/P-6440 NL

1 OF 1
AD A
095425

END
DATE
FILMED
3-81
DTIC

AD A095 325

MATCHING AND ABSTRACTION IN KNOWLEDGE SYSTEMS

Frederick Hayes-Roth

January 1980

P-6440

PREFACE

This briefing was presented at the Symposium on Artificial Intelligence in Information Science during the 1979 Annual Meeting of the American Society for Information Science (ASIS) in Minneapolis. It covers several areas of common interests to both AI and information science.

MATCHING AND ABSTRACTION IN KNOWLEDGE SYSTEMS

INTRODUCTION

We all suspect that there may be only a few simple but central problems in Information Science. Perhaps, if we could just solve those problems, we would attain the utopian image that motivated us in our earliest days of involvement in this field. But, of course, we cannot really solve any of these hard problems; instead we must resign ourselves to engineering moderate improvements in existing technologies. And that's why you hear so many details about the engineering aspects of what we are trying to do.

The first problem is creating what I'll call a "Knowledge System," putting into the computer what people have variously called knowledge, or representations of interesting relationships, or expertise, like what a word means, or how it ought to generate inferences. The second one is getting the system to work. The first problem is a human and theoretical limitation; the second is an engineering limitation.

And the third problem is a methodological one. Most of the interesting problems that humans solve are not solved by following a particular algorithm deterministically to some simple solution. Rather, solutions are usually selected from a large set of possible, more or less "good" answers to a question; that is, a simple question to retrieve some information usually produces a number of partially correct responses, and that produces a requirement to search a set of alternatives for the preferred ones.

A central theoretical problem common to the two fields of Artificial Intelligence (AI) and Information Science (IS) concerns the question of partial matching: How do I compare one thing to another thing? That is, what is the structure of ambiguity?

Ambiguity, say in the comparison of two things, arises because there are many ways to see two things as being similar. I'll conjecture for you today that this problem is one of the few core problems in this general area. To establish that, I'll go through a number of examples, and try to give you an intuitive sense, if not a formal understanding, of the issues.

MATCHING AND ABSTRACTION IN KNOWLEDGE SYSTEMS

KNOWLEDGE SYSTEMS AND DATA REPRESENTATION

ROLES OF PARTIAL MATCHING

- INFORMATION RETRIEVAL
- GENERALIZATION AND INDUCTION
- INTERPRETATION

HIERARCHIES AND ABSTRACTION

FIGURE 1

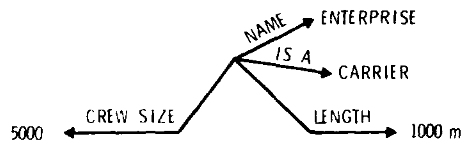
I want to relate this problem of matching to abstraction. In particular, I want to convey the idea that hierarchies, as we have known, play a crucial role in structuring knowledge and in enabling us to solve many knowledge-related problems, such as getting knowledge in and getting it out of the system. I'll talk about the use of partial matching and information retrieval, and how it relates to generalization and induction, and how we can use partial matches to interpret data, or interpret a query, etc. And then I'll wind up by discussing some of the key research issues.

The kinds of databases we use are called knowledge-based systems, or just knowledge systems. These systems are usually computer languages for writing descriptions of objects, descriptions of how they relate to one another, and some inference rules that describe what kind of inferences to draw when you find an object of a certain sort or in some relationship of interest. Then the problem arises of which inferences to draw, given that you have only a finite amount of time.

KNOWLEDGE SYSTEMS AND DATA REPRESENTATION

OBJECTS

THE ENTERPRISE IS A CARRIER WHOSE LENGTH IS 1000 m
AND WHOSE CREW SIZE IS 5000 AND . . .



INFERENCE RULES

IF THERE IS A CARRIER WHOSE LENGTH IS GREATER THAN 800 m
AND WHOSE RANGE IS UNKNOWN, THEN SET THE RANGE
TO 1000 miles

FIGURE 2

At Rand we have a few programming languages for non-computer-expert people to use, where they can write descriptions like the one in this figure: "The Enterprise is a carrier whose length is 1000 meters and whose crew size is 5000 men." In the computer, that turns into a graph structure with a node, and various links, where every link has an attribute type, such as a name, and then a value.

The systems actually solve problems by having encoded in them some human expertise about how to draw inferences. For example, if there's a carrier whose length is greater than 800 meters, and whose range is unknown, then set the range to 1000 miles. If I wanted to figure out the carrier's range, I could either apply this rule to all data instances in the database, or I could work backwards to see if the premises for this conclusion are justified.

People are now finding hierarchies to be quite useful. That's not surprising, because through all time, people have been trying to use hierarchies. What is unusual is that these programming systems for creating knowledge-based systems now are providing natural structures for creating hierarchies, are simplifing the description of the knowledge of the world, and are simplifying the number of rules that one has to create because one can state general inference procedures quite generally if they apply to many lower-level members of a hierarchy.

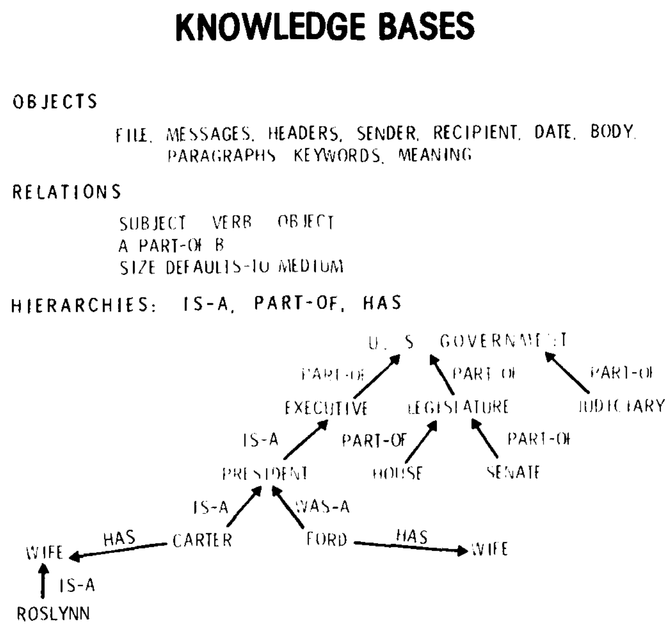
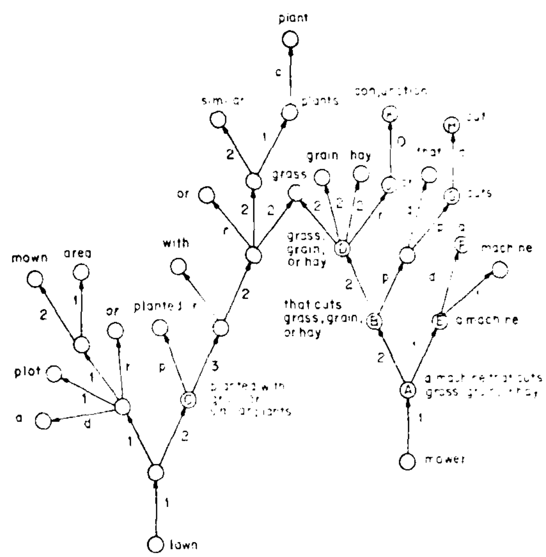


FIGURE 3

Let me just give you an example: This figure is a graph representation of part of what you would say if you wanted to describe the government. You would say that there are three parts of the U.S. government: the Executive, the Legislative, and the Judiciary. The President is an executive who is part of the U.S. government. Carter is President now; he has a wife; Roslynn is a wife.

These three kinds of relations (is-a, part-of, and has) go a long way toward simplifying a great many descriptions of the world. And people have devised quite simple algebraic rules for how to solve such general questions as: "Is Roslynn part of the U.S. government?" Or: "Is Carter part of the U.S. government?" Or, if one wants to say something about all parts of the U.S. government, knowing whether those inferences apply to these things.

At Rand we are generating a program called ROSIE that is intended for wide military use (and we hope domestic agencies of the government will use it,too) for putting in their knowledge of the world, putting in some rules they would like the computer to apply routinely, and having it apply them. We find that once we get people started with this kind of system, non-programmers can keep it up to date and extend it.



A LAWN IS A MOWN AREA OR PLOT PLANTED WITH GRASS OR SIMILAR PLANTS

A MOWER IS A MACHINE THAT CUTS GRASS, GRAIN OR HAY

FIGURE 4

This figure is from a previous project that I did with Dave McDonald at Carnegie-Mellon. Our goal was to avoid completely the programming problem, if we could. So we created a system to create these kinds of hierarchical networks, essentially, by directly parsing the American Heritage Dictionary definitions.

These two definitions were two of the word senses of a lawn and a mower, out of that dictionary. The kinds of research problems we were studying were "So, what's a lawnmower?" for example; or, in general, how could one extend a knowledge base, either by taking in the cumulative human wisdom as recorded in such great books as the dictionary, or by synthesizing new meanings by means of some general search processes through such knowledge bases. We wll return to this problem in a little while.

INFORMATION RETRIEVAL

QUERY IS A SET OF OBJECTS, VARIABLES AND RELATIONS

REPLIES ARE A SET OF OBJECTS AND RELATIONS THAT PARTIALLY MATCH THE QUERY

FIGURE 5

Now let me tell you why I think partial matching is one of the key theoretical issues. I look at information retrieval as a partial matching problem, where a query in Artificial Intelligence terms would be represented as a set of relations among a set of objects, which may have some variables that are to be instantiated, such as: "Who is the President of the U.S.?" That might be described as a graph, where you have President of U.S. as the known part of the graph, and you want to find all the instances in the data base that contain that incomplete graph. The replies are the things in the data base that partially match the query. A good reply is something that satisfies all the constraints that the query entails, but you may not always be able to do that.

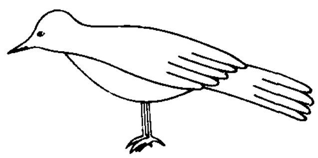


FIGURE 6

I want to give you another example to demonstrate that there are many different answers depending on what the question is. Suppose I have a data base with two birds, a churl and a jovie. I don't care what kind of description you would propose that I have in there, but let's suppose we had the two in Figure 6. A typical human problem, and one which is an analog of the general information retrieval problem, is to identify something when only part of it is present.

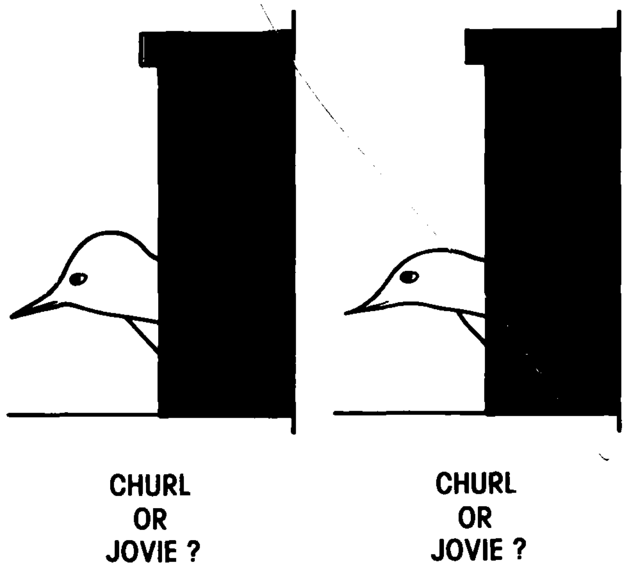


FIGURE 7

Given that these parts are the same size, the same orientation, and the same scale as the two original items that contain them, there are many ways to solve this problem. If any one of these attributes varies from the original, there are no good ways to solve this partial-matching problem.

DESIRABLE PROPERTIES OF ANY MODEL

(1) PART RECOGNIZABILITY (CLASSIFIABILITY)
A PART OF AN EXAMPLE CAN BE AS RECOGNIZABLE (CLASSIFIABLE) AS A WHOLE

(2) ATTRIBUTE COMBINATION EFFECT
A PART CAN BE RECOGNIZED (CLASSIFIED) BECAUSE SOME COMBINATION OF ITS ATTRIBUTES IS STORED (DIAGNOSTIC)

(3) PART-WHOLE CONTINUITY
ASSUMING THAT THE SAME MECHANISM UNDERLIES RECOGNITION (CLASSIFICATION) OF SMALL, MEDIUM, OR LARGE PARTS,
AN EXAMPLE CAN BE RECOGNIZED (CLASSIFIED) BECAUSE SOME COMBINATION OF ITS ATTRIBUTES IS STORED (DIAGNOSTIC)

(4) STRENGTH EFFECT
CONFIDENCE IN RECOGNITION (CLASSIFICATION) JUDGMENTS IS RELATED TO MEMORY REPRESENTATION STRENGTH AND STRENGTH INCREASES WITH PRESENTATION FREQUENCY

FIGURE 8

Part of what we've been doing is just confirming our intuition that human information processing satisfies what you might take to be some trivial, but intuitively desirable, properties. I hope some of these things sound trivial to you, but in fact most of the psychological literature mirrors information systems algorithms by trying to short-circuit the complexity of dealing with arbitrary configurations of partial descriptions of objects in order to do "look up."

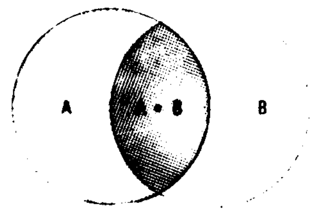
ASSUMPTIONS OF THE PROPERTY-SET MODEL

- (PART RECOGNIZABILITY, ATTRIBUTE COMBINATION EFFECT, AND PART-WHOLE CONTINUITY)
  - THE ELEMENTS OF THE MEMORY REPRESENTATIONS FOR PRESENTED EXAMPLES ARE COMBINATIONS OF SIMULTANEOUSLY OCCURRING ATTRIBUTES (PROPERTY-SETS)
- (STRENGTH EFFECT)
  - THE STRENGTHS OF PROPERTY-SETS IN MEMORY ARE INCREASING FUNCTIONS OF THEIR FREQUENCY AND SALIENCY IN PRESENTED ITEMS

FIGURE 9

We find that people are very good at recognizing a whole from a part. They accomplish this by dealing with a combination of attributes as if it were a configuration, that is, they don't treat things independently. The more information you give them the better, and the more familiar they are with something the better they get, too. In our research we develop computer programs and models of people which mirror such capabilities, and then try to solve the ensuing problems. Some of these programs require extraordinary amounts of computer time.

PARTIAL AND BEST MATCHES



ABSTRACTION: A • B = COMMON COMPONENTS OF DESCRIPTIONS A AND B

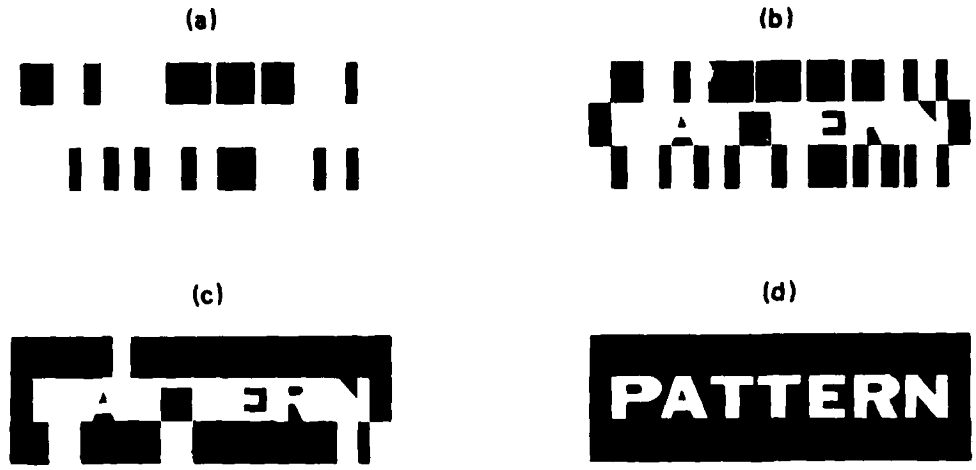
RESIDUALS { A − A • B = PROPERTIES TRUE OF A ONLY
{ B − A • B = PROPERTIES TRUE OF B ONLY

PARTIAL MATCH: PM(A,B) = (A • B, A − A • B, B − A • B)

FIGURE 10

Now I want to talk about what goes technically under the term of sub-graph homomorphism, which I'll simply call a partial match. The basic idea here that if I have two representations of some information, say "A" and "B", I am often interested in comparing them. The comparison of major interest, which I'll call "A" * "B", is a description of everything common to the two initial descriptions. Now, as we get into these general, structured knowledge bases, we see that there's more than one answer, more than one way to hold up a graph and "embed" it within another graph.

You can do some interesting things with this simple comparison operator. You can use the commonalities to induce new concepts, abstractions, and generalizations. You can even make use of the "residuals," the properties of the initial descriptions that are uniquely associated with only one of the two items compared. These residuals induce interesting structures over the data base. So, when I refer to a partial match in general, I may occasionally emphasize either the best match between the two compared structures or their corresponding residuals.



(a) Example 1
(b) Example 1 * Example 2
(c) Example 1 * Example 2 * Example 3
(d) Example 1 * Example 2 * Example 3 * Example 4

FIGURE 11

In 1906 a psychologist by the name of Sir Francis Galton used the technology of his day, which was photography, to form a general theory of how human beings learn and recognize things. He called it a composite photograph theory. The problem he was trying to solve was this: How is it that I can recognize a face, regardless of the aspect, or angle of the face, or distance; how is it that I build up a composite template to recognize people's faces from multiple views? His theory was that each person's face might be represented as a photographic transparency. The photographic transparencies would be overlaid, superimposed, homologously, until only the commonalities would emerge.

This figure shows a sequence of superpositions of descriptions of something, like a transparency of a face. By superimposing them, the common characteristics should evolve.

Here I'm using features, which are present, and representing them as simply transparent. Things that are absent are represented as black. Galton's theory was that if the brain could somehow magically superimpose these things, over time, their commonalities would emerge to define the "pattern."

BASIC USES OF PARTIAL MATCHES

- **ABSTRACTING COMMONALITIES AND IDENTIFYING DIFFERENCES**

  SEVERAL EXAMPLES ARE COMPARED FOR:

  CONCEPT LEARNING
  PATTERN DISCOVERY
  RULE INDUCTION
  PREDICATE DISCOVERY
  ANALOGICAL REASONING

- ***PATTERN RECOGNITION AND CLASSIFICATION BY CONSTRAINT SATISFACTION***

  DATA DESCRIPTION PARTIAL-MATCHED TO PROTOTYPE DESCRIPTION

  COMMONALITIES - SATISFIED CONSTRAINT
  RESIDUALS - NOISE ERROR DEVIATION

- **INTERPRETATION OF DATA IN A SYSTEM OF FRAMES**

  MULTIPLE ALTERNATIVE INTERPRETATIONS [illegible] ARE PLAUSIBLE
  VARIOUS INTERPRETATIONS ARE CONSISTENT UNRELATED INCONSISTENT
  DATA DESCRIPTION PARTIAL MATCHED TO HIERARCHICAL SYSTEM OF TEMPLATES FRAMES
  OVERALL INTERPRETATION IS BEST MATCH BETWEEN DATA AND MULTILEVEL FRAME SYSTEM

<u>FIGURE 12</u>

But again, real problems arise. How do you get two structures to line up with one another? How do you orient them? You almost have to solve the problem of what's common to the face before you know what the face is. But, we've made some progress. I'm going to try and review some of that for you, and give you some examples of how these abstracted commonalities help define new concepts, and perhaps even define some rules that you could use in general. I might give you multiple examples of how you'd want a system to behave, and what inferences to draw on what case. The system would pull out the right inferences, because it would generalize the rule. I'll talk about discovering some new predicates to compact a data base that has several things which partially match with one another, and, to the extent I can, I'll talk about pattern recognition.

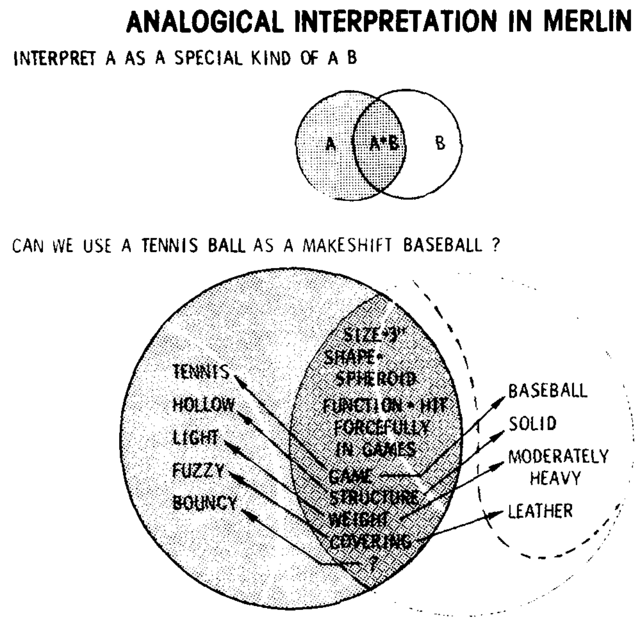


FIGURE 13

One nice example comes out of the Artificial Intelligence literature, which was done by Moore and Newell of Carnegie-Mellon. It's a system called Merlin. They were interested in the partial matching of two concepts to find out, for example, in what sense a "trailer" could be a "house," or whether a "human being" can be seen as a "work horse" (that is, the general problem of how to look at one concept as an instance of another). This is a problem of partial matching. The Moore and Newell approach helps illuminate some general properties.

When I played the suburban version of sandlot baseball, we used a tennis ball instead of a hardball because a hardball was very dangerous. This suggested an example for today's meeting: How could you interpret a tennis ball as a hardball? In what sense is a tennis ball a kind of baseball? Merlin proposes to look at a tennis ball as one concept, and a baseball as another concept, and then to find ways in which they are similar. The dictionary says that they are both around three inches, they are both spheroid shapes, they are both hit forcefully in games, but the game for a tennis ball is tennis, whereas the game for a baseball is baseball. The structure of a tennis ball is hollow, whereas the structure of a baseball is solid. One is covered in leather, the other in fabric, etc.

This show us that if you have in your knowledge base the definition of one concept in terms of relations and attribute values of other concepts, hierarchically, that is, many of these terms are concepts which themselves have refined definitions in the dictionary.

Once you have a structure where everything is encoded in terms of other things, you can often get very good, very fast matches by lining up the two concepts on their shared types of attributes, and then treating the residuals (that is the things that are different about them) as variations on that conceptual theme. For example, you find that a tennis ball might work as a makeshift baseball, as long as the fact that it is designed for tennis is not critical. As long as none of these differences is critical, the substitution is okay.

Merlin also allows you to recurse on this problem. If you want to know how the fuzzy cover compares with the leather one, you would ask the same type of comparison question recursively. One key idea is that if the knowledge is structured hierarchically, you get a partial match on some of the concepts that line up, and then you can recursively compare the residuals by partial matching.

Of course, no one's going to tell you if the solution you find is "good enough"; that question is exogenous to this kind of problem.

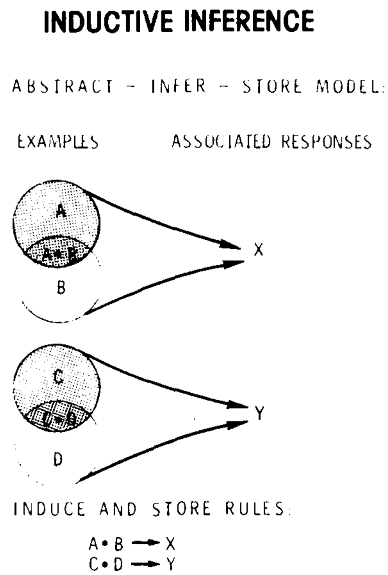


FIGURE 14

Partial matching is also used in a certain kind of inductive inference.

Suppose I have multiple examples: "A" and "B" here are both supposed to entail some inference "X" or some response "X." And I have "C" and "D," both of which are supposed to lead you to the same response, namely "Y." The inductive theory we've been developing proposes that the best kind of generalization you can make would be that anything that has the commonalities of both A and B ought to lead to the response X. That's what I have in this figure: A * B ---> X, and similarly for C * D ---> Y. This kind of theory can be embellished, but here I'll only give you an example of the way it's used.

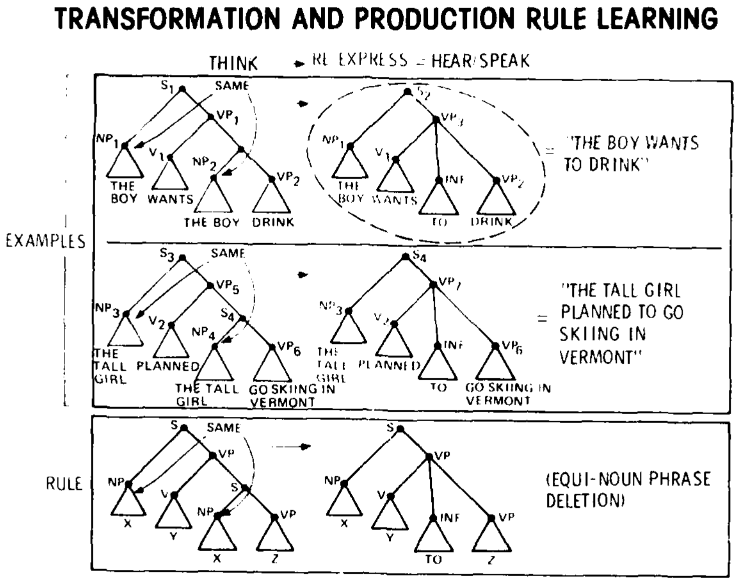


FIGURE 15

We have used this kind of theory in an experiment on the induction of transformational grammar rules--not because we are interested in transformational grammar rules, but because it is a set of rules that many people have studied. For example, you might have a deep structure representation of a sentence, like, "The boy wants the boy to drink," meaning, "The boy wants that the boy [should] drink." One of the rules of transformational grammar, the equi-noun phrase deletion rule, says that the sentence should be re-written as, "The boy wants to drink."

Now, you might have another sentence which says, "The tall girl planned to go skiing in Vermont" in its before and after forms. And if you superimpose these graphs on each other and pull out the best partial match, what you get is a rule that looks like this, where the residuals, like "the tall girl" versus "the boy" are replaced by general variables, free variables, to be instantiated by any corresponding noun phrase in a new sentence that fits that slot. You can induce a variety of rules like that if you spend a lot of computer time.

PREDICATE DISCOVERY

CORRESPONDING RESIDUAL VALUES FROM PARTIALLY COMPARABLE SITUATIONS IDENTIFY NEW PREDICATES

EXAMPLES

BECAUSE JOHN IS SO TALL, IT IS DIFFICULT TO FIND CLOTHES THAT FIT HIM.

BECAUSE MARY IS SO SHORT, IT IS HARD TO GET CLOTHES THAT CAN FIT HER.

BECAUSE JOANNE IS SO FAT, IT IS IMPOSSIBLE TO GET APPAREL THAT IS THE RIGHT SIZE.

BECAUSE TOM IS SO SKINNY, IT IS NOT POSSIBLE TO FIND CLOTHES THAT ARE SUITABLE.

PARTIAL ⇩ MATCH

(BECAUSE U IS SO V, IT IS W TO X)

	RESIDUALS	NEW PREDICATES
U:	{JOHN, MARY, JOANNE, TOM}	NAME
V:	{TALL, SHORT, FAT, SKINNY}	BODY SIZE
W:	{DIFFICULT, HARD, IMPOSSIBLE, NOT POSSIBLE}	DIFFICULTY
X:	{FIND CLOTHES THAT CAN FIT HIM GET CLOTHES THAT CAN FIT HER} . . .	(*FIND CLOTHES ... FIT*) {FIND, GET} {CLOTHES, APPAREL} . . .

<u>FIGURE 16</u>

Another application, rather than using the commonalities of compared sentences, would focus on the corresponding differences, for example, "the tall girl" versus "the boy," once you had lined up the two structures. If you do this, you can discover new predicates and can probably compress a great deal of language and generate a lot of syntactic structures for various sub-specialities of scientific fields.

This figure lists a number of sentences to show what happens when you partially match them, without even the benefit of having a grammar, and to show that by lining up the commonalities, a grammar rapidly emerges over this small set. You could also apply it more generally to larger domains.

Let's line up these words and try to maximize some goodness-of-fit measure among them. What we get is a match that says, "Well, notice that they all have 'because' and they also have 'something is so,' 'it is something to x'." These are the corresponding places in the sentences that have different residuals. Now, note that each one of these correspondences defines what may not already be a concept in your computer language, but is apparently an implicit concept in English. I don't know what to call them, necessarily, but I'll just take the illustration a little further. A useful kind of inference for a system that is trying to assimilate all this knowledge and restructure its data base would be that category U is a set of <u>names</u>, because they are all names. Category V is something about <u>body size</u>. W describes some <u>degree of difficulty</u>; but in fact, more examples would cause this concept to be weakened to some expression of "ease" or "facility."

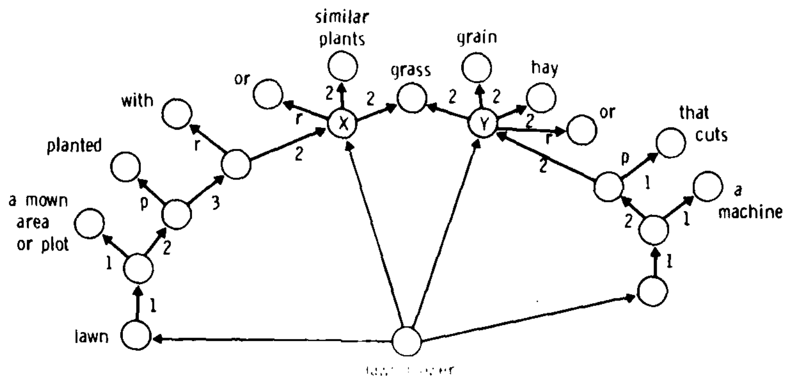
Notice here that I have large structures that don't correspond to any simple category. So now, as in Merlin, we'll recursively apply partial matching, and we get a structure like category X, "Find clothes that fit," where these are now secondary predicates, which are induced. "Find" and "get" are instances of this general category. We can see many combinatorial issues arising when we try to explore all these alternatives in building an actual system.

It's hard for people to express all of this type of knowledge for a computerized database, because each one of these ambiguous category structures of language is more or less important, depending on what one is interested in. And that's why we want to get away from hand-crafting particular meanings in terms of complex computer programs. We would rather have this kind of induction happen dynamically, in the context of a problem that has to be solved, where we can bring to bear all the relevant experiences through partial matching. That's a real wish. How might that happen?

SYNTHESIS OF NOVEL SEMANTIC INTERPRETATIONS

PROBLEM: INTERPRET A NOVEL PHRASE (LAWN MOWER) GIVEN ONLY CONSTITUENT MEANINGS

SOLUTION: PARTIAL-MATCH THE CONSTITUENT MEANINGS AND EXPRESS THE RESULT VERBALLY



METHOD: PERFORM INTERSECTION SEARCH OF SEMANTIC NET

RESULT: A LAWN MOWER IS A MACHINE THAT CUTS GRASS OR SIMILAR PLANTS

FIGURE 17

We frequently encounter suggestions to exploit intersection searches in knowledge networks. Loosely speaking, a path between two points may define the solution or the meaning of the relationship between them. That is exactly the method used in our dictionary task. First, we created these hierarchies: "A lawn is a mown area planted with grass or similar plants," and the mower definition was, "a machine that cuts hay, grain, or grass." Remember we created this by just reading the dictionary, and now we want to ask, "What's a lawn mower?" We did this more generally for noun-noun phrases, adjective-noun phrases, subject-verb-object phrases. The general idea was to find a meaningful way in which one thing could modify another one, constrained only by the fact that in certain English phrases, the syntax constrains one component to apply to another, rather than vice versa.

We used a method which was a generalization of something Fiksel and Bower had published for a totally different purpose, to create a parallel automata for information retrieval to answer queries. We started search processes at the node for "lawn" and "mower" (not on the figure) and then searched in all possible directions looking for a meaningful intersection. We came up with the intersection shown, and used a few simple algebraic rules that specified how to transform this kind of intervening path into a simpler expression, and in turn, how to paraphrase that expression in English.

We built a completely lexically based language understanding system for a very small portion of English. No conceptual primitives were entered in the system, and the main method was to compare examples by partial matching over these structures. Thus we find "A lawn mower is a machine that cuts grass, or similar plants."

Now it doesn't work for everything, but it worked for a suprisingly large number of things. We never really ran up against what I would call fundamental obstacles.

THE PARTIAL MATCH ADMISSIBILITY CRITERION

THE MORE SIMILAR A AND B ARE,
THE FASTER THE PARTIAL MATCH SHOULD BE.

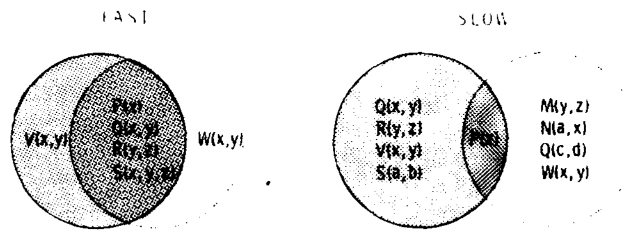


FIGURE 18

In the time remaining I want to discuss some of the interesting problems that remain to be solved. The partial match admissibility criterion is one. Suppose you have a good algorithm for a partial match. One thing it ought to satisfy is this test: If I give you two things to compare, the more similar they are the faster your algorithm should be.

For example, if you have a spelling corrector on a computer system, when you type in a word it's supposed to tell you what the right word is. The more similar the input is to a correct word, the faster the corrector should be. I can imagine writing a special-purpose program for that. But in information retrieval systems in general, where a query consists of several keys and the answer is found by taking the inverted indexes for each key and intersecting them, you get just the opposite performance. I don't have the solution to that; I came today in the hope that somebody here would give me the answer.

HIERARCHIES AND ABSTRACTION
– SOME OBSERVATIONS –

HIGHER-LEVEL CONCEPTS MAY BE INFERRED BY PARTIAL MATCHING

<CATEGORIES OF FUNCTIONALLY SUBSTITUTABLE ENTITIES>

THESE ABSTRACTIONS REFORMULATE THE DATA-BASE

<HIGHER-LEVEL, LESS PRIMITIVE CODES>
<DATA REDUCTION>

MATCHES EXPLOIT THE HIERARCHY TO SPEED SEARCH

<HIGH-LEVEL CODE MATCHES OCCUR FIRST>
<FEWER CHANCE HITS ON COMMON ATTRIBUTES>

FIGURE 19

Actually, we do have some ideas. One is that partial matches must be structured to run over a hierarchical data base-- and it's the hierarchies in part that give you the speed. At the risk of oversimplifying, let me address some overall conclusions of this research area.

The first observation that I want to leave you with is that we can use this idea of partial matching to infer higher-level concepts, such as those predicates I was talking about earlier, like "fit" and "name." Those were just the types of things that Merlin used in order to go faster in its comparisons. These new concepts were categories of functionally substitutable entities. We found that we could substitute this or that and still have the same general structure. Second, once I had found those concepts, I could recode my entire knowledge base to have all these additional relations. I would then have higher-level descriptions than I started with, that is, each predicate would be more restricted in applicability, even though the higher-level terms would also have lower-level descriptors below it. In turn, these high-level predicates could produce a data reduction factor, typical to taxonomies.

The third point is to use this hierarchy to speed the search, as in the example I gave you earlier. High-level coding helps you in many ways. It is just the opposite from most of psychological theorizing that says, "I understand things in terms of primitive concepts; I break high-level codes down to digest them." With the approach discussed here, you go the other way around. You work dynamically, comparing the current situation to all of your relevant experience, but at the highest possible level of description. This I have described elsewhere as "wait-and-see" inference. It seems eminently reasonable, functionally powerful, and--with today's machines--extraordinarily slow.

THE KEY RESEARCH ISSUES

- EXTENDING THE KNOWLEDGE REPRESENTATIONS
- REFORMULATING KNOWLEDGE TO SIMPLIFY SEARCHES
- IMPROVED HARDWARE AND ALGORITHMS FOR MATCHING

FIGURE 20

What are we working on now? We are trying to represent more knowledge than we can currently put into our computers, and we are trying to create programs that everybody can use to generate some big knowledge bases, as in a legal reasoning project at Rand. We are trying to create for civil justice research a complete description of all laws, legal rules, and their application in a body of actual cases. That's a large order, so we are currently restricting ourselves to a very small area of law.

Once you get this knowledge in the computer, the key is to decide what kind of search problem you need to solve. Then you need to reformulate it, so that you've already pulled out the common generalizations that can speed up the essential search processes. As to improved hardware, I'm hedging my bets because it may be practically infeasible for a large class of problems. I think improved algorithms may be a far better bet for most of the major problems of immediate interest.

BIBLIOGRAPHY

Fiksel, J. R., and Bower, G. H. Question answering by a semantic network of parallel automata. Journal of Mathematical Psychology, 13, 1976, 1-45.

Hayes-Roth, B., and Hayes-Roth, F. Concept learning and the recognition and classification of exemplars. Journal of Verbal Learning and Verbal Behavior, 16, 1977, 321-338.

Hayes-Roth, F. The role of partial and best matches in knowledge systems. In D. A. Waterman and F. Hayes-Roth (eds.), Pattern-Directed Inference Systems. New York: Academic Press, 1978.

Hayes-Roth, F., and McDermott, J. An interference matching technique for inducing abstractions. Communications of the ACM, 21, 1978, 401-411.

Hayes-Roth, F., Waterman, D. A., and Lenat, D. B. Principles of pattern-directed inference systems. In D. A. Waterman and F. Hayes-Roth (eds.), Pattern-Directed Inference Systems. New York: Academic Press, 1978.

Lenat, D. B., and Harris, G. Designing a rule system that searches for scientific discoveries. In D. A. Waterman and F. Hayes-Roth (eds.), Pattern-Directed Inference Systems. New York: Academic Press, 1978.

McDonald, D., and Hayes-Roth, F. Inferential searches of knowledge network as an approach to extensible language-understanding systems. In D. A. Waterman and F. Hayes-Roth (eds.), Pattern-Directed Inference Systems. New York: Academic Press, 1978.

Moore, J., and Newell, A. How can Merlin understand? In L. W. Gregg (ed), Knowledge and Cognition. New York: Lawrence Erlbaum Associates, 1974.

Stefik, M. An examination of a frame-structured representation system. Proc. 6th Int. Joint Conf. on AI, Tokyo, Japan, 1979.

Waterman, D. et al. Design of a rule-oriented system for implementing expertise. The Rand Corporation, N-1158-1, May 1979.

Waterman, D. A., and Hayes-Roth, F. An overview of pattern-directed inference systems. In D. A. Waterman and F. Hayes-Roth (eds.), Pattern-Directed Inference Systems. New York: Academic Press, 1978.

DAT
ILM