SOME EXPERT SYSTEM NEED COMMON SENSE

John McCarthy
Computer Science Department
Stanford University
Stanford, CA 94305
jmc@cs.stanford.edu
http://www-formal.stanford.edu/jmc/

1984

Abstract

An *expert system* is a computer program intended to embody the knowledge and ability of an expert in a certain domain. The ideas behind them and several examples have been described in other lectures in this symposium. Their performance in their specialized domains are often very impressive. Nevertheless, hardly any of them have certain *common sense* knowledge and ability possessed by any non-feeble-minded human. This lack makes them "brittle". By this is meant that they are difficult to extend beyond the scope originally contemplated by their designers, and they usually don't recognize their own limitations. Many important applications will require common sense abilities. The object of this lecture is to describe common sense abilities and the problems that require them.

Common sense facts and methods are only very partially understood today, and extending this understanding is the key problem facing artificial intelligence.

This isn't exactly a new point of view. I have been advocating "Computer Programs with Common Sense"since I wrote a paper with that title in

1958. Studying common sense capability has sometimes been popular and sometimes unpopular among AI researchers. At present it's popular, perhaps because new AI knowledge offers new hope of progress. Certainly AI researchers today know a lot more about what common sense is than I knew in 1958 — or in 1969 when I wrote another paper on the subject. However, expressing common sense knowledge in formal terms has proved very difficult, and the number of scientists working in the area is still far too small.

One of the best known expert systems is MYCIN (Shortliffe 1976; Davis, Buchanan and Shortliffe 1977), a program for advising physicians on treating bacterial infections of the blood and meningitis. It does reasonably well without common sense, provided the user has common sense and understands the program's limitations.

MYCIN conducts a question and answer dialog. After asking basic facts about the patient such as name, sex and age, MYCIN asks about suspected bacterial organisms, suspected sites of infection, the presence of specific symptoms (e.g. fever, headache) relevant to diagnosis, the outcome of laboratory tests, and some others. It then recommends a certain course of antibiotics. While the dialog is in English, MYCIN avoids having to understand freely written English by controlling the dialog. It outputs sentences, but the user types only single words or standard phrases. Its major innovations over many previous expert systems were that it uses measures of uncertainty (not probabilities) for its diagnoses and the fact that it is prepared to explain its reasoning to the physician, so he can decide whether to accept it.

Our discussion of MYCIN begins with its ontology. The ontology of a program is the set of entities that its variables range over. Essentially this is what it can have information about.

MYCIN's ontology includes bacteria, symptoms, tests, possible sites of infection, antibiotics and treatments. Doctors, hospitals, illness and death are absent. Even patients are not really part of the ontology, although MYCIN asks for many facts about the specific patient. This is because patients aren't values of variables, and MYCIN never compares the infections of two different patients. It would therefore be difficult to modify MYCIN to learn from its experience.

MYCIN's program, written in a general scheme called EMYCIN, is a so-called *production system*. A production system is a collection of rules, each of which has two parts — a pattern part and an action part. When a rule is activated, MYCIN tests whether the pattern part matches the database. If so this results in the variables in the pattern being matched to whatever entities are required for the match of the database. If not the pattern fails and MYCIN tries another. If the match is successful, then MYCIN performs the action part of the pattern using the values of the variables determined by the pattern part. The whole process of questioning and recommending is built up out of productions.

The production formalism turned out to be suitable for representing a large amount of information about the diagnosis and treatment of bacterial infections. When MYCIN is used in its intended manner it scores better than medical students or interns or practicing physicians and on a par with experts in bacterial diseases when the latter are asked to perform in the same way. However, MYCIN has not been put into production use, and the reasons given by experts in the area varied when I asked whether it would be appropriate to sell MYCIN cassettes to doctors wanting to put it on their micro-computers. Some said it would be ok if there were a means of keeping MYCIN's database current with new discoveries in the field, i.e. with new tests, new theories, new diagnoses and new antibiotics. For example, MYCIN would have to be told about Legionnaire's disease and the associated *Legionnella* bacteria which became understood only after MYCIN was finished. (MYCIN is very stubborn about new bacteria, and simply replies "unrecognized response".)

Others say that MYCIN is not even close to usable except experimentally, because it doesn't know its own limitations. I suppose this is partly a question of whether the doctor using MYCIN is trusted to understand the documentation about its limitations. Programmers always develop the idea that the users of their programs are idiots, so the opinion that doctors aren't smart enough not to be misled by MYCIN's limitations may be at least partly a consequence of this ideology.

An example of MYCIN not knowing its limitations can be excited by telling MYCIN that the patient has *Cholerae Vibrio* in his intestines. MYCIN will cheerfully recommend two weeks of tetracycline and nothing else. Presumably this would indeed kill the bacteria, but most likely the patient will be dead of cholera long before that. However, the physician will presumably know that the diarrhea has to be treated and look elsewhere for how to do it.

On the other hand it may be really true that some measure of common sense is required for usefulness even in this narrow domain. We'll list some areas of common sense knowledge and reasoning ability and also apply the criteria to MYCIN and other hypothetical programs operating in MYCIN's domain.

1 WHAT IS COMMON SENSE?

Understanding common sense capability is now a hot area of research in artificial intelligence, but there is not yet any consensus. We will try to divide common sense capability into common sense knowledge and common sense reasoning, but even this cannot be made firm. Namely, what one man builds as a reasoning method into his program, another can express as a fact using a richer ontology. However, the latter can have problems in handling in a good way the generality he has introduced.

2 COMMON SENSE KNOWLEDGE

We shall discuss various areas of common sense knowledge.

1. The most salient common sense knowledge concerns situations that change in time as a result of events. The most important events are actions, and for a program to plan intelligently, it must be able to determine the effects of its own actions.

Consider the MYCIN domain as an example. The situation with which MYCIN deals includes the doctor, the patient and the illness. Since MYCIN's actions are advice to the doctor, full planning would have to include information about the effects of MYCIN's output on what the doctor will do. Since MYCIN doesn't know about the doctor, it might plan the effects of the course of treatment on the patient. However, it doesn't do this either. Its rules give the recommended treatment as a function of the information elicited about the patient, but MYCIN makes no prognosis of the effects of the treatment. Of course, the doctors who provided the information built into MYCIN considered the effects of the treatments.

Ignoring prognosis is possible because of the specific narrow domain in which MYCIN operates. Suppose, for example, a certain antibiotic had the precondition for its usefulness that the patient not have a fever. Then MYCIN might have to make a plan for getting rid of the patient's fever and verifying that it was gone as a part of the plan for using the antibiotic. In other domains, expert systems and other AI programs have to make plans, but MYCIN doesn't. Perhaps if I knew more about bacterial diseases, I would conclude that their treatment sometimes really does require planning and that lack of planning ability limits MYCIN's utility.

The fact that MYCIN doesn't give a prognosis is certainly a limitation.

For example, MYCIN cannot be asked on behalf of the patient or the administration of the hospital when the patient is likely to be ready to go home. The doctor who uses MYCIN must do that part of the work himself. Moreover, MYCIN cannot answer a question about a hypothetical treatment, e.g. "What will happen if I give this patient penicillin?" or even "What bad things might happen if I give this patient penicillin?".

2. Various formalisms are used in artificial intelligence for representing facts about the effects of actions and other events. However, all systems that I know about give the effects of an event in a situation by describing a new situation that results from the event. This is often enough, but it doesn't cover the important case of concurrent events and actions. For example, if a patient has cholera, while the antibiotic is killing the cholera bacteria, the damage to his intestines is causing loss of fluids that are likely to be fatal. Inventing a formalism that will conveniently express people's common sense knowledge about concurrent events is a major unsolved problem of AI.

3. The world is extended in space and is occupied by objects that change their positions and are sometimes created and destroyed. The common sense facts about this are difficult to express but are probably not important in the MYCIN example. A major difficulty is in handling the kind of partial knowledge people ordinarily have. I can see part of the front of a person in the audience, and my idea of his shape uses this information to approximate his total shape. Thus I don't expect him to stick out two feet in back even though I can't see that he doesn't. However, my idea of the shape of his back is less definite than that of the parts I can see.

4. The ability to represent and use knowledge about knowledge is often required for intelligent behavior. What airline flights there are to Singapore is recorded in the issue of the International Airline Guide current for the proposed flight day. Travel agents know how to book airline flights and can compute what they cost. An advanced MYCIN might need to reason that Dr. Smith knows about cholera, because he is a specialist in tropical medicine.

5. A program that must co-operate or compete with people or other programs must be able to represent information about their knowledge, beliefs, goals, likes and dislikes, intentions and abilities. An advanced MYCIN might need to know that a patient won't take a bad tasting medicine unless he is convinced of its necessity.

6. Common sense includes much knowledge whose domain overlaps that of the exact sciences but differs from it epistemologically. For example, if I spill the glass of water on the podium, everyone knows that the glass will

break and the water will spill. Everyone knows that this will take a fraction of a second and that the water will not splash even ten feet. However, this information is not obtained by using the formula for a falling body or the Navier-Stokes equations governing fluid flow. We don't have the input data for the equations, most of us don't know them, and we couldn't integrate them fast enough to decide whether to jump out of the way. This common sense physics is contiguous with scientific physics. In fact scientific physics is imbedded in common sense physics, because it is common sense physics that tells us what the equation $s = 0.5gt^2$ means. If MYCIN were extended to be a robot physician it would have to know common sense physics and maybe also some scientific physics.

It is doubtful that the facts of the common sense world can be represented adequately by production rules. Consider the fact that when two objects collide they often make a noise. This fact can be used to make a noise, to avoid making a noise, to explain a noise or to explain the absence of a noise. It can also be used in specific situations involving a noise but also to understand general phenomena, e.g. should an intruder step on the gravel, the dog will hear it and bark. A production rule embodies a fact only as part of a specific procedure. Typically they match facts about specific objects, e.g. a specific bacterium, against a general rule and get a new fact about those objects.

Much present AI research concerns how to represent facts in ways that permit them to be used for a wide variety of purposes.

3 COMMON SENSE REASONING

Our ability to use common sense knowledge depends on being able to do common sense reasoning.

Much artificial intelligence inference is not designed to use directly the rules of inference of any of the well known systems of mathematical logic. There is often no clear separation in the program between determining what inferences are correct and the strategy for finding the inferences required to solve the problem at hand. Nevertheless, the logical system usually corresponds to a subset of first order logic. Systems provide for inferring a fact about one or two particular objects from other facts about these objects and a general rule containing variables. Most expert systems, including MYCIN, never infer general statements, i.e. quantified formulas.

Human reasoning also involves obtaining facts by observation of the world, and computer programs also do this. Robert Filman did an interesting thesis on observation in a chess world where many facts that could be obtained by deduction are in fact obtained by observation. MYCIN's doesn't require this, but our hypothetical robot physician would have to draw conclusions from a patient's appearance, and computer vision is not ready for it.

An important new development in AI (since the middle 1970s) is the formalization of nonmonotonic reasoning.

Deductive reasoning in mathematical logic has the following property — called monotonicity by analogy with similar mathematical concepts. Suppose we have a set of assumptions from which follow certain conclusions. Now suppose we add additional assumptions. There may be some new conclusions, but every sentence that was a deductive consequence of the original hypotheses is still a consequence of the enlarged set.

Ordinary human reasoning does not share this monotonicity property. If you know that I have a car, you may conclude that it is a good idea to ask me for a ride. If you then learn that my car is being fixed (which does not contradict what you knew before), you no longer conclude that you can get a ride. If you now learn that the car will be out in half an hour you reverse yourself again.

Several artificial intelligence researchers, for example Marvin Minsky (1974) have pointed out that intelligent computer programs will have to reason nonmonotonically. Some concluded that therefore logic is not an appropriate formalism.

However, it has turned out that deduction in mathematical logic can be supplemented by additional modes of nonmonotonic reasoning, which are just as formal as deduction and just as susceptible to mathematical study and computer implementation. Formalized nonmonotonic reasoning turns out to give certain rules of conjecture rather than rules of inference — their conclusion are appropriate, but may be disconfirmed when more facts are obtained. One such method is *circumscription*, described in (McCarthy 1980).

A mathematical description of circumscription is beyond the scope of this lecture, but the general idea is straightforward. We have a property applicable to objects or a relation applicable to pairs or triplets, etc. of objects. This property or relation is constrained by some sentences taken as assumptions, but there is still some freedom left. Circumscription further constrains the property or relation by requiring it to be true of a minimal set of objects.

As an example, consider representing the facts about whether an object can fly in a database of common sense knowledge. We could try to provide axioms that will determine whether each kind of object can fly, but this would make the database very large. Circumscription allows us to express the assumption that only those objects can fly for which there is a positive statement about it. Thus there will be positive statements that birds and airplanes can fly and no statement that camels can fly. Since we don't include negative statements in the database, we could provide for flying camels, if there were any, by adding statements without removing existing statements. This much is often done by a simpler method — the *closed world assumption* discussed by Raymond Reiter. However, we also have exceptions to the general statement that birds can fly. For example, penguins, ostriches and birds with certain feathers removed can't fly. Moreover, more exceptions may be found and even exceptions to the exceptions. Circumscription allows us to make the known exceptions and to provide for additional exceptions to be added later — again without changing existing statements.

Nonmonotonic reasoning also seems to be involved in human communication. Suppose I hire you to build me a bird cage, and you build it without a top, and I refuse to pay on the grounds that my bird might fly away. A judge will side with me. On the other hand suppose you build it with a top, and I refuse to pay full price on the grounds that my bird is a penguin, and the top is a waste. Unless I told you that my bird couldn't fly, the judge will side with you. We can therefore regard it as a communication convention that if a bird can fly the fact need not be mentioned, but if the bird can't fly and it is relevant, then the fact must be mentioned.

References

Davis, Randall; Buchanan, Bruce; and Shortliffe, Edward (1977). Production Rules as a Representation for a Knowledge-Based Consultation Program, *Artificial Intelligence*, Volume 8, Number 1, February.

McCarthy, John (1960). Programs with Common Sense, *Proceedings of the Teddington Conference on the Mechanization of Thought Processes*, London: Her Majesty's Stationery Office. (Reprinted in this volume, pp. 000–000).

McCarthy, John and Patrick Hayes (1969). Some Philosophical Problems from the Standpoint of Artificial Intelligence, in B. Meltzer and D. Michie (eds), *Machine Intelligence 4*, Edinburgh University. (Reprinted in B. L.

Webber and N. J. Nilsson (eds.), *Readings in Artificial Intelligence*, Tioga, 1981, pp. 431–450; also in M. J. Ginsberg (ed.), *Readings in Nonmonotonic Reasoning*, Morgan Kaufmann, 1987, pp. 26–45; also in this volume, pp. 000–000.)

McCarthy, John (1980). Circumscription — A Form of Nonmonotonic Reasoning, *Artificial Intelligence*, Volume 13, Numbers 1,2. (Reprinted in B. L. Webber and N. J. Nilsson (eds.), *Readings in Artificial Intelligence*, Tioga, 1981, pp. 466–472; also in M. J. Ginsberg (ed.), *Readings in Nonmonotonic Reasoning*, Morgan Kaufmann, 1987, pp. 145–152; also in this volume, pp. 000–000.)

Minsky, Marvin (1974). *A Framework for Representing Knowledge, M.I.T. AI Memo 252.*

Shortliffe, Edward H. (1976). *Computer-Based Medical Consultations: MYCIN, American Elsevier, New York, NY.*

ANSWERS TO QUESTIONS DISCUSSION OF THE PAPER

QUESTION: You said the programs need common sense, but that's like saying, If I could fly I wouldn't have to pay Eastern Airliness $44 to haul me up here from Washington. So if the programs indeed need common sense, how do we go about it? Isn't that the point of the argument?

DR. MCCARTHY: I could have made this a defensive talk about artificial intelligence, but I chose to emphasize the problems that have been identified rather than the progress that has been made in solving them. Let me remind you that I have argued that the need for common sense is not a truism. Many useful things can be done without it, e.g. MYCIN and also chess programs.

QUESTION: There seemed to be a strong element in your talk about common sense, and even humans developing it, emphasizing an experiential component — particularly when you were giving your example of dropping a glass of water. I'm wondering whether the development of these programs is going to take similar amounts of time. Are you going to have to have them go through the sets of experiences and be evaluated? Is there work going on in terms of speeding up the process or is it going to take 20 years for a program from the time you've put in its initial state to work up to where it has a decent amount of common sense?

DR. MCCARTHY: Consider your 20 years. If anyone had known in 1963 how to make a program learn from its experience to do what a human does after 20 years, they might have done it, and it might be pretty smart by now. Already in 1958 there had been work on programs that learn from experience. However, all they could learn was to set optimal values of numerical parameters in the program, and they were quite limited in their ability to do that. Arthur Samuel's checker program learned optimal values for its parameters, but the problem was that certain kinds of desired behavior did not correspond to any setting of the parameters, because it depended on the recognition of a certain kind of strategic situation. Thus the first prerequisite for a program to be able to learn something is that it be able to represent internally the desired modification of behavior. Simple changes in behavior must have simple representations. Turing's universality theory convinces us that arbitrary behaviors can be represented, but they don't tell us how to represent them in such a way that a small change in behavior is a small change in representation. Present methods of changing programs amount to education by brain surgery.

QUESTION: I would ask you a question about programs needing common sense in a slightly different way, and I want to use the MYCIN program as an example.

There are three actors there — the program, the physician, and the patient. Taking as a criterion the safety of the patient, I submit that you need at least two of these three actors to have common sense.

For example if (and sometimes this is the case) one only were sufficient, it would have to be the patient because if the program didn't use common sense and the physician didn't use common sense, the patient would have to have common sense and just leave. But usually, if the program had common sense built in and the physician had common sense but the patient didn't, it really might not matter because the patient would do what he or she wants to do anyway.

Let me take another possibility. If only the program has common sense and neither the physician nor the patient has common sense, then in the long run the program also will not use the common sense. What I want to say is that these issues of common sense must be looked at in this kind of frame of reference.

DR. MCCARTHY: In the use of MYCIN, the physician is supposed to supply the common sense. The question is whether the program must also have common sense, and I would say that the answer is not clear in the MYCIN case. Purely computational programs don't require common sense, and none of the present chess programs have any. On the other hand, it seems clear that many other kinds of programs require common sense to be useful at all.

/@steam.stanford.edu:/u/ftp/jmc/someneed.tex: begun 1996 May 12, latexed 1996 May 12 at 1:24 p.m.