AI Magazine Volume 8 Number 4 (1987) (© AAAI)

What Is AI, Anyway?

Roger C. Schank

In this article, the scientific and technological goals of artificial intelligence, and a proposal of ten fundamental problems in AI research are discussed This article is an introduction to Scientific DataLink's microfiche publication of the Yale AI technical reports In this context, examples of research conducted at the Yale Artificial Intelligence Project relating to each of the research problems is presented.

Because of the massive, often quite unintelligible publicity that it gets, artificial intelligence is almost completely misunderstood by individuals outside the field. Even AI's practitioners are somewhat confused about what AI really is.

Is AI mathematics? A great many AI researchers believe strongly that knowledge representations used in AI programs must conform to previously established formalisms and logics, or the field is unprincipled and *ad hoc.* Many AI researchers believe that they know how the answer will turn out even before they have figured out what exactly the questions are. They know that some mathematical formalism or other must be the best way to express the contents of the knowledge which people have. Thus, to these researchers, AI is an exercise in the search for the proper formalisms to use in representing knowledge.

Is AI software engineering? A great many AI practitioners seem to think so. If you can put knowledge into a program, then this program must be an AI program. This conception of AI, derived as it is from much of the work going on in industry in expert systems, has served to confuse AI people tremendously about what the correct focus of AI ought to be and what the fundamental issues in AI are. If AI is just so much software engineering, if building an AI program primarily means the addition of domain knowledge such that a program knows about insurance or geology, for example, then what differentiates an AI program in insurance from any other computer program which works within the field of insurance? Under this conception, it is difficult to determine where software engineering leaves off and where AI begins.

Is AI linguistics? A great many AI researchers seem to think that building grammars of English and putting those grammars on a machine is AI. Of course, linguists have never thought of their field as having much to do with AI at all. However, as money for linguistics has begun to disappear and money for AI has increased, it has become increasingly convenient to claim that work on language which had nothing to do with computers at all has some computational relevance. Suddenly, theories of language that were never considered by their creators to be process models at all are now proposed as AI models.

Is AI psychology? Would building a complete model of human thought processes and putting it on a computer be considered a contribution to AI? Many AI researchers could not care less about the human mind, yet the human mind is the only kind of intelligence that we can reasonably hope to study. We have an existence proof. We know the human mind works. However, in adopting this view, one still has to worry about computer models that display intelligence but are clearly in no way related to how humans function. Are such models *intelligent?* Such issues inevitably force one to focus on the issue of the nature of intelligence apart from its particular physical embodiment.

In the end, the question of what AI is all about probably doesn't have just one answer. What AI is depends heavily on the goals of the researchers involved, and any definition of AI is dependent upon the methods that are being employed in building AI models. Last, of course, it is a question of results. These issues about what AI is exist precisely because the development of AI has not yet been complet-

ed. They will disappear entirely when a machine really is the way writers of science fiction have imagined it could be.

Most practitioners would agree on two main goals in AI The primary goal is to build an intelligent machine. The second goal is to find out about the nature of intelligence. Both goals have at their heart a need to define intelligence AI people are fond of talking about intelligent machines, but when it comes down to it, there is very little agreement about what exactly constitutes intelligence. It follows that little agreement exists in the AI community about exactly what AI is and what it should be. We all agree that we would like to endow machines with an attribute we really can't define Needless to say, AI suffers from a lack of definition of its scope.

One way to attack this problem is to attempt to list some features that we would expect an intelligent entity to have. None of these features would define intelligence, indeed a being could lack any one of them and still be considered intelligent. Nevertheless each attribute would be an integral part of intelligence in its way.

Let me list the features I consider to be critical and then briefly discuss them. They are communication, internal knowledge, world knowledge, intentionality, and creativity.

Communication

An intelligent entity can be communicated with. We can't talk to rocks or tell trees what we want, no matter how hard we try With dogs and cats we cannot express many of our feelings, but we can let them know when we are angry. Communication is possible with them. If it is difficult to communicate with someone, we might consider the person unintelligent If the communication lines are narrow with a person, if the individual can only understand a few ideas, we might consider this person unintelligent. No matter how smart your dog is, he can't understand when you discuss physics, which does not mean that the dog doesn't understand something about physics. You can't discuss physics with your pet rock either, but it doesn't understand physics at all. Your small child might know some physics, but discussions of this subject have to be put in terms the child can understand. In other words, the easier it is to communicate with an entity, the more intelligent it seems. Obviously, many exceptions exist to this general feature of intelligence, for example, people who are considered intelligent who are impossible to talk to. Nevertheless, this feature of intelligence is still significant, even if it is not absolutely essential.

Internal Knowledge

We expect intelligent entities to have some knowledge about themselves They should know when they need something, they should know what they think about something, and they should know that they know it. At present, probably only humans can do all this "knowing." We cannot really know what dogs know about what they know We could program computers to seem like they know what they know, but it would be hard to tell if they really did. To put this idea another way, we really cannot examine the insides of an intelligent entity in such a way as to establish what it actually knows. Our only choice is to ask and observe. If we get an answer that seems satisfying, then we tend to believe the entity we are examining has some degree of intelligence. Of course, this factor is another subjective criterion to be sure and a feature that when absent can signify nothing.

World Knowledge

Intelligence also involves being aware of the outside world and being able to find and utilize the information that one has about the outside world. It also implies having a memory in which past experience is encoded and can be used as a guide for processing new experiences. You cannot understand and operate in the outside world if you treat every experience as if it were brand new. Thus, intelligent entities must have an ability to see new experiences in terms of old ones. This statement implies an ability to retrieve old experiences that would have to have been codified in such a way as to make them available in a variety of different circumstances. Entities that do not have this ability can be momentarily intelligent but not globally intelligent. There are cases of people who are brain damaged who can do fine in a given moment but forget what they have done soon after. The same is true of simple machines which can do a given job but do not know that they have done it and have no ability to draw on this or other experiences to guide them in future jobs.

Intentionality

Goal-driven behavior means knowing when one wants something and knowing a plan to get what one wants. Usually, a presumed correspondence exists between the complexity of the goals that an entity has and the sheer number of plans which an entity has available to accomplish these goals. Thus, a tree has none or next to none of these plans and goals, a dog has somewhat more, and a person has quite a few; very intelligent people probably have more. Of course, sheer number of recorded plans would probably not be a terrific measure of intelligence. If it were, machines that met that criterion could easily be constructed. The real criterion with respect to plans has to do with interrelatedness of plans and their storage in a way that is abstract enough to allow a plan constructed for situation A to be adapted and used in situation B

Creativity

Finally, every intelligent entity is assumed to have some degree of creativity. Creativity can be defined weakly, including, for example, the ability to find a new route to one's food source when the old one is blocked. Of course, creativity can also mean finding a new way to look at something that changes one's world in some significant way. It certainly means being able to adapt to changes in one's environment and to be able to learn from experience. Thus, an entity that doesn't learn is probably not intelligent, except momentarily.

Now, as I said, one needn't have all these characteristics to be intelligent, but each is an important part of intel-

ligence. This statement having been made, where do current AI programs fit in? It seems clear that no AI model is too creative as yet, although various ideas have been proposed in this regard lately. It also seems clear that no AI models have a great deal of internal knowledge. In general, AI programs don't know what they know, nor are they aware of what they can do. They might be able to summarize a news wire, but they don't know that they are summarizing it.

However, programs that have goals and plans to accomplish these goals have been around since the inception of AI. Work on such programs has spawned a variety of ideas on how planning can be accomplished, particularly within the domain of problem solving. Programs that have external knowledge have usually not been considered part of AI at all. Database retrieval is not in any way connected with AI, although it has been clear to AI researchers that they must eventually concern themselves with how knowledge is best organized in order to have really intelligent machines Nevertheless, many programs for organizing and retrieving knowledge do, of course, exist

Programs that communicate with computers have been around as long as there have been computers, but this communication has been less than satisfactory. Most noncomputer professionals complain bitterly about the difficulty in getting a computer to do what you want, and of course, the computer industry has been responsive to this complaint, producing better and better interfaces. However, in the end, computers will not really be easy to use until they can see, hear, read, and generally understand what we say to them and what we want them to do. In AI, these subjects have always been considered important parts of the field, and much research has been done on them.

As AI became more commercialized, one would have imagined the parts of AI research that were the most advanced in terms of engineering would have become those areas where the commercial action would begin. But as often happens, salesmanship and market readiness often determine what gets sold. Thus, AI entered the world through the creation of so-called expert systems, which were engineering attempts to take some of the problem-solving and planning models that had been proposed in AI and give them real-world relevance The problem was that these experts lacked what I term internal knowledge and creativity. In addition, it is difficult to have an expert who doesn't know what it knows, how it came to know it, or how to adapt if circumstances are somewhat different than they were supposed to be. Most of all, experts with no memories are no experts at all.

[AI's] primary goal is to build an intelligent machine. The second goal is to find out about the nature of intelligence.

In part as a result of the commercialization of expert systems, equating AI with expert systems in the public eye, and in part as a result of the usual battles AI has always faced with older fields of inquiry that relate to it, AI is in a serious state of disruption

Most AI people seem to have chosen one of two routes to get them out of their state of confusion. The first of these routes I call the applications route. In this view of AI, the job is to build real working systems. Whether these systems are AI loses its import as one begins to work on them The problem is to make them work at all, not to be a purist about what is or is not AI. As anyone who has ever worked on a large software engineering program knows, this task is so complex that it makes all other problems pale by comparison. Making big programs work is hard. When they are finished are they AI? Does it matter?

The second route is what I call the scientific route. This route sounds good in principle, and it has as its premise a desire to avoid the commercialization of AI and work only on impossible problems such as the brain, or neat problems such as logic. Let the applications route people do as they will, the scientific route people have chosen simply to ignore them and bolt the door.

Thus, without actually deciding to do so, the AI community has made a decision. Either one defines AI as a modern methodological tool now being used in the ancient enterprise of the study of mind, *the scientific answer*, or one's definition of AI is, in essence, the *applications answer*, namely an attempt to create a certain new computer technology that relates to some behaviors previously done only by humans.

This division seems fine in principle; many fields have a scientific, theoretical group and an applications group that derives its work from the scientific work This situation would be nice in AI too if this were the case. What actually is the case is that the scientific workers are, for the most part, concerned with issues which are far away from potential applications. In addition, the applications folk have been busy applying results from earlier days that are known to be seriously inadequate, which does not mean that they are not building useful applications; sometimes they are. It does mean that for all intents and purposes, the two routes have nothing to do with each other.

One problem with the applications answer is that it is very imprecise. Is all new computer technology to be labeled AI? Certainly, if one reads the advertisements in the computer magazines, it is easy to believe that AI is anything anyone says it is; there is no definition. However, to an AI researcher (as opposed to someone involved in an AI business), only a small fraction of the advances in computer software and hardware seem to qualify as advances in AI. The technology that AI people want to create usually involves solving some fundamental problem, the nature of what kinds of elements are part of a computer program. Further, it usually means getting a machine to do what previously only humans have done (rather than simply improving existing techniques). The problem with this

definition has been obvious to AI people for some time. As soon as something radically new has been accomplished and computers have done it, this achievement is no longer uniquely human and, thus, no longer AI. One question that needs to be answered on the technological side is, "Can some definition about the nature of AI software be made such that under all circumstances, it will be seen as uniquely part of, or derived from, AI?"

What is really the case is that it is not possible to clearly define which pieces of new software are AI and which are not. In actuality, AI must have an issues-related definition. In other words, people do arithmetic and so do computers. The fact is, however, that no one considers a program which calculates to be an AI program, nor would they, even if the program calculated in exactly the way people do. The reason this is so is that calculation is not seen as a fundamental problem of intelligent behavior and that computers are already better at calculation than people are. This two-sided definition, based on the perception of the fundamental centrality of an issue with respect to its role in human intelligence, and the practical viewpoint of how good current computers are already at accomplishing a task constitute how one defines whether a given problem is legitimately an AI problem. For this reason, much of the good work in AI has been just answering the question of what the issues are.

To put this argument another way, what AI is is defined not by the methodologies used in AI but by the problems attacked by these methodologies. A program is not an AI program because it uses Lisp or Prolog certainly. By the same token, a program is not an AI program because it uses some form of logic or if-then rules. Expert systems are only AI programs if they attack some AI issue. A rule-based system is not an AI program just because it uses rules or was written with an expert system shell. It is an AI program if it addresses an AI issue.

One factor about AI issues, though, is that they change. What was an issue yesterday might not be one today. Similarly, the issues that I believe to be critical today might disappear 10 years from now. Given that this is the case, defining AI by issues can make AI a rather odd field with a constantly changing definition. However, some problems will endure:

1. Representation
2. Decoding
3. Inference
4. Control of Combinatorial Explosion
5. Indexing
6. Prediction and Recovery
7. Dynamic Modification
8. Generalization
9. Curiosity
10. Creativity

Representation

Probably the most significant issue in AI is the old problem of the representation of knowledge. "What do we know, and how do we get a machine to know it?" is the central issue in AI. An AI program or theory that makes a statement about how knowledge ought to be represented which is of a generality greater than the range of knowledge covered by the program itself is a contribution to AI.

Our early work in natural language processing focused on representation issues. We began with conceptual dependency to represent primitive actions (Schank 1972). At Yale, we developed other knowledge structures, such as scripts and plans, for representing larger conceptual entities (Schank and Abelson 1977; Cullingford 1978; Wilensky 1978). We designed a system of social acts for representing the actions of social institutions, such as governments (Schank and Carbonell 1978). As we became concerned with the role of memory in cognitive processing, we developed memory organization packets (MOPs) (Schank 1979; Lebowitz 1980; Kolodner 1980; Dyer 1982), as a refinement of our previous work. Our current work focuses on the process of explanation; again we have developed a new representation system—explanation patterns (Schank 1986).

Decoding

It is of no use to have a nice knowledge representation if there is no way to translate from the real world into this representation. In natural language, or vision systems, for example, decoding is often the central problem in constructing an AI program. Sometimes, of course, the decoding work is so difficult that the programmers forget to concern themselves with what they are decoding into, that is, what the ideal representation ought to be, so they make the work harder for themselves. Deciding the representation of a given fact, that it is predicate calculus or syntactic phrase markers, for example, can complicate the problem, relegating the decoding work to some other, often nonexistent program.

Our work in natural language has required that we have programs which convert natural language into whatever internal representation system we use. These programs are called *parsers* or *conceptual analyzers* (Riesbeck and Schank 1976; Schank, Lebowitz, and Birnbaum 1978; Gershman 1979; Birnbaum and Selfridge 1979). Recent developments include direct memory access parsing which couples the parsing process to memory itself (Riesbeck and Martin 1985).

Inference

Information is usually more than the sum of its parts. Once we decode a message (visual, verbal, symbolic, or whatever), we must begin extracting the content of this message. Usually, the content is much more than has been expressed directly. We don't say every nuance of what we mean. We expect our hearer to be smart enough to figure some of it out. Similarly, we must attempt to figure out the significance of what we have seen, making assumptions about what it all means. This problem is called *inference.*

Human memory is highly inferential, even about prior experiences and retrieval of information. People are capable of answering questions from incomplete data. They can figure out if they should know something and whether they might be able to figure it out. Such self-awareness depends strongly upon an ability to know how the world works in general— the representation problem again. Building a program that knows if it would know a thing is a very important task.

Inference has always been at the heart of our natural language programs. The script applier mechanism (SAM) (Cullingford 1978) made inferences based on expectations from stereotypical events. Schank (1978b) provides a detailed history of the role of inference in our early research

Control of Combinatorial Explosion

Once you allow a program to make assumptions beyond what it has been told about what may be true, the possibility that it could go on forever doing this assuming becomes quite real. At what point do you turn off your mind and decide that you have thought enough about a problem? Arbitrary limits are just that, arbitrary. It seems a safe assumption that there is a structure to our knowledge which guides the inference process. Knowing what particular knowledge structure we are in while processing can help us determine how much we want to know about a given event; that is, contexts help narrow the inference process. Many possible ways exist to control the combinatorics of the inference process: deciding among them and implementing them is a serious AI problem if the combinatorial explosion is first started by an AI process.

The scripts in SAM and Frump (DeJong 1979) provided one means of controlling inference and directing search. The goal trees in Politics (Carbonell 1979) were another means. We also suggested "interestingness" as a method for focusing inferences (Schank 1978a), which was applied in the program IPP (Lebowitz 1980)

Indexing

It is all well and good to know a great deal, but the more you know, the harder it should be to find what you know. Along these same lines, the most knowledgeable person on earth should also be the slowest to say anything Such statements are called the paradox of the expert in psychology. They are paradoxes precisely because they are untrue. Obviously, people must have ways of organizing their knowledge so that they can find what they need when they need it. Originally, this problem was called the search problem in AI However, viewed as a search problem, the implication was that faster search methods were what was needed. This fact would imply that experts were people who searched their databases quickly, which seems quite absurd. It is the organization and labeling of memory and episodes in memory that is the key issue here. For any massive system, that is, for any real AI system, indexing is a central and, possibly, the central problem. AI programs are usually not large enough to make their answers to the indexing question meaningful, but the construction of programs of the appropriate size should become more important in the years ahead.

Frump, with its dozens of scripts for newspaper stories, was one of the first Yale programs to have enough knowledge to make indexing an issue Cyrus (Kolodner 1980) focused on the specific issue of organization and indexing of memory. Our MOPs representation provided a means of testing various indexing strategies (Schank 1979, Schank 1982). Dyer's Boris program had numerous types of knowledge structures that had to be accessed (Dyer 1982)

Prediction and Recovery

Any serious AI program should be able to make predictions about how events in its domain will turn out. This ability is what understanding really means, that is, knowing to some extent what is coming When these predictions fail, which they certainly must in any realistic system, an intelligent program should not only recover from the failure, but it must explain the failure That is, programs must understand their own workings well enough to know what an error looks like and be able to correct the rule that caused this error in addition to being able to recognize the situation when it occurs again To explain, a computer should be able, by use of the same basic scientific theory, to do an adequate job of forecasting stocks or weather or playing a game of chess or coaching a football team What I mean by "the same basic theory" is that the theory of prediction, recovery from error, error explanation, and new theory creation should be identical in principle, regardless of domain.

Most of our natural language programs trigger numerous expectations at many levels The programs must be able to reject or substantiate their predictions (DeJong 1979, Granger 1980, Lytinen 1984)

Dynamic Modification

AI practitioners went through a long period of trying to find out how to represent knowledge We needed to find out what was learned before we could even consider working on learning itself However, most of us have always wanted to work on learning. Learning is, after all, the quintessential AI issue What makes people interesting, what makes them intelligent is that they learn. People change with experience. The trouble with almost all the programs which we have written is that they are not modified by their experiences. No matter how sophisticated a story understander might seem, it loses all credibility as an intelligent system when it reads the same story three times in a row and fails to get mad or bored or even to notice. Programs must change as a result of their experiences, or they will not do anything very interesting.

Similarly, any knowledge structures, or representations of knowledge that AI researchers create, no matter how adequately formulated initially, must change over time. Understanding how they are changed by actual use during the course of processing information is one of the major problems in representation itself. Deciding when to create a new structure or abandon an old one is a formidable problem. Thus, new AI programs should be called upon to assimilate information and change the nature of the program in the course of this assimilation. Clearly, such programs are necessary before the knowledge-acquisition problem can be adequately attacked It should also be clear that an AI program which cannot build itself gradually (not requiring that all its knowledge stuffed in at the beginning), is not really intelligent.

I will now give a definition of AI that most of our programs will fail: AI

is the science of endowing programs with the ability to change themselves for the better as a result of their own experiences The technology of AI is derived from the science of AI and is, at least for now, unlikely to be intelligent. However, it should be the aim of every current AI researcher to endow programs with this kind of dynamic intelligence

One of the first learning programs at Yale was Selfridge's Child In recent years, learning or adaptive behavior has been the standard. Cyrus and IPP served as prototypes for the current era.

AI is the science of endowing programs with the ability to change themselves for the better as a result of their own experiences.

Generalization

A program that can form a generalization from experience and can be tested would be of great significance. This program would have to be able to draw conclusions from disparate data. The key aspect of a good generalization maker is the ability to connect experiences that are not obviously connectable. This element is the essence of creativity. A key AI problem, therefore, is to understand new events and make predictions about future events by generalizing from prior events. These generalizations would likely be inadequate at first, but eventually new theories that fit the data should emerge. Ultimately, human expertise is embodied not in rules but in cases. People can abstract rules about what they do, of course, but the essence of their expertise, that part which is used in the most complex cases, is derived from particular and rather singular cases that stand out in their minds. The job of the expert is to find the most relevant case to reason from in any given instance. Phenomena such as reminding enhance this ability to generalize by providing additional data to consider. The very consideration of seemingly irrelevant data makes for a good generalizer. In other words, AI programs should be able to come up with ideas on their own.

Again, IPP provided a model for generalization. The recent work of Riesbeck (Riesbeck 1983) and Bain (Bain 1984) shows new ways to explore generalization and its application to new situations.

Curiosity

Cats, small children, and most adults are curious. They ask questions about what they see, wonder about what they hear, and object to what they are told This curiosity is not so wondrous when we realize that once a system makes predictions, these predictions might fail, and the system should wonder why. The ability to wonder why, to generate a good question about what is going on, and the ability to invent an answer, to explain what has gone on to oneself, is at the heart of intelligence. We would accept no human who failed to wonder or explain as very intelligent. In the end, we will have to judge AI programs by the same criteria.

Beyond simply coming up with generalizations by noticing similarities, a program should also explain why the observed behavior should be so. We developed a theory of explanation in a series of technical reports (Schank 1984a, Schank 1984b, Schank and Riesbeck 1985, Schank 1985) and in a recent book (Schank 1986). We have also recognized that curiosity, as the underlying stimulus for learning, would be better exploited in education itself, particularly in the application of computers to education (Schank and Slade 1985).

Creativity

Scientists and technologists would both agree that what is most fascinating of all is the possibility that computers will someday surpass human beings They are most likely to achieve this goal by being creative in some way Principles of creativity, combined with the other powers of the computer, are likely to create this ultimate fantasy. To this end, I believe it is necessary for AI people to become familiar with work in other fields that bears on this issue Issues such as consciousness and development relate here also. Thus, relating ideas in AI to those in allied fields with the purpose of coming to some new scientific conclusions is an important task.

Tale-Spin, one of the earliest Yale AI programs, created stories (Meehan 1976). The key to Tale-Spin's creativity was an understanding of goal interactions among the characters in the Aesoplike stories. Most recently, the program Chef (Hammond 1984) exercised creativity in quite another domain: cooking. Chef created recipes to account for novel combinations of ingredients. Like Tale-Spin, Chef's creativity was guided by its knowledge of the interaction among the ingredients, specifically by understanding how the various elements interact and serve to satisfy numerous culinary goals.

Which Problems Are Most Important?

All these problems are important, of course, but one thing above all: an AI program that does not learn is no AI program. Now, I understand that this maxim would not have made much sense in the past. However, one of the problems of defining AI is, as I have said, that AI could, by past definitions, be nearly anything. We have reached a new stage. We have a much better idea of what is learned; therefore, it is time to demand learning of our programs. AI programs have always been a promise for the future, a claim about what we could build someday Each thesis has been the prototype of what we might build if only we would. Well, from the technological perspective, the time to build is now. From the scientific perspective, after the issue of what is learned is taken care of, the issue for AI is learning, although we probably don't have to wait to finish with the first issue in order to start on the second.

In principle, AI should be a contribution to a great many fields of study. AI has already contributed some to psychology, linguistics, and philosophy as well as other fields. AI is, potentially, the algorithmic study of

processes in every field of inquiry. As such, the future should produce AI anthropologists, AI doctors, AI political scientists, and so on There might also be some AI computer scientists, but on the whole, I believe, AI has less to say, in principle, to computer science than to any other discipline. The reason that this statement has not been true heretofore is an accident of birth. AI people have been computer scientists: therefore, they have tended to contribute to computer science Computer science has needed tools, as has AI, and on occasion, these tools have coincided. AI is actually a methodology applicable to many fields. It is just a matter of time until AI becomes part of other fields and that the issue of what constitutes a contribution to AI will be reduced to the question of what constitutes a contribution in the allied field. At that time, what will remain of AI will be precisely the issues which transcend these allied fields, whatever they might be. In fact this statement might be the best available working definition of what constitutes a successful contribution to AI today, namely, a program whose inner workings apply to similar problems in areas completely different from the one that was originally tackled.

In some sense, all subjects are really AI. All fields discuss the nature of man. AI tries to do something about it. From a technological point of view, AI matters to the extent that its technology matters, which is a hard question to answer. However, from a scientific point of view, we are trying to answer the only questions that really do matter.

Acknowledgments

This work was supported in part by the Defense Advanced Research Projects Agency and the Office of Naval Research under contract N00014-85-K-0108, the Air Force Office of Scientific Research under contract AFOSR-85-0343, and the National Library of Medicine under contract 1-R01-LM04251.

References

Bain, W M 1984 Toward a Model of Subjective Interpretation, Technical Report, 324, Dept of Computer Science, Yale Univ

Birnbaum, L , and Selfridge, M 1979 Problems in Conceptual Analysis of Natural Language, Technical Report, 168, Dept of Computer Science, Yale Univ

Carbonell, J. 1979 Subjective Understanding: Computer Models of Belief Systems Ph D. diss , Technical Report, 150, Dept of Computer Science, Yale Univ

Cullingford, R. 1978 Script Application: Computer Understanding of Newspaper Stories Ph D. diss , Technical Report, 116, Dept. of Computer Science, Yale Univ

DeJong, G 1979. Skimming Stories in Real Time: An Experiment in Integrated Understanding Ph D diss., Technical Report, 158, Dept. of Computer Science, Yale Univ

Dyer, M. 1982. In-Depth Understanding: A Computer Model of Integrated Processing for Narrative Comprehension, Technical Report, 219, Dept of Computer Science, Yale Univ

Gershman, A. 1979 Knowledge-Based Parsing Ph D. diss , Technical Report, 156, Dept. of Computer Science, Yale Univ.

Granger, R H 1980 Adaptive Understanding: Correcting Erroneous Inferences Ph D diss., Technical Report, 171, Dept of Computer Science, Yale Univ

Hammond, K 1984 Indexing and Causality: The Organization of Plans and Strategies in Memory, Technical Report, 351, Dept. of Computer Science, Yale Univ

Kolodner, J. L 1980 Retrieval and Organizational Strategies in Conceptual Memory: A Computer Model Ph D. diss , Technical Report, 187, Dept of Computer Science, Yale Univ

Lebowitz, M 1980. Generalization and Memory in an Integrated Understanding System Ph.D. diss , Technical Report, 186, Dept of Computer Science, Yale Univ

Lytinen, S 1984 The Organization of Knowledge in a Multilingual, Integrated Parser Ph D diss , Technical Report, 340, Dept of Computer Science, Yale Univ

Meehan, J 1976. The Metanovel: Writing Stories by Computer Ph D. diss , Technical Report, 74, Dept of Computer Science, Yale Univ

Riesbeck, C. K 1983 Knowledge Reorganization and Reasoning Style, Technical Report, YALEU/DCS/RR 270, Dept of Computer Science, Yale Univ.

Riesbeck, C K , and Martin, C. E. 1985 Direct Memory Access Parsing, Technical Report, 354, Dept of Computer Science, Yale Univ

Riesbeck, C , and Schank, R C 1976 Comprehension by Computer: Expectation-Based Analysis of Sentences in Context, Technical Report, 78, Dept of Computer Science, Yale Univ.

Schank, R C 1986 *Explanation Patterns Understanding Mechanically and Creatively.* Hillsdale, N J.: Lawrence Erlbaum Associates

Schank, R C 1985 Questions and Thought, Technical Report, 385, Dept of Computer Science, Yale Univ

Schank, R C 1984a. Explanation: A First Pass, Technical Report, 330, Dept of Computer Science, Yale Univ

Schank, R. C 1984b. The Explanation Game, Technical Report, 307, Dept. of Computer Science, Yale Univ

Schank, R C 1982 *Dynamic Memory A Theory of Learning in Computers and People* Cambridge, Mass.: Cambridge University Press.

Schank, R C. 1979 Reminding and Memory Organization: An Introduction to MOPs, Technical Report, 170, Dept. of Computer Science, Yale Univ

Schank, R C. 1978a Inference in the Conceptual Dependency Paradigm: A Personal History, Technical Report, 141, Dept. of Computer Science, Yale Univ

Schank, R C. 1978b Interestingness: Controlling Inferences, Technical Report, 145, Dept of Computer Science, Yale Univ

Schank, R C 1972. Conceptual Dependency: A Theory of Natural Language Understanding *Cognitive Psychology* 3(4): 552-631

Schank, R C , and Abelson, R 1977. *Scripts, Plans, Goals, and Understanding* Hillsdale, N J : Lawrence Erlbaum Associates

Schank, R C , and Carbonell, J. 1978. Re: The Gettysburg Address Representing Social and Political Acts, Technical Report, 127, Dept. of Computer Science, Yale Univ.

Schank, R C , and Riesbeck, C 1985 Explanation: A Second Pass, Technical Report, 384, Dept of Computer Science, Yale Univ

Schank, R C , and Slade, S 1985 Education and Computers: An AI Perspective, Technical Report, 431, Dept of Computer Science, Yale Univ

Schank, R. C ; Lebowitz, M ; and Birnbaum, L 1978 Integrated Partial Parsing, Technical Report, 143, Dept of Computer Science, Yale Univ

Selfridge, M 1980 A Process Model of Language Acquisition Ph D diss., Technical Report, 172, Dept of Computer Science, Yale Univ

Wilensky, R 1978. Understanding Goal-Based Stories. Ph D diss , Technical Report, 140, Dept of Computer Science, Yale Univ