MACHINE LEARNING: CHALLENGES OF THE 80'S, (EDITED TRANSCRIPTS OF A PANEL DISCUSSION)

by

R. S. Michalski
S. Amarel
D. Lenat
D. Michie
P. H. Winston

Machine Learning: An Artificial Intelligence Approach Vol. II, R.S. Michalski, T.J. Carbonell and T.M. Mitchell (Eds.), Los Altos, CA, pp. 27-41, Morgan Kaufmann Publishers, 1986.

MACHINE LEARNING

An Artificial Intelligence Approach
Volume II

Contributors:

Saul Amarel
John R. Anderson
Ranan B. Banerji
Robert C. Berwick
Gary L. Bradshaw
Mark H. Burstein
Jaime G. Carbonell
Gerald DeJong
Nachum Dershowitz
Thomas G. Dietterich
Kenneth D. Forbus
Jean-Gabriel Ganascia
Dedre Gentner
John H. Holland
Smadar T. Kedar-Cabelli
Yves Kodratoff
Patrick Langley
Michael Lebowitz
Douglas B. Lenat
Sridhar Mahadevan
Ryszard S. Michalski
Donald Michie
Allen Newell
J. Ross Quinlan
Paul S. Rosenbloom
Claude Sammut
Bernard Silver
Herbert A. Simon
Robert E. Stepp III
Gail E. Thornburg
Paul E. Utgoff
Patrick H. Winston
Jan M. Zytkow

Editors:

Ryszard S. Michalski
University of Illinois
at Urbana-Champaign, IL

Jaime G. Carbonell
Carnegie-Mellon University
Pittsburgh, PA

Tom M. Mitchell
Rutgers University
New Brunswick, NJ

MORGAN KAUFMANN
PUBLISHERS, INC.
95 FIRST STREET, LOS ALTOS, CALIFORNIA 94022

2

MACHINE LEARNING:

Challenges of the Eighties

Edited transcript of a panel discussion held at the 2nd International Machine Learning Workshop, Allerton House, University of Illinois, June 22–24, 1983

Ryszard S. Michalski
(Moderator)

Saul Amarel

Douglas B. Lenat

Donald Michie

Patrick H. Winston

Transcribed and edited by
Gail Thornburg and
Ryszard S. Michalski

Michalski:

Now that our Workshop is coming to a close, it is time to summarize our thoughts and discuss some of the issues important to our field. To start with, let me raise some questions particularly suitable for discussion. First of all, what are the most important tasks for machine learning research for the near and not-so-near future? Next, what is the role of machine learning in AI? How important and how feasible is automated knowledge acquisition for expert systems? Shouldn't we stress this area much more than we stressed it in the past?

Another interesting issue is the role of domain-specific versus general approaches to machine learning. As you know, for a long time many researchers avoided research that could be called general methods of learning. It was believed that such research was not going to bring any interesting results, that the resulting systems would be very inefficient, and that the whole area of general learning was a hopeless task.

As a result of such attitudes, many researchers started to explore learning issues in the context of specific problems. This domain-specific research led to interesting results and impressive learning systems. There was, however, a bad side to it; it led to a situation in which certain groups worked in their little niches, deeply involved with their favorite domain-specific problems and not communicating sufficiently with other groups. They often developed their own terminology, unaware that it was more or less isomorphic to the terminology of some other groups, and this hampered interaction and the progress of the field. Moreover, that kind of domain-specific research didn't lead to any more general understanding of the problems of the field and didn't lead to any new theories or principles. Certainly the time has come for us to identify more general principles in our field.

A related issue for discussion is the question of pure versus applied machine learning research. Should we stress the theoretical research, or should we be more oriented toward designing and implementing practical learning systems?

Another issue to consider is whether we should continue to study in depth single learning strategies, or whether we should now attempt to build integrated learning systems that employ several strategies. Clearly the existing embodiment of a learning system, a human being, can learn using a variety of strategies simultaneously. Moreover, the strategies that people use change with their accumulation of knowledge. We know that children learn differently than adults do. The major difference is that adults already have a large store of knowledge about the external world and so can use this knowledge when they learn new things. Therefore, using strategies involving analogy is generally more appropriate for them than for children, who don't yet have much knowledge and thus cannot use analogical reasoning to the same extent.

Another important topic for discussion concerns the terminology and description languages useful for machine learning research. I have already mentioned that one of the problems we face is that researchers in this field use a variety of terminologies, some of them isomorphic. Identification of some of those isomorphisms would be very useful for the further progress of the field and would make it easier for new researchers to enter the field. An interesting problem related to this is the role of formal languages such as predicate calculus in our field. What is their usefulness for representing program-generated knowledge versus their usefulness as well-defined formalisms for describing learning algorithms.

Finally, are we already a field? If so, what are our unifying principles and our goals? And one more issue: We are now facing the development of new computer architectures, so we may study learning problems using machines to be developed—connection machines, Boltzmann machines, and so forth. Will this new development in computer architectures bring us some important new tools that will help us in machine learning research?

I will end on this question and introduce to you our panelists: Saul Amarel, Rutgers University; Donald Michie, University of Edinburgh; Doug Lenat, Stanford University; and Patrick Winston, MIT. They will air their own views on these and other topics in this order. At the end, using my prerogative as moderator, I will make some final comments.

Amarel:

I understand that what we're supposed to talk about is a view of the field five to ten years from now. Ryszard has put many questions to us, and I'm going to address some of them.

First of all, on the issue of pure research versus applied research, I think we need both. I think we need both in parallel, and I think we should do both even on the same project. This has been my own philosophy for some time. We have to develop both parts of the activity. We need applications, specific explorations, to generate ideas about how to approach new problems and new challenges. We need the basic research part in order to tie things together and to see what is equivalent to what, what is superior to what, and why. We also need to see how things relate to each other as well as to other areas of AI and computer science, or to psychology.

My own sense is that problems of theory formation—and some problems of learning—are no different from other issues in the area of problem solving. I would very much like to see an integration and an overall framework to encompass problems in all domains, including domains of theory formation. To date, we have put a tremendous amount of emphasis on derivation problems, on pathfinding problems, on interpretation problems, and on other areas that we call problem solving. There are some areas of problem solving that have to do with constraint satisfaction, where the number of constraints to be satisfied is typically large; also, we have problems in which we must handle a large number of interacting goals. If you look carefully at the problem methodologies used in these areas, you will find them not very different from methodologies that we use in some areas of hypothesis generation, theory formation, and learning. I think we should do a little more work on relating these various areas to each other, from two points of view: what the problem formulation is and what the problem-solving processes are.

I don't think we should split these issues. We could of course split the problem-solving domain into subdomains, in accordance with certain ideas having to do with the methodology of solution, complexity, or the degree of dependence between problem conditions. In my mind the degree of dependence between problem conditions is probably the most fundamental parameter for thinking about different methods of problem solving. In many of the derivation problems—that is, many of the problems that are usually conceived as conventional problem solving—we usually work in one space. Here we talk about moving from state to state, of using operators, inference rules, and so on. The more you delve into theory formation problems, the more you see that you have to work in more than one space. In most instances we work in two spaces: in the space of solutions and in the space of problem conditions. And most of the time the difficulty arises because the languages and the sets of concepts that are available to us in these two spaces are different.

The best strategy we can then pursue, which indeed we pursue in a very intuitive way, is to try to establish early enough a link between one space and the other and to formulate the problem in just one space as much as possible, in order to be able to solve it with well-known methods. The entire issue of how to handle problems involving both a solution space and a problem condition space, as well as how to link the spaces and how to coordinate the two-space search, has been with us for some time. This was recognized in the early 1970s by several people; one might cite a very good paper by Simon and Lea in this area.[1] I think this issue will require more thinking.

As I said, I think the basic issue in choosing a problem-solving method is really the question of problem decomposability. If we can decompose a problem, we can assign a method to a goal, independently of other goals, making our problem-solving activity relatively easy. In most of the problem-solving efforts in AI we have taken this kind of approach. As soon as goals become very interdependent, we cannot reason very clearly from problem conditions to goals, and we have a difficult problem. This is what creates a major difficulty in problems of theory formation and learning: it's very difficult to decompose the problems. To the extent that we are able to decompose, or to the extent that we are going to be good in handling methodologies of decomposition, we will also improve our ability to develop effective methodologies for formation problems.

I was very interested in the many comments and the considerable amount of work being done in the area of analogical reasoning. Personally, I would very much like to have a moratorium on the term *analogy,* because it is regarded in somewhat different ways by different people. That's a problem that we have in the field, anyway—trying to determine how to use such terms as *intelligence* or *learning.* Since these terms have different connotations for different people, agreement is extremely difficult to accomplish. We have to be a little more precise in operational terms, in defining what we mean when we talk about a particular analogical reasoning methodology, and so on. As I was listening to various approaches, my own sense was as follows: As Ryszard was saying, studying learning in an environment with a tremendous amount of relevant knowledge is very important. Most of our learning is being done

[1]H. A. Simon and G. Lea, "Problem Solving and Rule Induction: A Unified View," in *Knowledge and Cognition,* L. Gregg (Ed.), Erlbaum, Hillsdale, N. J., 1974.

in a situation in which we know of many problems and of ways of solving them; we know the structures and methods in other domains, and in some way we try to import that knowledge and bring it to bear on the problem at hand.

Possibly the entire issue of analogy could be subsumed under the following mechanism: Given a problem, we must find some other problem known to us that is somehow similar to the initial problem. We use the "similar" problem as a focusing mechanism for selecting schemata that are promising, to start at least a part of the problem-solving activity. We import the "similar" problem, we use the method that worked for that problem, and then we go on to do a different kind of problem solving in order to complete our task. I cannot possibly conceive how analogical reasoning alone can do the entire job. The most difficult part is not the identification of the analogy but the assimilation of the analogy, the repair, and the additional work needed in order to finish the job, *after* the analogy has been imported.

And I would very much like to see more work done on the use of analogical reasoning and the use of repair strategies to finish a piece of analogical reasoning as a basis for solving a new problem. Also, in terms of theory formation, my recent work shows that the most difficult aspect of the problem is not encountered in the early stages of theory formation. The difficulty is at the end of the process, when you have "almost correct" theories and would like to converge on a solution. The amount of reasoning that is needed then is enormous, and the techniques used are much more demanding, which is precisely where I think domain knowledge in large quantities must come into play. I don't see how we can form theories in certain areas without already knowing quite a bit about the area itself. This is essentially Bill Martin's postulate, that you must know quite a bit about a domain if you want to do some learning in it. I think it's absolutely essential.

As for applications, I certainly think that we need much more work in the area of learning, in the creation of knowledge bases for expert systems. There's no doubt that this is the only way we can go. The field is asking for it, not only in terms of the expert systems of today, where we have a thousand rules or more, but also in terms of situations—and these are both fundamental problems and application problems—where a system is already working, but we would like to identify subdomains of the system where, on the basis of the active experience of a system, things could be done even better. We would like to be able to identify specialized methods in any given subdomain, as human experts do, and make them available to that particular subdomain. The entire movement from novice or average performance to expert performance requires this identification of special characteristics of subdomains and special methodologies that could be applied to them, so that the result will be increased expertise. The entire area of expertise acquisition in the context of expert systems is extremely interesting for learning, both in terms of basic kinds of problems and in terms of impact on the building of expert systems.

There are more specific things I want to say about current projects. It concerns me that I don't see any projects of the Meta-DENDRAL type around us today. What are we going to have after Meta-DENDRAL? This was a major, interesting, well-chosen domain, with very interesting challenges, where one could develop many ideas about theory formation. I would like us to find ways whereby any given group that did not have the interest and the staffing to continue such a project could develop arrangements so that some other group could continue the project (I know all the difficulties involved) and perhaps build on the experience of the first group, to try to move beyond the stage at which, for instance, Meta-DENDRAL was left. I would very much like to see more scientific theory formation going on, maybe in the biological sciences, or perhaps in areas of physics. This would be extremely important for us to pursue.

Now there is another area of learning that I find might be very useful for us and might relate a little more closely to psychological investigations. This has to do with developing environments for problem solving with appropriate graphics and monitoring capabilities, where one could watch the operation of an expert in a domain. It could be a designer (e.g., a designer of an engineering system), another kind of professional such as a manager, and at the same time it could give various aids to that professional. I am not talking yet about an automatic system. Rather, those professionals will have aids, and at the same time they will have monitoring facilities to record what they are doing. It would be an excellent thing to capture, in some gradual fashion, some of the ways in which those managers or professionals or designers do things—and learn how people actually do things—by using these kinds of environments. This has implications for experimental environments and facilities. That's where I can see those psychologists who are interested in human-machine interactions or in learning generally interacting with involved nonpsychologists.

A final thought: I have a feeling that if we want to advance in the area of concept discovery and in the area of theory formation in a deep way, then we should be doing more things of the kind that Doug Lenat did in the AM system. This means that we should be thinking *not* only about one specific learning problem but about a cluster of interconnected learning problems. The output of one can in some way be utilized as an input to another and can in some cases enable us to come back and revise ideas about concepts that we have been using in a component problem. Isolated, very simple formation or learning activities are very important to our understanding of some of the basic methodology. Yet how much more interesting it would be to have a set of activities in a specific domain—such as mathematics or physics—in which we could see how the various activities interact. No scientists work exclusively on one problem at one time. They always work on several problems, and they transfer knowledge from problem to problem; in addition, of course, they bring to each problem considerable knowledge from outside the problem domain.

At this point I think I should end my comments and let my other colleagues on the panel speak. Thank you.

Michie:

I should like to endorse a theme that I take from Saul Amarel's remarks, namely, the anchoring effect of a good choice of problem. AI work is now at a crossroads. More accurately, it is at a Y-junction of the kind that experimental psychologists like to use to test animals. One arm of the Y-junction leads to a philosophy of free-floating work. The other leads to a sense of direction derived from well-defined, hard problems.

I see an analogy here to the early days of aeronautics. The use of balloons offers a useful caricature of the free-floating school. In a balloon one is happy to float wherever the wind blows and to exchange anecdotes with other balloonists about interesting glimpses of whatever terrain one happens to pass over.

The really hard problem in aeronautics was that of directional flight, which confronted the heavier-than-air school. Unfortunately, in every branch of systematic inquiry the free-floating approach has a fatal attraction for the administrators of science. They feel they really understand that kind of thing. So the first people to venture into the directional styles must not be too surprised if the political and administrative leaders of society give them a hard time and seek to coax or deflect them into unstructured explorations in which all concerned can relax.

Let me tell you about this as it worked out in the case of powered flight. Early experiments met success in the hands of two very hard-headed, scientifically trained engineers, the Wright brothers, followed almost immediately afterwards by Cody in the United Kingdom. By about 1908–1909 an infant technology had taken root, in many ways comparable to the infant technology of intelligent, computer-controlled robotics that characterized the late 1960s and the early 1970s in the AI field. The balloonists were still pottering on, and they were more successful than the heavier-than-air people in the higher reaches of science/political wisdom.

The British prime minister set up a subcommittee of the Imperial Defense Committee, chaired by Lord Escher. This committee worked for a few months in late 1908 and reported early in 1909. They took a variety of evidence from officers and politicians of the defense establishment on whether the heavier-than-air principle had a future.

After finely sifting the evidence, they came to the unanimous conclusion that it did not. The subcommittee's recommendation to the Imperial Defense Committee was that all work on heavier-than-air flight should be canceled and government resources redeployed to the study of balloons.

The prime minister of the time, H. H. Asquith, is on record in the minutes of the Imperial Defense Committee as pronouncing himself highly satisfied with this decision. Shortly after this, Bleriot flew the Channel, attracting a great deal of publicity. By good chance, Lord Escher was an intelligent man of high integrity. He began to worry that possibly his committee had made a terrible error. After further thought and study, he put a heavily documented case to the prime minister, to the effect that his committee had made a mistake and that it was in the national interest for Britain to arm herself with an effective fleet of heavier-than-air machines.

There is a moral in this story for artificial intelligence. The kind of work that was being done at SRI on the SHAKEY project was a typical hard problem on which all the intellectual and other resources of the AI craft had to be brought to bear to establish success. Along with similar projects at Stanford, MIT, and Edinburgh, this investigation into world modeling, recognition, and planning had to be discontinued because the world around us can understand coffee table talk about these topics, but it is repelled and mystified by sustained and detailed experiment.

Yet as far as our professional criteria are concerned, there is no way out but to select hard problems to act as forcing functions. The fact that a free-floating, liberal arts approach can warm the hearts of administrators should be taken not as a positive rallying point but as a point against.

In terms of practice, what does this mean? Our field, which is infant still and lacks a hardened skeleton on which to hang a definite morphology, needs a style of practice determined by the professional standards and rules of evidence customary in experimental and theoretical science. It should aim to supersede the standards and rules of evidence customary in the liberal arts and in some of the less developed engineering disciplines such as computer science.

For the future, let me humbly suggest that our next meeting be restricted to papers that report on completed results. Any philosophizing about future work that they may additionally contain will then at least be accompanied by a directional point of reference. In well-established branches of science, no one would consider operating to any more permissive criteria.

Lenat:

Now let me start off by agreeing, in a way, with something that Donald said earlier, that the field of machine learning can be anchored by working on hard, specific, very well-defined problems. In fact, I think that's the reason the field has looked at-anchor for so long.

Assuming that we want to progress from our relatively primitive state of technology, I think we will have to send out small craft and hope that some of them do make it back safely. More seriously, though, if we do want anchors of the kind that Donald was talking about, something that we can use productively, then those anchors should be the sources of power that our programs tap into and that we tap into in our research.

The first source of power is synergy. *Synergy* means getting out more than you put in, in dealings between programs and human beings. In the work we do with EURISKO—for example, the toy naval ship design—are things that neither we nor the program alone could do, and it's the human-machine synergy that I think we're

really tapping. We're exploiting and technically combining the different capabilities of each of us. Then, of course, there's synergy between the programs we build and the work that other AI researchers do in areas other than machine learning. Putting learning modules onto the front of expert systems, for instance, is that sort of synergy. And finally, there is synergy with other machine learning researchers, so that we can get our programs to cooperate, work together, argue together. That's something that by and large has not been tapped, but I think it is a source of power just waiting there to be exploited.

The next source is analogy, with two types of uses, though Amarel wants to see that word banned. One use is to generate plausible, potentially true conjectures, ideas, conceptualizations, and ways of looking at the world, as well as ways to explore them (perhaps through other means) to see if they really are true.

The second use for analogy is in knowledge acquisition, for instance, in getting material into the knowledge base of an expert system. We do this all the time, by looking around for a unit or a frame or a rule that is similar to what we want to enter, getting the unit, copying it, and editing it. While the "copy and edit" process is a trivial kind of analogizing, less trivial analogizing would presumably provide less trivial kinds of knowledge acquisition aids.

The next source of power is heuristics, and I have nothing more to say about that right now. In case you aren't familiar with this, you can see Jerry DeJong's puzzle [a word puzzle distributed to all conference participants] for a clear definition of what it's good for, or my 1984 *AI Journal* articles.

Next is representation. Again, there are two issues here. One is having and finding natural representations, which I think is very important. The other is changing representations, also very important, one of the kinds of things that Saul mentioned.

Finally there is a certain catchall category, in which we find things like parallelism, morphological analysis, sources of power we haven't discovered yet.

What I really want to focus on is what we can do to exploit these sources of power in the coming decade. If we carry this exercise further, then somewhere up at the very top level would be the goals that we are trying to achieve—but I'm not interested in that high a level. At an intermediate level, one concern is the human-machine interface. This is something that can tap into the human-machine synergy, obviously. In the human-machine interface there are several aspects of concern. One of them is I/O, but that's not really part of our business—that's for people in hardware or other areas of AI to worry about. These are things like having snazzy forms of Ivan Sutherland's old helmet you can wear—that is, sunglasses that project separate CRT images on the inside of each eye; accelerometers that sense your head, neck, and eye movements, so that as you turn your head the scene changes in real time; nice things like that mean you're not limited by small screen sizes in what your display area can be. Of course with the hat it's natural to want accoutrements like gloves that sense your hands. . . . Anyway, we're not going to worry about that—but somebody else will.

Then, obviously, we'll want things like natural language and speech recognition and—remembering that we'll have these gloves and these funny glasses on—we might as well start using nonverbal cues as well. Again, let's let someone else worry about that, but keep in mind, it's going to happen.

The thing we *can* do to exploit the synergy with human beings is to start thinking about models of sessions at a terminal between a person and the program that's running, or in fact, models of individual people. One way to do that is to start taxonomizing sessions and taxonomizing groups of individuals; so, for instance, you know that if a user starts a sentence with the word *let* that user must be a mathematician—and we treat mathematicians differently from human beings.

Michie: The user could be a priest.

Lenat: Yes, but I suspect we would treat him as a phenomenon similar to a mathematician.

Next we have the synergy with the other AI researchers and their programs. The way we foster that is to build our systems as portable modules that can be plugged into various other sorts of systems. Similarly, if we build those modules in a very clean way so that they can plug into each other, then we can start getting synergy among various learning programs. This is one of the main directions that I see the work in EURISKO taking in the future. We're going to try to clean it up and get it into a form I can give to the world to look at or use, depending on the audience.

As for you-know-what [analogy], we could do the generation of plausible hypotheses that we'd like using it, if only we had a broad enough knowledge base. I think the thing that's held us back is that if the program only knows about plane geometry and has to come up with an analogy, it must be an analogy from plane geometry. What lets people do analogizing, or generation and exploitation of metaphors, so effectively is the enormous range of knowledge we have. This is not so much the depth but the breadth of knowledge, several orders of magnitude more than any system's program has. The kinds of tasks to work on there include putting an encyclopedia on-line, not in a textual sense, but actually in a knowledge base, so that it can be used. There is a group at Atari working with Alan Kay and myself doing that kind of project with roughly a 13- to 15-year time frame. There's an article on it at IJCAI this summer [1983], if you're interested.

Besides putting in encyclopedic facts about the world, you want to add commonsense facts about the world. Let's say there are a thousand "fact words" in basic English, maybe another thousand or two that should have been there. And if you're going to do such a project in any finite amount of time, you won't want to take the approach that Pat Hayes took with water, spending several years and doing it right. Instead you must take a day or so, think about what a two-year-old child knows about water, write it down, and go on to the next word. I think that's the right tack to take for ten years, to see what happens. Imagine—we get a little bit of knowledge about each of several thousand commonsense concepts into a program, and then, parenthetically, technical knowledge, the kind of knowledege one would find in expert systems.

If we had that for several different domains in one place, that might also lead to exploiting analogy.

Just as we want to have a broad knowledge base to exploit analogy, we want a broad heuristic base to exploit the power of heuristics and the way they organize. You need to consider all the world's heuristics, which you might do by looking at thousands of specific heuristics and starting to generalize them very slowly, to build up some huge tree of heuristics.

Once you have that kind of heuristic base you can tap into analogy and other things. And again, I think that this is something that's doable, just barely, in the coming decade—a moderate job, not a complete job.

And then I even have the nerve to put representation bases on our "list of knowledge to accumulate," though we only have about eight representations we know about—so you might as well have a program that knows about them and can choose among them and occasionally even augment them. I see that as one of the real opportunities for work in the future.

Another thing—if you took a look at the paper I have in the proceedings for this conference, you'll have noticed that I talk a lot about cognitive economy—programs that model, monitor, and modify themselves. That's another way of tapping into both change of representation and of heuristics.

Nowhere in my talk is there anything about theory, and so—since someone will probably ask—we *can* fit theory in here if you want. You can talk about work on the nature of learning, and if you do, then what you're really looking into is a kind of synthesis of all the things that are going on elsewhere in the picture, plus perhaps some idea of what's happening with human beings. And notice all the work on human cognition—just a very narrow fragment of what people could be working on. Why is that? Why not worry about societies, organizations, and machines instead of just organisms? Why not worry about organisms, why just worry about cognition? Learning goes on, on different time scales: by hours rather than minutes at the immune system level, by years at the corporate level, or even over millenia via evolution.

The final kind of learning theory that I think is worth doing is dimensions of learning, the kinds of things that Michalski, Carbonell, and Mitchell talk about in their chapter in *Machine Learning I.* It's very useful because it lets you do morphological analysis. You can start by saying, here are the ten dimensions along which learning systems can be categorized. Observe that all the systems that we've built so far cluster here and here and here, yet there are vast areas of the space that aren't populated by any system. Why is that?

Thinking about those sorts of issues can lead you to new insights about what's less easy and hard and why. Or, occasionally, they can lead you to say that someone ought to build a system that has these properties.

Before closing, let me respond to a couple of things that other people have said earlier. One of them related to the role of machine learning in AI. I see a kind of coroutine role, just as if I said, "Gee, I'd really like to have natural language and speech recognition modules in my learning system." Then suppose the natural language people say, "Gee, I'd really like to have a learning module in my natural language system." Those are both reasonable things to say, and I envision a kind of symbiosis—*maybe* synergy—developing there.

As for pure versus applied research, I think labels like that are a kind of red herring. When we build expert systems, for instance, the real problem in getting consensus among experts is that they have slightly different meanings for the terms they use. All the time gets wasted in arguments that involve mere syntactical disagreement. I think the same thing is likely to happen in the pure/applied issue, with a lot of time wasted arguing about various categories of what should and shouldn't be done. Let's not worry so much about terminology.

As for individual versus integrated learning systems, I think that integrated systems are almost going to be a necessity. Again, looking back over AI, we see lots of individual mechanisms that were originally *the* chief source of power in programs—Perceptrons, automatic backtracking, unification, and resolution. In all those cases, people got real excited and they developed programs that had this or that mechanism as their sole source of power, and they had some initial early successes. Then they "lost big" three to five years down the road, and everyone got turned off and went into other fields. Seven to ten years later, we started coming back with a new perspective, saying "Hey, these are really neat things to use as sub-sub-submechanisms all through our programs." The same thing is going to happen here, I think; we should start integrating before we find ourselves at the wrong end of another seven-year backlash.

Winston:

I propose to make a claim that we are faced with one danger and one opportunity, but first I want to say that I think things are basically in pretty good shape. This was a splendid meeting, an uplifting one in comparison with most I attend. It seems to me that there are four reasons why we should be happy to be in the position we are in.

Reason number one is that the research that we're doing is well balanced. The various dimensions—practical, applied, special, general, formal, informal—all look good to me in the sense that the dimensions are generally well balanced.

The second point is that we're not diverted much, in comparison with other parts of the field. I don't see very many people doing what I would consider silly recursions into noncritical problems. We don't see very many papers claiming that a new control structure is essential *before* we can do any work on learning, or that we have to invent new programming languages before we can get started. Some

say we need these things, of course, but there's no general sense of futility for lack of some tool.

The third thing is that there are very few, if any, "snake oil salespeople" making ridiculous promises. If you look at other parts of artificial intelligence, that's not the case. We have an obligation to continue to insure that our little subfield maintains that kind of distinction.

Fourth, I think we're doing basically the right things. It's not just a matter of generalized balance, but the fact that there are now some new things that weren't being done, that should have been done, and that are, in fact, beginning to get done. We now see papers on guessing and confirming structure, work on quality and process, and new reflections on what we can say about the educational process. Those are all important pursuits that happily are now under way.

Now let me go back to my original two-point list of one danger and one opportunity. The danger is that our field is about to become too successful. If I may invoke a precedent and try to build an analogy, I would not be at all surprised to find that our specialty is going the way of expert systems. That is, vast public interest, hundreds of spin-off companies, depletion of university resources, industrial raids—the whole works. The usual corruption that all of that brings is a serious danger that I think we have to face. And I don't know what to do about it.

We could very well become the banner part of AI in the next few years, it seems to me, displacing expert systems. Again we need to go back to a kind of social pressure, I suppose, to insure that we're not corrupted by that popularity.

The opportunity has to do with ambition fueled by hardware. When I was doing computer vision, it was unthinkable to consider anything other than running a 3-by-3 operator over a 256-by-256 image. That was a procrustean bed on which to lie, and we didn't get very far as a consequence. So, what's the analog of that today? It *might* be fooling around with a single story instead of hundreds of stories.

What I'm driving at is that we have all this interest in supercomputers lying out there, and I don't think we've thought much about how to exploit it. I'm willing to argue both sides of this question, of course. I don't think we should dash off and build machines for learning. On the other hand, I think it's worth a little more thought than has gone into it so far—that is, thought about what we could do with supermachines by way of superuses. Doug Lenat of course thinks about using lots of machines, but very few of us, if any, have thought about how we might use unbelievably fast machines to do unbelievably quick matching and operations of that sort.

So, I think that's the opportunity. If things work out as they did in vision, then we'll be thinking much better thoughts in ten years as a consequence. Now it's laughable to run a 3-by-3 operator over *anything;* you view vision problems much differently. When you can run 30-by-30 operators over 1000-by-1000 pictures in a quarter of a second, you change the nature of your thinking. Similarly, when we can make hundreds of analogies, not just a few, our perspective will shift increasingly toward a more global view.

Michalski:

Thank you, Patrick. I find myself very much in agreement with other panelists; maybe I should feel a little disappointed.

Winston: I'll argue with you.

Michalski:

But there are a few issues that I feel were insufficiently discussed. One is theory: Are we anywhere close to building something that could be called the computational theory of learning? Well, certainly we are at an early stage in our research, and we cannot say that we are in a position to build any complete general theory of learning.

Still, I think that work toward a computational theory of learning has some good points. I am not saying that we should all go to work on a theory of learning, naturally, but I think a few of us should give some thought to this issue. Theory can give us a better understanding of the relationships existing among different directions and different techniques, can clean up terminology, and, most of all, can help us to teach the subject of machine learning. When we have a clean theory, although not complete—and actually not even "clean"—we can more easily discuss our problems and build upon what we have already done. We can also identify isomorphisms and relationships between different concepts and different methods.

I could identify several groups in the past whose methods were almost equivalent except for the terminologies they used. So while there was the appearance of something new and different, in substance it was not necessarily so. Another thing that I would like to argue—and I probably will be a minority here—is that there is a need for research on what could be termed *multi-purpose methods* (or *generic task* methods).

What do I mean by such multi-purpose methods? I don't mean methods that are, shall we say, quintessentially general techniques of learning, applicable to all problems. Rather, I mean the following: A certain sufficiently broad yet cohesive subdomain of problems is identified, and then an effort is made to develop a learning method applicable to any problem that falls into such a domain.

These multi-purpose methods can be equipped with knowledge of a particular domain to which the method applies, so this would not be a "knowledge-free" method. However, if the method is sufficiently robust, it could be adapted easily to a range of tasks, so we would not have just one method per task. That's my idea of a multi-purpose method.

You may ask for examples, and there are examples of such methods. The easiest thing for me to say is that some of the methods that we developed in Illinois, like those implemented in INDUCE, AQ11, or CLUSTER programs, represent, in my opinion, that kind of work. They are not general in the sense that they can solve every kind of problem, but they are multi-purpose as they can be useful for a range of problems of a specific type, occurring in many applications. In other words, if a problem satisfies certain constraints and criteria, then the method can be applied.

Finally, I was somewhat surprised that so few papers in this workshop were devoted to the area of knowledge acquisition for expert systems. We know that knowledge acquisition is the bottleneck in the development of AI systems—in particular, of expert systems. Using current methods, this painstaking process may take years, and I believe that we as researchers in machine learning should devote new efforts to this area. Although we had few talks on this subject, it is certainly an important research direction to explore.

With these remarks, I propose to close our discussion. To all panelists I extend the warmest thanks for their contribution.