WHY PEOPLE THINK COMPUTERS CAN'T

Marvin Minsky, MIT

First published in AI Magazine, vol. 3 no. 4, Fall 1982. Reprinted in Technology Review, Nov/Dec 1983, and in The Computer Culture, (Donnelly, Ed.) Associated Univ. Presses, Cranbury NJ, 1985

Most people think computers will never be able to think. That is, really think. Not now or ever. To be sure, most people also agree that computers can do many things that a person would have to be thinking to do. Then how could a machine seem to think but not actually think? Well, setting aside the question of what thinking actually is, I think that most of us would answer that by saying that in these cases, what the computer is doing is merely a superficial imitation of human intelligence. It has been designed to obey certain simple commands, and then it has been provided with programs composed of those commands. Because of this, the computer has to obey those commands, but without any idea of what's happening.

Indeed, when computers first appeared, most of their designers intended them for nothing only to do huge, mindless computations. That's why the things were called "computers". Yet even then, a few pioneers -- especially Alan Turing -- envisioned what's now called "Artificial Intelligence" - or "AI". They saw that computers might possibly go beyond arithmetic, and maybe imitate the processes that go on inside human brains.

Today, with robots everywhere in industry and movie films, most people think Al has gone much further than it has. Yet still, "computer experts" say machines will never really think. If so, how could they be so smart, and yet so dumb?

========== CAN MACHINES BE CREATIVE? ==========

We naturally admire our Einsteins and Beethovens, and wonder if computers ever could create such wondrous theories or symphonies. Most people think that creativity requires some special, magical "gift" that simply cannot be explained. If so, then no computer could create - since anything machines can do (most people think can be explained.

To see what's wrong with that, we must avoid one naive trap. We mustn't only look at works our culture views as very great, until we first get good ideas about how ordinary people do ordinary things. We can't expect to guess, right off, how great composers write great symphonies. I don't believe that there's much difference between ordinary thought and highly creative thought. I don't blame anyone for not being able to do everything the most creative people do. I don't blame them for not being able to explain it, either. I do object to the idea that, just because we can't explain it now, then no one ever could imagine how creativity works.

We shouldn't intimidate ourselves by our admiration of our Beethovens and Einsteins. Instead, we ought to be annoyed by our ignorance of how we get ideas - and not just our "creative" ones. Were so accustomed to the marvels of the unusual that we forget how little we know about the marvels of ordinary thinking. Perhaps our superstitions about creativity serve some other needs, such as supplying us with heroes with such special qualities that, somehow, our deficiencies seem more excusable.

Do outstanding minds differ from ordinary minds in any special way? I don't believe that there is anything basically different in a genius, except for having an unusual combination of abilities, none very special by itself. There must be some intense concern with some subject, but that's common enough. There also must be great proficiency in that subject; this, too, is not so rare; we call it craftsmanship. There has to be enough self-confidence to stand against the scorn of peers; alone, we call that stubbornness. And certainly, there must be common sense. As I see it, any ordinary person who can understand an ordinary conversation has already in his head most of what our heroes have. So, why can't "ordinary, common sense" - when better balanced and more fiercely motivated - make anyone a genius,

So still we have to ask, why doesn't everyone acquire such a combination? First, of course, it sometimes just the accident of finding a novel way to look at things. But, then, there may be certain kinds of difference-in-degree. One is in how such people learn to manage what they learn: beneath the surface of their mastery, creative people must have unconscious administrative skills that knit the many things they know together. The other difference is in why some people learn so many more and better skills. A good composer masters many skills of phrase and theme - but so does anyone who talks coherently.

Why do some people learn so much so well? The simplest hypothesis is that they've come across some better ways to learn! Perhaps such "gifts" are little more than tricks of "higher-order" expertise. Just as one child learns to re-arrange its building-blocks in clever ways, another child might learn to play, inside its head, at Fe-arranging how it learns!

Our cultures don't encourage us to think much about learning. Instead we regard it as something that just happens to us. But learning must itself

consist of sets of skills we grow ourselves; we start with only some of them and and slowly grow the rest. Why don't more people keep on learning more and better learning skills? Because it's not rewarded right away, its payoff has a long delay. When children play with pails and sand, they're usually concerned with goals like filling pails with sand. But once a child concerns itself instead with how to better learn, then that might lead to exponential learning growth! Each better way to learn to learn would lead to better ways to learn - and this could magnify itself into an awesome, qualitative change. Thus, first-rank "creativity" could be just the consequence of little childhood accidents.

So why is genius so rare, if each has almost all it takes? Perhaps because our evolution works with mindless disrespect for individuals. I'm sure no culture could survive, where everyone finds different ways to think. If so, how sad, for that means genes for genius would need, instead of nurturing, a frequent weeding out.

========== PROBLEM SOLVING. ==========

We can hardly expect to be able to make machines do wonders before we find how to make them do ordinary, sensible things. The earliest computer programs were little more than simple lists and loops of commands like "Do this. Do that. Do this and that and this again until that happens". Most people still write programs in such languages (like BASIC or FORTRAN) which force you to imagine everything your program will do from one moment to the next. Let's call this "do now" programming.

Before long, Al researchers found new ways to make programs. In their "General Problem Solver" system, built in the late 1950's- Allen Newell, J.C.Shaw and Herbert A.Simon showed ways to describe processes in terms of statements like "If the difference between what you have and what you want is of kind D, then try to change that difference by using method M." This and other ideas led to what we call "means-ends" and "do if needed" programming methods. Such programs automatically apply rules whenever they're needed, so the programmers don't have to anticipate when that will happen. This started an era of programs that could solve problems in ways their programmers could not anticipate, because the programs could be told what sorts of things to try, without knowing in advance which would work. Everyone knows that if you try enough different things at random, eventually you can do anything. But when that takes a million billion trillion years, like those monkeys hitting random typewriter keys, it's not intelligence -- just Evolution. The new systems didn't do things randomly, but used "advice" about what was likely to work on each kind of problem. So, instead of wandering around at random, such programs could sort of feel around, the way you'd climb a hill in the dark by always moving up the slope. The only trouble was a tendency to get stuck on smaller peaks, and never find the real mountain tops.

Since then, much Al research has been aimed at finding more "global" methods, to get past different ways of getting stuck, by making programs take larger views and plan ahead. Still, no one has discovered a "completely general" way to always find the best method -- and no one expects to.

Instead, today, many Al researchers aim toward programs that will match patterns in memory to decide what to do next. I like to think of this as "do something sensible" programming. A few researchers -- too few, I think -- experiment with programs that can learn and reason by analogy. These programs will someday recognize which old experiences in memory are most analogous to new situations, so that they can "remember" which methods worked best on similar problems in the past.

========== CAN COMPUTERS UNDERSTAND? ==========

Can we make computers understand what we tell them? In 1965, Daniel Bobrow wrote one of the first Rule-Based Expert Systems. It was called "STUDENT" and it was able to solve a variety of high-school algebra "word problems"., like these:

The distance from New York to Los Angeles is 3000 miles. If the average speed of a jet plane is 600 miles per hour, find the time it takes to travel from New York to Los Angeles by jet.

Bill's father's uncle is twice as old as Bill's father. Two years from now I Bill's father will be three times as old as Bill. The sum of their ages is 92.
Find Bill's age.

Most students find these problems much harder than just solving the formal equations of high school algebra. That's just cook-book stuff -- but to solve the informal word problems, you have to figure out what equations to solve and, to do that, you must understand what the words and sentences mean. Did STUDENT understand? It used a lot of tricks. It was programmed to guess that "is" usually means "equals". It didn't even try to figure out what "Bill's fathers' uncle" means -- it only noticed that this phrase resembles "Bill's father". It didn't know that "age" and "old" refer to time, but it took them to represent numbers to be put in equations. With a couple of hundred such word-trick-facts, STUDENT sometimes managed to get the right answers.

Then dare we say that STUDENT "understands" those words? Why bother. Why fall into the trap of feeling that we must define old words like "mean" and "understand"? It's great when words help us get good ideas, but not when they confuse us. The question should be: does STUDENT avoid the "real meanings" by using tricks?

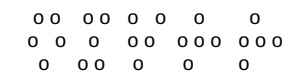
Or is it that what we call meanings really are just clever bags of tricks. Let's take a classic thought-example, such as what a number means. STUDENT obviously knows some arithmetic, in the sense that it can find such sums as "5 plus 7 is 12". But does it understand numbers in any other sense - say, what 5 "is" - or, for that matter, what are "plus" or "is"? What would ‹say if I asked you, "What is Five"? Early in this century, the philosophers Bertrand Russell and Alfred North Whitehead proposed a new way to define numbers. "Five", they said, is "the set of all possible sets with five members". This set includes each set of five ball-point pens, and every litter of five kittens. Unhappily, it also includes such sets as "the Five things you'd least expect" and "the five smallest numbers not included in this set" -- and these lead to bizarre inconsistencies and paradoxes. The basic goal was to find perfect definitions for ordinary words and ideas. But even to make the idea work for Mathematics, getting around these inconsistencies made the Russell-Whitehead theory too complicated for practical, common sense, use. Educators once actually tried to make children use this theory of sets, in the "New Mathematics" movement of the 1960's; it only further set apart those who liked mathematics from those who dreaded it. I think the trouble was, it tried to get around a basic fact of mind: what something means to me depends to some extent on many other things I know.

What if we built machines that weren't based on rigid definitions? Wont they just drown in paradox, equivocation, inconsistency? Relax! Most of what we people "know" already overflows with contradictions; still we survive. The best we can do is be reasonably careful; let's just make our machines that careful, too. If there remain some chances of mistake, well, that's just life.

========== WEBS OF MEANING. ==========

If every meaning in a mind depends on other meanings in that mind, does that make things too ill-defined to make a scientific project work? No, even when thing go in circles, there still are scientific things to do! Just make new kinds of theories - about those circles themselves! The older theories only tried to hide the circularities. But that lost all the richness of our wondrous human meaning-webs; the networks in our human minds are probably more complex than any other structure Science ever contemplated in the past. Accordingly, the detailed theories of Artificial Intelligence will probably need, eventually, some very complicated theories. But that's life, too.

Let's go back to what numbers mean. This time, to make things easier, well think about Three. I'm arguing that Three, for us, has no one single, basic definition, but is a web of different processes that each get meaning from the others. Consider all the roles "Three" plays. One way we tell a Three is to recite "One, Two, Three", while pointing to the different things. To do it right, of course, you have to (i) touch each thing once and (ii) not touch any twice. One way to count out loud while you pick up each object and remove it. Children learn to do such things in their heads or, when that's too hard, to use tricks like finger-pointing. Another way to tell a Three is to use some Standard Set of Three things. Then bring ‹set of things to the other set, and match them I one-to-one: if all are matched and none are left, then there were Three. That "standard I Three" need not be things, for words like "one, two, three" work just as well. For Five we have a wider choice. One can think of it as groups of Two and Three, or One and Four. Or, one can think of some familiar shapes -. a pentagon, an X, a Vee, a cross, an aeroplane; they all make Fives.



Because each trick works in different situations, our power stems from being able to shift from one trick to another. To ask which meaning is correct - to count, or match, or group - is foolishness. Each has its uses and its ways to support the others. None has much power by itself, but together they make a versatile skill-system. Instead of flimsy links in chain of definitions in the mind, each word we use can activate big webs of different ways to deal of things, to use them, to remember them, to compare them, and so forth. With multiply-connected knowledge-nets, you can't get stuck. When any sense of meaning fails, you can switch to another. The mathematician's way, once you get into the slightest trouble, you're stuck for good!

Why, then, do mathematicians stick to slender chains, each thing depending as few things as is possible? The answer is ironic: mathematicians want to get stuck! When anything goes wrong, they want to be the first to notice it. The best way to be sure of that is having everything collapse at once! To them, fragility is not bad, because it helps them find the perfect proof, lest any single thing they think be inconsistent with any other one. That's fine for Mathematics; in fact, that's what much of mathematics is. It's just not good Psychology. Let's

face it, our minds will always hold some beliefs that turn out wrong.

I think it's bad psychology, when teachers shape our children's mathematics into long, thin, fragile, definition tower-chains, instead of robust cross-connected webs. Those chains break at their weakest links, those towers topple at the slightest shove. And that's what happens to a child's mind in mathematics class, who only takes a moment just to watch a pretty cloud go by. The purposes of ordinary people are not the same as those of mathematicians and philosophers, who want to simplify by having just as few connections as can be. In real life, the best ideas are cross-connected as can be. Perhaps that's why our culture makes most children so afraid of mathematics. We think we help them get things right, by making things go wrong most times! Perhaps, instead, we ought to help them build more robust networks in their heads.

========== CASTLES IN THE AIR. ==========

The secret of what something means lies in the ways that it connects to all the other things we know. The more such links, the more a thing will mean to us. The joke comes when someone looks for the "real" meaning of anything. For, if something had just one meaning, that is, if it were only connected to just one other thing, then it wold scarcely "mean" at all!

That's why I think we shouldn't program our machines that way, with clear and simple logic definitions. A machine programmed that way might never "really" understand anything -- any more than a person would. Rich, multiply-connected networks provide enough different ways to use knowledge that when one way doesn't work, you can try to figure out why. When there are many meanings in a network, you can turn things around in your mind and look at them from different perspectives; when you get stuck, you can try another view. That's what we mean by thinking!

That's why I dislike logic, and prefer to work with webs of circular definitions. Each gives meaning to the rest. There's nothing wrong with liking several different tunes, each one the more because it contrasts with the others. There's nothing wrong with ropes - or knots, or woven cloth - in which each strand helps hold the other strands together - or apart! There's nothing very wrong, in this strange sense, with having all one's mind a castle in the air!

To summarize: of course no computer could understand anything real -- or even what a number is - if forced to single ways to deal with them. But neither could a child or philosopher. So such concerns are not about computers at all, but about our foolish quest for meanings that stand by themselves, outside any context. Our questions about thinking machines should really be questions about our own minds.

========== ARE HUMANS SELF-AWARE? ==========

Most people assume that computers can't be conscious, or self-aware; at best they can only simulate the appearance of this. Of course, this assumes that we, as humans, are self-aware. But are we? I think not. I know that sounds ridiculous, so let me explain.

If by awareness we mean knowing what is in our minds, then, as every clinical psychologist knows, people are only very slightly self-aware, and most of what they think about themselves is guess-work. We seem to build up networks of theories about what is in our minds, and we mistake these apparent visions for what's really going on. To put it bluntly, most of what our "consciousness" reveals to us is just "made up". Now, I don't mean that we're not aware of sounds and sights, or even of some parts of thoughts. I'm only saying that we're not aware of much of what goes on inside our minds.

When people talk, the physics is quite clear: our voices shake the air; this makes your ear-drums move -- and then computers in your head convert those waves into constituents of words. These somehow then turn into strings of symbols representing words, so now there's somewhere in your head that "represents" a sentence. What happens next?

When light excites your retinas, this causes events in your brain that correspond to texture, edges, color patches, and the like. Then these, in turn, are somehow fused to "represent" a shape or outline of a thing. What happens then?

We all comprehend these simple ideas. But there remains a hard problem, still. What entity or mechanism carries on from there? We're used to saying simply, that's the "self". What's wrong with that idea? Our standard concept of the self is that deep inside each mind resides a special, central "self" that does the real mental work for us, a little person deep down there to hear and see and understand what's going on. Call this the "Single Agent" theory. It isn't hard to see why every culture gets attached to this idea. No matter how ridiculous it may seem, scientifically, it underlies all principles of law, work, and morality. Without it, all our canons of responsibility would fall, of blame or virtue, right or wrong. What use would solving problems be, without that myth; how could we have societies at all?

The trouble is, we cannot build good theories of the mind that way. In

every field, as Scientists we're always forced to recognize that what we see as single things - like rocks or clouds, or even minds - must sometimes be described as made of other kinds of things. We'll have to understand that Self, itself, is not a single thing.

========== NEW THEORIES ABOUT MINDS AND MACHINES. ==========

It is too easy to say things like, "Computer can't do (xxx), because they have no feelings, or thoughts". But here's a way to turn such sayings into foolishness. Change them to read like this. "Computer can't do (xxx), because all they can do is execute incredibly intricate processes, perhaps millions at a time". Now, such objections seem less convincing -- yet all we did was face one simple, complicated fact: we really don't yet know what the limits of computers are. Now let's face the other simple fact: our notions of the human mind are just as primitive.

Why are we so reluctant to admit how little is known about how the mind works? It must come partly from our normal tendency to repress problems that seem discouraging. But there are deeper reasons, too, for wanting to believe in the uniqueness and inexplicability of Self. Perhaps we fear that too much questioning might tear the veils that clothe our mental lives.

To me there is a special irony when people say machines cannot have minds, because I feel we're only now beginning to see how minds possibly could work -- using insights that came directly from attempts to see what complicated machines can do. Of course we're nowhere near a clear and complete theory - yet. But in retrospect, it now seems strange that anyone could ever hope to understand such things before they knew much more about machines. Except, of course, if they believed that minds are not complex at all.

Now, you might ask, if the ordinary concept of Self is so wrong, what would I recommend in its place? To begin with, for social purposes, I don't recommend changing anything - it's too risky. But for the technical enterprise of making intelligent machines, we need better theories of how to "represent", inside computers, the kinds of webs of knowledge and knowhow that figure in everyone's common-sense knowledge systems. We must develop programs that know, say, what numbers mean, instead of just being able to add and subtract them. We must experiment with all sorts of common sense knowledge, and knowledge about that as well.

Such is the focus of some present-day Al research. True, most of the world of "Computer Science" is involved with building large, useful, but shallow practical systems, a few courageous students are trying to make computers use other kinds of thinking, representing different kinds of knowledge, sometimes, in several different ways, so that their programs won't get stuck at fixed ideas. Most important of all, perhaps, is making such machines learn from their own experience. Once we know more about such things, we can start to study ways to weave these different schemes together. Finally, we'll get machines that think about themselves and make up theories, good or bad, of how they, themselves might work. Perhaps, when our machines get to that stage, we'll find it very easy to tell it has happened. For, at that point, they'll probably object to being called machines. To accept that will be will be difficult, but only by this sacrifice will machines free us from our false mottos.

========== KNOWLEDGE AND COMMON SENSE ==========

We've all enjoyed those jokes about the stupid and literal behavior of computers. They send us silly checks and bills for $0.00. They can't tell when we mean "hyphen" from when we mean minus They don't mind being caught in endless loops, doing the same thing over again a billion times. This total lack of common sense is one more reason people think that no machine could have a mind. It's not just that they do only what they're told, it's also that they're so dumb it's almost impossible to tell them how to do things right.

Isn't it odd, when you think about it, how even the earliest Al programs excelled at "advanced" subjects, yet had no common sense? A 1961 program written by James Slagle could solve calculus problems at the level of college students; it even got an A on an MIT exam. But it wasn't till around 1970 that we managed to construct a robot programs that could see and move well enough to handle ordinary things like children's building blocks and do things like stack them up, take them down, rearrange them, and put them in boxes.

Why could we make programs do those grown-up things before we could make them do those childish things? The answer is a somewhat unexpected paradox: much "expert" adult thinking is basically much simpler than what happens in a child's ordinary play! It can be harder to be a novice than to be an expert! This is because, sometimes, what an expert needs to know and do can be quite simple -- only, it may be very hard to discover, or learn, in the first place. Thus, Galileo had to be smart indeed, to see the need for calculus. He didn't manage to invent it. Yet any good student can learn it today.

The surprising thing, thus, was that when it was finished, Slagle's program needed only about a hundred "facts" to solve its college-level calculus problems. Most of them were simple rules about algebra. But

others were about how to guess which of two problems is likely to be easier; that that kind of knowledge is especially important, because it helps the program make good judgments about what to do next. Without this such programs only thrash about; with it they seem much more purposeful. Why do human students take so long to learn such rules? We do not know.

Today we know much more about making such "expert" programs -- but we still don't know much more about making programs with more "common sense". Consider all the different things that children do, when they play with their blocks. To build a little house one has to mix and match many different kinds of knowledge: about shapes and colors, space and time, support and balance, stress and strain, speed, cost, and keeping track. An expert sometimes can get by with deep but narrow bodies of knowledge - but common sense is, technically, a lot more complicated.

Most ordinary computer programs do just the things they're programmed for. Some AI programs are more flexible; when anything goes wrong, they can back up to some previous decision and try something else. But even that is much too crude a base for much intelligence. To make them really smart, we'll have to make them more reflective. A person tries, when things go wrong, to understand what's going wrong, instead of just attempting something else. We look for causal explanations, or excuses, and, when we find them, add them to our networks of belief and understanding. We do intelligent learning. Some day programs, too, could do such things -- but first we'd need a lot more research to find out how.

========== UNCONSCIOUS FEARS AND PHOBIAS. ==========

I'll bet that when we try to make machines more sensible, we'll find that learning what is wrong turns out to be as important as learning what's correct. In order to succeed, it helps to know the likely ways to fail. Freud talked about censors in our minds, that keep us from forbidden acts or thoughts. And, though those censors were proposed to regulate our social activity, I think we use such censors, too, for ordinary problem solving -- to know what not to do. Perhaps we learn a new one each time anything goes wrong, by constructing a process to recognize similar circumstances, in some "subconscious memory".

This idea is not popular in contemporary psychology, perhaps because censors only suppress behavior, so their activity is invisible on the surface. When a person makes a good decision, we tend to ask what "line of thought" lies behind it. But we don't so often ask what thousand prohibitions might have warded off a thousand bad alternatives. If censors work inside our minds, to keep us from mistakes and absurdities, why can't we feel that happening? Because, I suppose, so many thousands of them work at once that, if you had to think about them, you'd never get much done. They have to ward off bad ideas before you "get" those bad ideas.

Perhaps this is one reason why so much of human thought is "unconscious". Each idea that we have time to contemplate must be a product of many events that happen deeper and earlier in the mind. Each conscious thought must be the end of processes in which it must compete with other proto-thoughts, perhaps by pleading little briefs in little courts. But all that we do sense of that are just the final sentences.

And how, indeed, could it be otherwise? There's no way any part of the mind could know everything that happens in the rest. Our conscious minds must be like high executives, who can't be burdened with the small details. There's only time for summaries from other, smaller parts of mind, that know much more about much less; the ones that do the real work.

========== SELF-CONSCIOUS COMPUTERS. ==========

Then, is it possible to program a computer to be self-conscious? People usually expect the answer to be "no". What if we answered that machines are capable, in principle, of even more and better consciousness than people have?

I think this could be done by providing machines with ways to examine their own mechanisms while they are working. In principle, at least, this seem possible; we already have some simple AI programs that can understand a little about how some simpler programs work. (There is a technical problem about the program being fast enough, to keep up with itself, but that can be solved by keeping records.) The trouble is, we still know far too little, yet, to make programs with enough common sense to understand even how today's simple AI problem-solving programs work. But once we learn to make machines that are smart enough to understand such things, I see no special problem in giving them the "self-insight" they would need to understand, change, and improve themselves.

This might not be so wise to do. But what if it turns out that the only way to make computers much smarter is to make them more self-conscious? For example, it might turn out to be too risky to assign a robot to undertake some important, long-range task, without some "insight" about it's own abilities. If we don't want it to start projects it can't finish, we'd better have it know what it can do. If we want it versatile enough to solve

new kinds of problems, it may need to be able to understand how it already solves easier problems. In other words, it may turn out that any really robust problem solver will to understand itself enough to change itself. Then, if that goes on long enough, why can't those artificial creatures reach for richer mental lives than people have. Our own evolution must have constrained the wiring of our brains in many ways. But here we have more options now, since we can wire machines in any way we wish.

It will be a long time before we learn enough about common sense reasoning to make machines as smart as people are. Today, we already know quite a lot about making useful, specialized, "expert" systems. We still don't know how to make them able to improve themselves in interesting ways. But when we answer such questions, then we'll have to face one, even stranger, one. When we learn how, then should we build machines that might be somehow "better" than ourselves? We're lucky that we have to leave that choice to future generations. I'm sure they won't want to build the things that well unless they find good reasons to.

Just as Evolution changed man's view of Life, Al will change mind's view of Mind. As we find more ways to make machines behave more sensibly, we'll also learn more about our mental processes. In its course, we will find new ways to think about "thinking" and about "feeling". Our view of them will change from opaque mysteries to complex yet still comprehensible webs of ways to represent and use ideas. Then those ideas, in turn, will lead to new machines, and those, in turn, will give us new ideas. No one can tell where that will lead and only one thing's sure right now: there's something wrong with any claim to know, today, of any basic differences between the minds of men and those of possible machines.