

When Will Machines Learn?

DOUGLAS B. LENAT

Principal Scientist and Director of AI, MCC, 3500 West Balcones Center Drive, Austin, Texas 78759

Why don't our learning programs just keep on going and become generally intelligent? The source of the problem is that most of our learning occurs at the fringe of what we already know. The more you know, the more (and faster) you can learn.

Unfortunately, fringe (analogical) reasoning is frequently employed purely as a dramatic device. For example, a news reporter talks about a child's valiant battle against disease; or a government issues a clinical-sounding report of a military containment and sterilization operation. This use obscures the fact that analogical reasoning is a critical component of human intelligence; it can help discover new concepts (e.g., is there a military analogue of vaccination? is there a medical analogue of propaganda?) and help flesh them out, as well as helping us to cope with novel situations.

The inverse of "the more you know. . ." is the real culprit: not knowing much implies slow learning. Even the largest machine learning programs (e.g., Eurisko) know only a tiny, tiny fraction of what even a six-year-old child knows (10^{*4} things versus 10^{*9} things). So Learning is fueled by Knowledge, and human-scale learning demands a human-scale amount of knowledge. I see two ways to get it:

1. The 100% Natural Approach: Figure out all the instincts, skills, needs, drives, and predispositions to learning that Nature (Evolution, God, . . .) has hard-wired into human brains and spinal cords and sense organs, and figure out how neonates' raw perception refines into usable knowledge. Then build such a system incorporating all of those things, plus, of course, the right sort of "body" and allow it to "live" for years in the real world: nurture it, let it play, let it bump into walls, teach it to talk, let it go to kindergarten, etc.
2. The Prime the Pump Approach: Codify, in one immense knowledge base, the tens of millions of facts, algorithms, heuristics, stories, representations, etc., that "everybody knows"—the things that the writer of a newspaper article can safely assume that the reader already knows (consensus reality knowledge).

Once the large consensus reality knowledge base exists, either via methodology (1) or (2), then the everyday sort of fringe learning takes over, and the system should be educable in the usual ways: by giving it carefully graded series of readings to do, asking it thought-provoking questions, and helping it over novel or difficult parts by posing a good metaphor drawn from its existing knowledge base.

There are many researchers who are working on limited forms of approach (1)—e.g., the CMU World Modeling Project—and approach (2)—e.g., the Stanford KSL Engineering Design Project.

The CYC project, which Mary Shepherd and I have been working on at MCC since late 1984, is aiming at the fully scaled-up approach (2). We knew when we started that we would have to overcome many representation thorns (e.g., how to deal with time, space, belief, awareness, causality, emotion, stuffs, etc.) and methodological thorns (e.g., how to have tons of knowledge enterers simultaneously editing the same KB, and how to keep their semantics from diverging).

Overcoming those thorns meant finding an adequate way to handle the 99% of the common cases that crop up in everyday life. For example, CYC only represents pieces of time that can be expressed using a simple grammar; those pieces of time are interrelated using a set of 50 relations (such as ends-during) derived by R.V. Guha. We have developed two dozen specialized inference methods (such as inheritance, automatic classification, slot-value subsumption, Horn clause rules) rather than having a system that relies on one general inference procedure. CYC can't easily represent or reason about "the Cantor set of moments from three to four p.m."—but then again, neither can most people! Time and again, that pragmatic focus (not always scruffy, by the way) has pulled us through. Lenat and Guha [1988] describes the CYC project in great detail and explains our solutions to each thorn.

Since 1984, we've been building and organizing and reorganizing our growing consensus reality KB in CYC. We now have about half a million entries in it, and we expect it to increase by one order of magnitude by mid-1990 and one more by the end of 1994. We expect that at roughly that point, a kind of crossover will occur, and it will be cost-effective to enlarge the system from that point onward by having it learn mostly on its own and from online texts.

Naturally, we must build up the CYC KB from some sort of primitives. We have assumed that it must be built from deeply understood knowledge rather than from complex "impenetrable" predicates (or slots or whatever). That is, you can't have *LaysEggsInWater* unless you also have eggs, water, and so on. At first, doing this just made life difficult; having a deep but small KB didn't pay off. Yet, fortunately, when we began to build CYC ever larger and larger, we found that the set of primitives began to converge. That is, it requires less and less work to enter each new fact. This phenomenon is not surprising (it was, e.g., predicted in Pat Hayes' [1985] *Naive Physics Manifesto*). Still, it was quite comforting to see it really happen!

What would divergence look like? One knowledge enterer might use another's already-entered terms to mean slightly different things. Or they might re-enter some knowledge that was already entered in the system under a different name. Thanks to an array of explicit and implicit methods for stating and enforcing semantics, the KB appears to be converging, not diverging.

One key to preventing divergence, much in the spirit of Jim Bennett's ROGET system, is having CYC itself actively help with its own continuing enlargement. For example, CYC brings its full KB to bear to help make guesses during the frame-copy&edit process, to help detect subtle conflicts and errors in the KB, to find analogies that turn out to be different individuals encoding the same knowledge in multiple ways, and to notice gaps and asymmetries in its KB.

So, in summary, you could say that we've chosen an engineering approach to getting a large initial KB. The various representation and methodology thorns have been faced up to and trimmed back; they have neither been avoided nor fixated upon.

Manual KB-building activity is not considered part of machine learning, so it may appear that I've disowned the ML field. Not so! The absence of a CYC-like consensus reality KB is the major bottleneck to automated knowledge acquisition. (Incidentally, I also believe that the same absence is holding back progress in other areas, such as the semantics part of natural language understanding or getting expert systems to be less brittle and to cooperate with each other.)

I was invited to write this paper because my 1975–1984 work on AM and Eurisko helped to spark the rebirth of the machine learning field. I fully expect that CYC will spark a vastly greater renaissance in that field, and that I will be rehabilitated (considered a learning researcher again) during the latter half of the 1990s.

As a parting shot, let me remark that since CYC started in September, 1984, and the crossover to automated KA was and still is scheduled to occur ten years later than that, a tongue-in-cheek answer to this editorial's apparently rhetorical title (When Will Machines Learn?) might be: "September 1, 1994."

References

- Hayes, P. 1985. Naive physics manifesto. In Hobbs and Moore (Eds.), *Formal theories of the common sense world*. Norwood: Ablex.
- Lenat, D., and Guha, R.V. 1988. The world according to CYC. (MCC Technical Report Number ACA-AI-300-88.) An expanded version of this, in book form, is (in press) *Building large knowledge-based systems: Representation and inference in the CYC project*. Addison-Wesley.