The Knowledge Principle: A Personal View

Paul Cohen

Delivered on the occasion of the Festschrift to honor Edward A. Feigenbaum

March 25, 2006

I have been thrilled and frustrated and puzzled by the knowledge principle and its proponents, notably Ed Feigenbaum, ever since graduate school. The thrill is easy to explain: Knowledge is gold, knowledge engineers manage the goldmines, and we, the knowledge researchers, figure out better ways to extract, store, transport and use the precious stuff. We are the boffins in the back room who made it all work, but unlike our poor, dreary cousins in metallurgy and other areas of engineering, we build machines to process ideas, concepts, and other mental stuff. We make thinking machines powered by knowledge. This is the vision I heard from Ed, as well as Bruce and others in the Heuristic Programming Project in the late 70s and early 80s.

Thrilling stuff, and not entirely science fiction, either. It really worked. Ed wrote a book about it with Pamela McCorduck and Penny Nii called the Rise of the Expert Company. Later today you will hear more about the commercial end of the knowledge revolution. And knowledge systems are still relevant and challenging to researchers, which is my subject, today. So we can thank Ed and his colleagues for this thrilling vision, from which so much has followed.

But it has been more than thirty years since MYCIN and longer since DENDRAL, and I don't feel like we have collectively changed the world as much as we expected to. The only reason to look backward is to help us go forward, so this is what I will do in the next few minutes. I got my PhD with Ed and Doug Lenat in 1983 and joined the University of Massachusetts as the "expert systems guy," but it was clear, even then, that no expert systems guy would get tenure. It took too damned long to build an expert system – we spent a couple of years working on one for the workup of chest pain – and at the end, what have you got, scientifically speaking? A handful of minor papers and a big, expensive, useless expert system. Big effort, small result, bad combination for tenure. I think many academic researchers had the same experience. Fortunately, I survived!

So I tried other ways to get more for less in the knowledge arena: Companies were starting to build tools and packages. Cool! One tool could contribute to many systems, I could get famous and maybe even rich, and no troublesome experts to deal with!

I had gone over to the Dark Side, to the world in which we never actually handled knowledge. In the years since, I have come to recognize a kind of free-lunchism in AI. We want self-organizing systems, learning systems, systems that extract knowledge from the web, and knowledge discovery from databases. But we don't actually want to shovel the stuff ourselves. I think of the paper Doug Lenat and Ed wrote in 1987, called On the Thresholds of Knowledge, as a kind of Henry V's solliloquy: We happy few…and all that. It said, We need lots and lots or knowledge, so let's roll up our sleeves and start shoveling. As an incentive, Ed and Doug promised that the ultra-knowledgable system would "go critical" and start to read and learn for itself. DARPA invested, and continues to invest, but I think everyone, even Doug Lenat, who has done more than anyone else along these lines and who is not shy of large numbers, miscalculated the scale of common sense.

For common sense is truly what is required. In the old days, common sense was discussed only in the context of the brittleness problem – the disturbing tendency of expert systems to say stupid things at the boundaries of what they know. But if you want to read a book, or drive to the store, or babysit a child, or read Wikipedia to become more knowledgable, then you need common sense.

Meanwhile, the Web sprang up and we quickly saw that it, too, could support all sorts of mental work, and it made child's play of storing and transporting knowledge, and people were only too happy to shovel the stuff in. The only catch was that all this knowledge could not easily be used by machines. So, on the occasion of the symposium to celebrate his retirement, Ed said something painful, somber and memorable.

He said that we no longer build machines to solve problems, we build machines that dump us at approximately the right place in information space and leave us to fend for ourselves.

Information retrieval, classifier systems, recommender systems, and much else that fit Ed's description were driven by statistical algorithms, not by manually-extracted expert knowledge. These algorithms capitalize on gargantuan amounts of information, much of it provided by the Web. The big question in my mind, today, is whether statistical methods will produce knowledge that can be used by machines to solve problems. Will the methods I earlier disparaged as free-lunchism actually save the day. Can information retrieval and statistical natural language and machine learning and data mining extract the kind of stuff machines will need to not merely dump us at approximately the right place in information space, but to do mental work and solve problems we care about.

I hope the answer is yes because I know only four ways to get knowledge into machines, and three of them aren't looking very good. There's the old-fashioned way, entering it by hand. There's the volunteer knowledge movement, and there's something I call developmental AI. And then there's statistical knowledge generation. The old-fashioned way is too hard. Doug Lenat and others have tried it for years, and I think Doug's last estimate was that he has got between ten percent and one-tenth of one percent of common sense knowledge. The volunteer knowledge movement is pretty young and hasn't made much progress on getting machine-understandable knowledge out of human volunteers. However, the success of Wikipedia suggests people are willing, and some results from OpenMind and similar projects suggest people might be able to tell machines what they know in a form that machines can use. I'll say something about the third method, developmental AI, in a moment.

The statistical folks are confident enough for Tom Mitchell to have invited bets, in the AI Magazine, no less, that a machine will read the web and produce useful knowledge in a formal representation within fifteen years. Statistical natural language is improving, and DARPA is putting some money into learning by reading. The signs are good.

If the knowledge revolution is to be grafted onto statistical algorithms, like old French vines onto new American root stock, then we had better try to ensure that the new statistical roots provide what the old knowledge vines need to thrive. A lot of statistical methods produce what Ed and others call shallow knowledge, typically mappings from one state, broadly construed, to another. Reinforcement learning can produce a policy to control a mobile robot without any conceptualization of space, motion, objects, and other common sense things. Machine translation systems can transform a sentence in English into one in Hindi without any comprehension of either. The irony is that reinforcement learning policies and sentence transformation rules are expert systems in all but name. Expert systems also map states into other states without any comprehension. MYCIN didn't know what febrile means, didn't know what the meninges are for, didn't know what a person is, or whether blood is part of a person. It produced damned good therapy recommendations, but I think equivalent performance, and comparably shallow knowledge, could probably be achieved today with statistical algorithms and sufficient amounts of training data.

I am hoping for more from the statistical revolution, and more from the end users of statistical knowledge-generating algorithms. As users we typically ask for classification, prediction, and superficial mappings from states to states. We rarely demand explanations. If we worked on applications that required machines to *explain* – not *how* they reached a conclusion but *why* the conclusion makes sense, then our machines would need a degree of understanding of the things they reason about. The most desirable kinds of explanations advert to causal relations, so, when possible, our statistical methods should extract causal models – of diseases and linguistic pragmatics, of terrorist tactics and purchasing preferences.

Ed seems to have reached a similar conclusion. I say this because Ed is very interested in storytelling, the business of constructing a plausible and presumably causal accounts of bunches of facts. I think Ed sees storytelling as an application that will require knowledge systems to include deep, causal knowledge.

Extracting causal models from databases and other sources is not beyond today's statistical technology, although it is hard to do, especially when there are hidden variables. I expect tomorrow's statistical knowledge-generating algorithms to deal with these problems as human scientists do, by active experimentation.

I want you to imagine this idea at work in a mobile robot. One fine day, as it wanders around its playpen, it notices that when its wheels turn forward, the image on its artificial retina gets bigger, and when its wheels turn backward, the image becomes smaller. Over time, it refines this association and even manages to use it to accomplish some tasks. And then, it makes an astonishing leap. It realizes that there is a reason the image changes size, and it is the same reason whether its wheels turn forwards or backwards. It generates a theoretical entity, distance, and a causal theory around it, and it tests whether anything other than changes in distance can affect the size of the image. In short, by concentrating not on associations but on causal explanations, our robot has learned some common sense. This is the developmental AI approach I mentioned earlier. I want to build an intelligence that develops common sense by interacting with the world, much as a developing infant does. Ed and I talk about this project quite a bit, and he is always very clear about it: Machines can do useful work with the upper slice of expert knowledge, they don't need the common sense below. Divide and conquer is a good research strategy, partial intelligences are inevitable for some time to come. That said, the threshold of knowledge at which machines go critical and read the Web undoubtedly includes common sense, and one thing I love about Ed is that he is always open to new ways to achieve his vision.