MIT MANAGEMENT EXECUTIVE EDUCATION

MIT CSAIL

MIT SLOAN SCHOOL OF MANAGEMENT

MIT COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE LABORATORY (CSAIL)

ARTIFICIAL INTELLIGENCE:

IMPLICATIONS FOR BUSINESS STRATEGY

ONLINE SHORT COURSE

MODULE 1 UNIT 2
Casebook Video 1 Transcript

DELIVERED IN COLLABORATION WITH getsmarter

MIT AI M1 U2 Casebook Video 1 Transcript

TOM MALONE: The term "artificial intelligence" is notoriously hard to define. Sometimes, for example, people use it to mean things that are hard for computers to do, like understanding English as opposed to things that we already know how to do with computers, like accounting.

More recently, with all the hype about AI, companies sometimes try to make their newest products sound cool by using the term "AI" to mean just about anything computers can do, including traditional databases or statistical techniques.

In this course, we'll start with a very simple definition of AI, which is: Machines acting in ways that seem intelligent. This simple definition is easy to remember, and I think accurately reflects how many people use the term. But, of course, the real story is more complicated than this simple definition. So, now, to give you a much fuller definition and a history of AI, here's Professor Patrick Winston.

PATRICK WINSTON: Well, these are exciting times. Computation isn't just changing some things; it's changing everything. So, when times are exciting, I think, it's a good idea to step back and think for a while about where we are and how we got here – where we are going and where we might go. What can happen next? Modern thinking about the possibility of intelligent systems all started with Turing's famous paper in 1950.

He, of course, knew that he couldn't define what intelligence was. So, because of that, he introduced what he called the Turing test. The idea was that if a human couldn't tell within five minutes if he was talking to a computer or a person, then the computer would be said to have passed the Turing test. Turing couldn't imagine the possibility of dealing with speech back in 1950. So, he was dealing with a teletype – much like what you would think of as texting today. And it was because Turing knew that he couldn't actually define what intelligence was – it's too hard, too slippery.

So, that's why he introduced the Turing test. But I've read that paper many times, and I think that what Turing was really after was not trying to define intelligence or to test for intelligence, but really to deal with all the objections that people had about why it wasn't going to be possible. So, in his paper Turing demolished all kinds of arguments. He demolished arguments about theological positions and mathematical conclusions, all sorts of things including one of my favorites, which was what goes back to 1840s when Babbage was trying to build the first computer. His programmer was Ada Lovelace, the Countess Lovelace. And she said, "Don't worry about a thing. They can only do what we program them to do." Of course, she could have said, "They can only do what we program them to do, and what we tell them how to do, and what they learn how to do on their own." But that might not have had quite the soothing effect she was looking for.

Any event, Turing needed to have some way of dealing with intelligence's definition because he was writing for mathematicians in a philosophical journal, but what he was really about was all of those objections. If he had been a student of Marvin Minsky, he probably wouldn't have bothered because Marvin Minsky introduced the term "suitcase word." He meant that a lot of words are like big suitcases you can stuff anything into; they're so broad they cover too many things for a precise definition.

So, that's what Minsky would have said about intelligence. He would say, "Too broad; it's a suitcase term." And if Turing had that idea he probably wouldn't have bothered with the Turing test. What Turing really told us was that serious people can think seriously about computers thinking and that there's no reason to doubt that computers will think someday. And, in a suitcase sense, that day is approaching. About ten years after Turing published his paper in 1950, important laboratories were set up by Marvin Minsky, John McCarthy, Allen Newell, and Herb Simon.

McCarthy's approach at Stanford was to start with mathematical logic. He spent his whole life trying to bend logic to his will. Newell and Simon focused on modelling human thinking. They developed systems that solved simple puzzles, work out simple problems in a manner that they believed was consistent with human experiments. Minsky's approach was harder to characterize. He believed that one representation met with an approach, no one of those could deliver a full understanding of intelligence. That was the central message of his seminal paper which was titled, *Steps Towards Artificial Intelligence* in 1961. You know, in retrospect, we can think that Turing told us we could do this. And that paper by Minsky told us what to do. So, that's why Turing and Minsky are often regarded as the real pioneers, the real founders of the field of artificial intelligence.

Well, in any event, that brings us to what some people call AI's first wave. In the early 1960s, James Slagle wrote a program that integrated symbolic expressions. He was trying to model what a freshman does at MIT when they learn that kind of mathematics. His program viewed as a freshman deserved an A+. It was a terrific program that did terrific kinds of work and broke a lot of new ground. Because Slagle's program performed so impressively, it's what I consider to be the signature program of AI's first wave. That key idea was called problem reduction. The idea is simple. You just take a hard problem and you break it into simpler problems. And then you break those simpler problems into problems that are still simpler until you've got something you can just, you can just do.

That's what problem reduction was about, but it's only one of a cornucopia of ideas that have emerged from AI research. At MIT, the work of Slagle was quickly followed by other successors. And by 1970, programs understood drawings, they learned from examples, they knew how to build structures. And one even answered questions much like Siri and Alexa do today. From all these programs we learn one really important lesson. And that lesson is that if you get the representation right, then you're almost done.

That lesson about representation is at the heart of how I define what AI is all about. I start by saying that AI is about models of thinking, and perception, and action. But, of course, that invites the question: What's a model? A model is something that we use and value because it behaves in some ways like the real thing. We can use models to understand, to explain, to predict and control. That's what all of science is really about.

And, in particular, why are representations important? Well, they're important because they support all of those models of thinking, perception, and action. But maybe I had better say another word or two about what a representation is. In general terms, it's a set of conventions for describing situations. And you already know about various kinds of conventions for describing situations. Back when you were in high school and did word problems, the algebra that came out of them was a representation for all those words in the problem. And representation might be some other kind of math. It could be a way of writing down rules. Another representation might be an enumeration of important features. And, for some problems, a good one might be a map or a picture.

So, now I can add a bit to my definition of AI. AI is about representations that support models of thinking, perception, and action. But I can even go a step further and say that the real value of a representation is it exposes constraint. And that constraint makes methods possible. So, now it's getting pretty complicated, but there's one more step. You have to organize all the methods in your system into some sort of overall architecture that deploys them. So, now we have something pretty complicated.

AI is about the architectures that deploy methods enabled by constraints exposed by representations that support models of thinking, perception, and action. And, of course, it's not just about doing, it's also about learning to do. So, whoa, that's a pretty big mouthful, but each part of that definition is important. The technology behind the second wave came about fifteen years after the first in the mid-1970s. Ed Shortliffe, who was a student at Stanford, developed the MYCIN system for diagnosing a class of diseases.

Like Slagle's integration program, it performed at a human level as good as, say, a primary care physician. The key idea was that at least some kinds of expertise can be captured in a collection of rules. That's one is a mouthful of medical jargon: If x's type is primary bacteremia and x's suspected portal of entry is gastrointestinal and a site of the culture of x is sterile, then there is evidence that x is bacterioides. Woah, that is medical jargony, but the point is that a whole collection of rules like that can do a pretty good job of diagnosing infectious diseases of the blood.

From my perspective, there's a second wave that's centered on rules like these as a way of representing knowledge. And programs based on MYCIN, the MYCIN precedent, were called rule-based expert systems. They were a dominant force for the next decade. And the rule idea remains an important part of the AI toolkit.

TOM MALONE: Did you understand all the concepts covered in this video? If you'd like to go over any of the sections again, please click on the relevant button.