Artificial General Intelligence

Rod Smith
associative-ai.com
5 May 2019

Abstract

Artificial intelligence (AI) research now strongly focuses on special purpose "narrow AI". Artificial general intelligence (AGI) research has stalled. One suggested reason is that current concepts used to understand general intelligence are inadequate and new ones are needed. But AGI has two subjects: (1) general intelligence, (2) the digital computer. AI including AGI accepts Turing's concept of the electronic device as a practical version of his abstract Turing machine design. I show why this account is inadequate for understanding the intrinsic machine, as AGI must, and conclude that to progress, AGI must abandon Turing. Further, I show that if Searle's Chinese room argument is correct, this means only that Turing's conception of the computer does not allow semantic content. It doesn't mean computers will never think. There are other conceptions. The CRA fails to address whether any of these are both true to the hardware and allow semantic content, and one of them may.

Body

What's the problem with AI? Those who say "Nothing" are usually talking about narrow AI – the research and development program of applying artificial intelligence (AI) theory to narrow domains such as identifying faces, economic trends, spending habits, and objects around moving vehicles. They're not talking about human-like general intelligence, AGI. Achieving the goal of AGI is now usually seen as distant: achieving an electronic digital computer that can learn through sensory perception, generalize in the same sort of way humans can, engage in idiomatic conversation, and survive in a changing and hostile world.

AGI is now regarded as a relatively minor sub-field within the overall research project of Artificial Intelligence. The number of recent AGI conferences compared to narrow-AI conferences clearly attests to that. But AI's original goal was AGI. Early signs of progress towards AGI were extremely promising. These early advances underpinned further successes but only in limited domains. After 60 years of research at the best Western universities and elsewhere, AGI is now regarded, at least for the foreseeable future, as intractable. Why has the AGI project failed to meet expectations?

I propose a fundamental reason. There's a basic problem. It's not a defect of machinery. Rather, it's a failure to understand the rudimentary nature of the machine – the one called the stored-program electronic digital computer.

As far as alleged conceptual failures go, this is significant because the computer is AI's only viable research and development platform with enough uniquely addressable memory and internal speed of state change. Understanding its true nature is essential. But the concepts used to understand it are defective. It's essential character has been misjudged. It's basic abilities have been misinterpreted, and core attributes have been overlooked.

This claim, of course, is *prima facie* absurd. The established wisdom of the esteemed field of computer science explains the nature of the machine. This scientific account isn't disputed, either within AI or by AI critics. John Searle, for instance, author of the rightly famous Chinese room argument strongly attacks AI but blindly accepts the computer science concept of the computer as a practical version of Turing's abstract Turing machine design.

I argue that the computer science construct is not only wrong, it's so wrong that it prevents AGI from seeing the true nature of the machine. The computer science concept might be ideal for understanding human use as a tool, and in fact derives from this use, but for understanding the intrinsic machine, as AGI must, it's woefully and fundamentally defective.

The basic mistake is to call a computer a computer. By that I mean the key failure is to understand the internal workings of device with the concept of computation. This concept is helpful when considering human uses but conceals the machine's essential nature. For AGI, a leading-edge and extremely important research project (success has the potential to end most or all poverty), it's difficult to imagine a worse type of mistake.

AI's idea of machine computation originated with British mathematician Alan Turing. In 1936, before the advent of the electronic digital machine, he explains, "*Computing is normally done by [a human] writing certain symbols on paper...*" (in his paper, "On Computable Numbers..."). That is, a human "manipulates" external symbols, interpretable shapes, inscribed on paper. But, Turing continues, "*We may now construct a machine to do the work of this [human] computer*". This machine is now known as the Turing machine. In detail:

> "*The machine is supplied with a "tape", (the analogue of paper) running through it, and divided into sections (called "squares") each capable of bearing a "symbol". ... In some of the configurations ... the machine writes down a new symbol on the scanned square*".

In 1950, after the advent of the electronic digital computer, Turing explains, "*..digital computing machines ... are in fact practical versions of the universal [Turing] machine*". (The term "universal machine", here, refers to a certain type of use of the 1936 Turing machine design.) Thus, computers internally manipulate symbols. For instance: "*The computer includes a store corresponding to the paper used by a human computer. It must be possible to write into the store any one of the combinations of symbols which might have been written on the paper*".

Hence, Turing explains the electronic internals of the new device with the concept of a human manipulating symbols according to rules about their shapes.

OK. But there's a glaring difference between what the human does and what Turing uses the respective concept to explain. The human computer manipulates external symbols according to rules about their shapes. The shapes are on pieces of paper on a desk, not inside the worker's head. The Turing machine, by contrast, manipulates *internal* symbols. And since the electronic digital computer is a practical version of the Turing machine, the electronic computer manipulates internal symbols, too, on Turing's account of the machine.

So the idea of a human computing with external shapes on paper is used to explain electronic workings inside the new device. In both cases, what is manipulated is regarded as the same type of thing – tokens whose shapes have interpretations, or meanings.

As Searle says: "*Alan Turing gave half a century ago ... the definition of the computer.*" (*The Mystery of Consciousness*, p. 59), "*A computer is by definition a device that [internally] manipulates formal symbols*" (*The Mystery of Consciousness*, p. 9), "*...all that 'formal' means here is that I can identify the symbols entirely by their shapes*" ("Minds, Brains, and Programs"), "*The computer operates by manipulating [inner] symbols*" (Mind, A Brief Introduction, p. 63), and on a related matter, "*symbols, by definition, have no meaning (or interpretation, or semantics)... except insofar as someone outside the system gives it to them*" ("Artificial Intelligence and the Chinese Room: An Exchange").

This, Turing's conception of machine computation as symbol manipulation, is the principal source of today's ubiquitous belief that computers are symbol manipulating devices. A concept faithfully pictured in Searle's famous Chinese room thought experiment analogy of the digital computer in which a man (the CPU) receives, finds, moves, compares, ejects ("manipulates") Chinese ideograms according to rules in a book about their shapes.

It's helpful to suppose that word processors actually process words. That when a document is "typed into the computer" as it's said, that the words in the document actually go into the machine. That when keyboard keys are pressed, the shapes on them leap off and race down the wires and into the device and are stored, and from there might be displayed, manipulated, printed by a printer. This great conceptual convenience is why word processing computers and word processing software is said to process words.

But such processing never occurs. That's because symbols never enter the machine. Similarly, the shapes on keys of a manual typewriter don't jump off, race down the metal connecting rods and into the hammers that impress the inked shapes onto paper. But it's convenient to suppose, in a sense, that they do. Just as it's helpful to think that computers store images and sounds. That when we download "Bat out of Hell" from the Internet, the song is actually stored in the machine. But this is a fiction. Images, symbols, words, sounds, none of these things are ever received, stored or manipulated inside a digital computer.

My claim is that this fiction, while extremely helpful for understanding human use of the device, has hugely negative implications for AGI which needs to consider the machine in and of itself.

AGI seeks to create a mind in a digital computer. For a start, if Turing's concept of machine computation is applied to this goal, then the machine mind will operate by manipulating inner symbols according to rules about their shapes. One problem with this idea is that, by its electronic design, the digital computer is completely incapable of internally reacting to the shapes of anything, even if symbols did actually enter the machine, which they don't.

To expand on this, to suppose that symbols enter the machine, as AI, Searle and many others do suppose, (a) is just plain false, (b) creates insurmountable semantic problems, and (c) blinds AGI to the true nature of the device.

I think, on reflection, it's pretty obvious that the symbol-processing concept of the computer derives from its human use. Certainly, no symbols are actually inside the device (except perhaps printed on capacitors or the main-board). But plenty are to be found on peripheral attachments. They can be seen liberally adhering to exposed surfaces of peripheral devices. They are prominent on keyboard keys, as pixel patterns in screens, and printers spray them or melt them into sheets of external paper.

It's no accident that the only pertinent symbols in today's computer systems are encrusted on exposed surfaces of attachments. That's where people can see them. They are there for humans to see. None are inside the machine. There is no internal symbol manipulation.

We used to happily say, I've just faxed the document to uncle Harold. But the document didn't zoom down the wire. In fact such a document is not sent. It's image is transduced into analogue voltages, and the voltages transmitted to uncle Harold's fax machine, which machine then transduces the voltages back into an image on paper. The idea that the document was sent is a convenient fiction. In exactly the same way, and for exactly the same reason, so is the idea that computers process symbols. In reality, the only place relevant interpretable shapes exist in a computer system is encrusted on exposed surfaces of attachments – put there for humans to see.

It might be said that computers do in fact manipulate inner symbols, it's just that the external ones change form before they enter the machine, change, that is, into a form conducive to the sort of electronic processing and storage that happens inside the machine.

Well, they don't change form. Dracula changes form – into a bat, Bruce Banner changes form into The Incredible Hulk, Odo changes form into whatever, but external symbols don't change form into internal anything. On the input side, symbols exist glued to the top surface of caps of plastic keyboard keys, shapes sometimes over-sprayed with hard clear lacquer so they won't get rubbed off by sweaty fingertips. These shapes don't get transduced. They don't enter the machine or any attachment. What an attachment internally creates and sends into the machine are clocked voltages. These electric objects are sub-microscopic groupings of mobile electrons. They don't have interpretable shapes, if they can be coherently said to have any shape at all. They were not caused by the shapes on the key caps except inasmuch as a human sees the shape and chooses to press the key.

Searle's Chinese room argument is perhaps the most well known philosophical debate of the past 50 years, in Western Philosophy, at least. Yet it's founded on the claim that computers process things that have an external but no internal semantics (whose shapes have meanings to observers, but the tokens that bear the shapes, or the shapes *per se*, do not intrinsically contain or carry the meanings).

Well, in fact the things computers process have no semantics – of any sort. These internal things being clumps of racing electrons. The clumps transmitted from a keyboard to the machine have a causal history that goes back to certain keys on the keyboard, but not to the dried shapes embossed on the top surfaces of the plastic caps. The shapes don't cause the clocked electrons (though a human eye sees the shape and the connected human brain causes a fingertip to press the key inscribed with the noticed shape). Intrinsically, a keyboard is a rectangular slab of rather inflexible digital skin containing a rather coarse array of touch detectors.

The things computers internally process have no meanings of any sort. But that doesn't imply that the machines will never think. It's just that Searle can't see why they might think. He's blinded by the myth, created by Turing, that computers internally process symbols (and by reacting to their shapes).

Searle's been blinded by the myth of human use. So he analyses the machine from the perspective of symbols, syntax, linguistic meaning, the shapes encrusted on exposed surfaces of attachments. He's analyzing the wrong thing. The intrinsic machine doesn't have attachments littered with shapes put there so humans can see them. He ought to apply his powers reason to what actually happens inside the machine, not to external pixel patterns, shapes glued to top surfaces of plastic caps, or powder melted into wood pulp.

The only time data or programs are symbolic is when they are displayed or printed. The program stack isn't a stack of symbols and hence isn't a stack of instructions, except insofar as the false idea that they are gives humans an easy-to-grasp though false picture of what happens inside the machine.

When the human computer writes certain symbols on paper, no one suggests that objects of the same shapes are bobbing about inside the person's skull, that the organic brain is reacting

to the shapes of mobile inner objects of the same or any shapes. But this is what Turing's computational theory of the electronic digital machine is saying, when used to try to understand the machine mind. It's an idea that's wrong from the start. It's an idea founded on myth. Electronic digital computers don't internally process symbols.

The fantasy that they do has led to such ideas as that declarative sentences can be typed into the machine and then constitute an internal knowledge base, a nonsense that led to the CYC and SOAR projects of the 1980s, for instance. And to the laughable concept that data inside the machine comprises knowledge representations. A fiction relentlessly advanced by, for instance, Russell and Norvig in their famous university text, *Artificial Intelligence: A Modern Approach*. This isn't science. It's alchemy.

The anthropomorphic bias that says computers internally process interpretable anything needs to be abandoned when it comes to trying to understanding the machine in its essential self, as AGI must.

The first step, I can suggest, to understanding the intrinsic device is to abandon Turing, computer science, and the computational concept of the machine. I know this seems like abandoning everything. That's because it *is* abandoning everything (except the machine). But that's a good idea – because everything is wrong (except the machine). The false conception that computers compute has unfortunately blinded AGI to the essential characteristics of the only device that might think, and has constrained for over 60 years possibly the most important technical project ever.

As for a different and better conception of the machine that for historical reasons of human use is called a "computer", there are other concepts. One of them seems a lot better.

Searle's Chinese room argument is regarded as concluding that computers will never think. But Searle uncritically accepts the computer science explanation of the machine as a practical version of Turing's Turing machine design. What the CRA actually shows – and why it is so important to AGI – is that Turing's concept and hence AI's concept of the machine does not allow the creation of an internal semantics (knowledge). The CRA does not deliver a verdict on the machine, but on Turing's conception of the machine. There are other conceptions.

Just because one such understanding doesn't allow an inner semantics, it doesn't follow that others are similarly hamstrung. The task of AGI, then, is to find a better concept of its device, one that is both true to the hardware and allows the development of inner semantic structures. But is there such an alternative yet accurate conception? Sure. At least one.