DOI:10.1145/3132724

David Lorge Parnas

Inside Risks

The Real Risks of Artificial Intelligence

Incidents from the early days of AI research are instructive in the current AI environment.

The vast increase in speed, memory capacity, and communications ability allows today's computers to do things that were unthinkable when I started programming six decades ago. Then, computers were primarily used for numerical calculations; today, they process text, images, and sound recordings. Then, it was an accomplishment to write a program that played chess badly but correctly. Today's computers have the power to compete with the best human players.

The incredible capacity of today's computing systems allows some purveyors to describe them as having "artificial intelligence" (AI). They claim that AI is used in washing machines, the "personal assistants" in our mobile devices, self-driving cars, and the giant computers that beat human champions at complex games.

Remarkably, those who use the term "artificial intelligence" have not defined that term. I first heard the term more than 50 years ago and have yet to hear a scientific definition. Even now, some AI experts say that defining AI is a difficult (and important) question—one that they are working on. "Artificial intelligence" remains a buzzword, a word that many think they understand but nobody can define.

Recently, there has been growing alarm about the potential dangers of artificial intelligence. Famous giants of the commercial and scientific world have expressed concern that AI will eventually make people superfluous. Experts have predicted AI will even replace specialized professionals such as lawyers. A Microsoft researcher recently made headlines saying, "As artificial intelligence becomes more powerful, people need to make sure it's not used by authoritarian regimes to centralize power and target certain populations."[a]

Application of AI methods can lead to devices and systems that are untrustworthy and sometimes dangerous.

Automation has radically transformed our society, and will continue to do so, but my concerns about "artificial intelligence" are different. Application of AI methods can lead to devices and systems that are untrustworthy and sometimes dangerous.

a Interview with Kate Crawford in *The Guardian*, March 13, 2017.

An Early Introduction to AI

As a student at Carnegie Mellon University (CMU),[b] I learned about "artificial intelligence" from some of the field's founders. My teachers were clever but took a cavalier, "Try it and fix it," attitude toward programming. I missed the disciplined approach to problem solving that I had learned as a student of physics, electrical engineering, and mathematics. Science and engineering classes stressed careful (measurement-based) definitions; the AI lectures used vague concepts with unmeasurable attributes. My engineering teachers showed me how to use physics and mathematics to thoroughly analyze problems and products; my AI teachers relied almost entirely on intuition.

I distinguished three types of AI research:

- building programs that imitate human behavior in order to understand human thinking;
- building programs that play games well; and
- showing that practical computerized products can use the methods that humans use.

Computerized models can help researchers understand brain function. However, as illustrated by Joseph Weizenbaum,[2] a model may duplicate the "black-box" behavior of some mechanism without describing that mechanism.

b CMU was then known as Carnegie Institute of Technology.

Writing game-playing programs is harmless and builds capabilities. However, I am very concerned by the proposal that practical products should apply human methods. Imitating humans is rarely the best way for a computer to perform a task. Imitating humans may result in programs that are untrustworthy and dangerous.

To explain my reservations about AI, this column discusses incidents from the early days of AI research. Though the stories are old, the lessons they teach us remain relevant today.

Heuristic Programming

AI researchers sometimes describe their approach as "heuristic programming." An early CMU Ph.D. thesis defined a *heuristic program* as one that "does not always get the right answer." Heuristic programs are based on "rules of thumb," that is, rules based on experience but not supported by theory.[c]

"Heuristic" is not a desirable attribute of software. People can use rules of thumb safely because, when rules suggest doing something stupid, most people won't do it. Computers execute their programs unquestioningly; they should be controlled by programs that can be demonstrated to behave correctly in any situation that might arise. The domain of applicability of a program should be clearly documented. Truly trustworthy programs warn their users whenever they are applied outside that domain.

Heuristics can be safely used in a program if:

- The specification allows several acceptable solutions and the heuristic is used either to select one of them or to determine the presentation order.
- The heuristic is intended to speed up a program that conducts a search that will either find a solution or establish that there is no solution.

In other situations, heuristic programming is untrustworthy programming.

What Alan Turing Really Said

Alan Turing is sometimes called the "Father of AI" because of a 1950 paper, "Computing Machinery and Intelligence."[1] It is frequently claimed that, in that paper, Turing proposed a test for machine intelligence.

Those who believe that Turing proposed a test for machine intelligence should read that paper. Turing understood that science requires agreement on how to measure the properties being discussed. Turing rejected "Can machines think?" as an unscientific question because there was no measurement-based definition of "think." That question is not one that a scientist should try to answer.

Turing wrote: "*If the meaning of the words 'machine' and 'think' are to be found by examining how they are commonly used it is difficult to escape the conclusion that the meaning and the answer to the question, "Can machines think?" is to be sought in a statistical survey such as a Gallup poll. But this is absurd. Instead of attempting such a definition I shall replace the question by another, which is closely related to it and is expressed in relatively unambiguous words.*"

Turing's proposed replacement question was defined by an experiment. He described a game (the imitation game) in which a human and a machine would answer questions and observers would attempt to use those answers to identify the machine. If questioners could not reliably identify the machine, that machine passed the test.

Turing never represented his replacement question as equivalent to "Can machines think?" He wrote, "*The original question, 'Can machines think?' I believe to be too meaningless to deserve discussion.*" A meaningless question cannot be equivalent to a scientific one.

Most of Turing's paper was not about either machine intelligence or thinking; it discussed how to test whether or not a machine had some well-specified property. He also speculated about when we might have a machine that would pass his test and demolished many arguments that might be used to assert that no machine could ever pass his test. He did not try to design a machine that would pass his test; there is no indication that he thought that would be useful.

Joseph Weizenbaum's Eliza

Anyone interested in the Turing Test should study the work of the late MIT

c Those who write heuristic programs rarely characterize the set of conditions under which the program would produce an incorrect result. See the section "An AI System for Constructing Parsers."

professor Joseph Weizenbaum.[3] In the mid-1960s, he created Eliza, a program that imitated a practitioner of Rogerian psychotherapy.[d] Eliza had interesting conversations with users. Some "patients" believed they were dealing with a person. Others knew that Eliza was a machine but still wanted to consult it. Nobody who examined Eliza's code would consider the program to be intelligent. It had no information about the topics it discussed and did not deduce anything from facts that it was given. Some believed Weizenbaum was seriously attempting to create intelligence by creating a program that could pass Turing's test. However, in talks and conversations, Weizenbaum emphasized that was never his goal. On the contrary, by creating a program that clearly was not "intelligent" but could pass as human, he showed that Turing's test was not an intelligence test.

Robert Dupchak's Penny-Matcher

Around 1964,[e] the late Robert Dupchak, a CMU graduate student, built a small box that played the game of "penny matching."[f] His box beat us consistently. Consequently, we thought it must be very intelligent.

It was Dupchak who was intelligent—not his machine. The machine only remembered past moves by its opponent and assumed that patterns would repeat. Like Weizenbaum, Dupchak demonstrated that a computer could appear smart without actually being intelligent. He also demonstrated that anyone who knew what was inside his box would defeat it. In a serious application, it would be dangerous to depend on such software.

Character Recognition

A popular topic in early AI research and courses was the character recognition problem. The goal was to write programs that could identify hand-drawn or printed characters. This task, which most of us perform effortlessly, is difficult for computers. The optical character recognition software that I use to recognize characters on a scanned printed page frequently errs. The fact that character recognition is easy for humans but still difficult for computers is used to try to keep programs from logging on to Internet sites. For example, the website may display[g] a CAPTCHA as shown here

and require the user to type "s m w m." This technique works well because the character recognition problem has not been solved.

Early AI experts taught us to design character recognition programs by interviewing human readers. For example, readers might be asked how they distinguished an "8" from a "B." Consistently, the rules they proposed failed when implemented and tested. People could do the job but could not explain how.

Modern software for character recognition is based on restricting the fonts that will be used and analyzing the properties of the characters in those fonts. Most humans can read a text, in a new font without studying its characteristics, but machines often cannot. The best solution to this problem is to avoid it. For texts created on a computer, both a human-readable image and a machine-readable string are available. Character recognition is not needed.

An AI System for Constructing Parsers

As a new professor, I made appointments with three famous colleagues to ask how to recognize a good topic for my students' Ph.D theses. The late Alan Perlis, the first recipient of ACM's prestigious Turing Award, gave the best answer. Without looking up from his work, he said, "Dave, you'll know one when you see it. I'm busy; get out of here!" Two other Turing Award winners, the late Allen Newell and the late Robert Floyd, met with me. Separately, both said that while they could not answer my question directly, they would discuss both a good thesis and a bad one. Interestingly, Newell's example of a good thesis was Floyd's example of a bad one.

The disputed thesis presented an AI program that would generate parsers[h] from grammars. Newell considered it good because it demonstrated that AI could solve practical problems. Floyd, a pioneer in the field of parsing, explained that nobody could tell him what class of grammars the AI parser generator could handle, and he could prove that that class was smaller than the class of languages that could be handled by previously known mathematical techniques. In short, while the AI system appeared to be useful, it was inferior to systems that did not use heuristic methods. Bob Floyd taught me that an AI program may seem impressive but come out poorly when compared to math-based approaches.

An AI System that "Understood" Drawings and Text

A 1967 AI Ph.D thesis described a program that purportedly "understood" both natural language text and pictures. Using a light pen and a graphics display,[i] a user could draw geometric figures. Using the keyboard, users could ask questions about the drawing. For example, one could ask "Is there a triangle inside a rectangle"? When the author demonstrated it, the program appeared to "understand" both the pictures and the questions. As a member of the examining committee, I read the thesis and asked to try it myself. The system used heuristics that did not always work. I repeatedly input examples that caused the system to fail. In production use, the system would have been completely untrustworthy.

The work had been supervised by another Turing Award recipient, Herbert Simon, whose reaction to my observing that the system did not work was, "The system was not designed for antagonistic users." Experience has shown that computer systems must be prepared for users to be careless and, sometimes, antagonistic. The techniques used in that thesis would not be acceptable in any commercial product. If heuristics are used in criti-

d Practitioners of Rogerian psychotherapy echo the patient's words in their responses.

e Dupchak's accidental death prevented publication of his work. I cannot give a precise date.

f Penny-matching is a two-player game. Each player uses a coin to make a head or tail choice. One player wins if both pick the same face; the other wins if the choices are different.

g This example was found in Wikipedia.

h Parsers, an essential component of compilers, divide a program into its constituent parts. Before Floyd's work, parsers were created by humans. Floyd's algorithm automatically generated parsers for a large class of languages.

i Advanced hardware for the time.

cal applications, legal liability will be a serious problem.

An AI Assembly-Line Assistant

An assembly line could run faster after tool-handling assistants were hired: Whenever workers finished using a tool, they tossed it in a box; when a tool was needed, the assistant retrieved it for the workers.

A top research lab was contracted to replace the human assistants with robots. This proved unexpectedly difficult. The best computer vision algorithms could not find the desired tool in the heap. Eventually, the problem was changed. Instead of tossing the tool into the box, assemblers handed it to the robot, which put it in the box. The robot remembered where the tool was and could retrieve it easily. The AI controlled assistant could not imitate the human but could do more. It is wiser to modify the problem than to accept a heuristic solution.

"Artificial Intelligence" in German[j]

When AI was young, a German psychology researcher visited pioneer AI researchers Seymour Papert and Marvin Minsky (both now deceased) at MIT. He asked how to say "artificial intelligence" in German because he found the literal translation (Künstliche Intelligenz[k]) meaningless.

Neither researcher spoke German. However, they invited him to an AI conference, predicting that he would know the answer after hearing the talks. Afterward, he announced that the translation was "natürliche Dummheit" (natural stupidity) because AI researchers violated basic rules of psychology research. He said that psychology researchers do not generally ask subjects how they solve a problem because the answers might not be accurate; if they do ask, they do not trust the answers. In contrast, AI researchers were asking chess players how they decide on their next move and then writing programs based on the player's answers.

Artificial Neural Networks

Another approach to AI is based on modeling the brain. Brains are a network of units called neurons. Some researchers try to produce AI by imitating the structure of a brain. They create models of neurons and use them to simulate neural networks. Artificial neural networks can perform simple tasks but cannot do anything that cannot be done by conventional computers. Generally, conventional programs are more efficient. Several experiments have shown that conventional mathematical algorithms outperform neural networks. There is intuitive appeal to constructing an artificial brain based on a model of a biological brain, but no reason to believe this is a practical way to solve problems.

The Usefulness of Physics and Mathematics

A researcher presented a paper on using AI for image processing to an audience that included experts in radar signal processing. They observed that the program used special cases of widely used signal-processing algorithms and asked "What is new in your work?" The speaker, unaware of techniques used in signal processing, replied, "My methods are new in AI." AI researchers are often so obsessed with imitating human beings that they ignore practical approaches to a problem.

A study of building temperature-control systems compared an AI approach with one developed by experienced engineers. The AI program monitored individual rooms and turned on the cooling/heating as needed. The engineers used a heat-flow model that included the building's orientation, the amount of sunlight hitting sections of the building, the heat absorption/loss characteristics of the building, and so on. Using this model, which allowed their system to anticipate needs, and the ability to pump heat from one part of the building to another, they designed a system that reduced temperature fluctuations and was more energy efficient.

Learning is not magic, it is the use of data collected during use to improve future performance.

Humans do not have the measurement and calculation ability that is available to a modern computer system; a system that imitates people won't do as well as one based on physical models and modern sensors.

Humans solve complex physics problems all the time. For example, running is complex. Runners maintain balance intuitively but have no idea how they do it. A solution to a control problem should be based on physical laws, and mathematics, not mimicking people. Computers can rapidly search complex spaces completely; people cannot. For example, a human who wants to drive to a previously unvisited location is likely to modify a route to a previously visited nearby place. Today's navigation devices can obtain the latest data and calculate a route from scratch and often find better routes than a human would.

Machine Learning

Another approach to creating artificial intelligence is to construct programs that have minimal initial capability but improve their performance during use. This is called *machine learning*. This approach is not new. Alan Turing speculated about building a program with the capabilities of a child that would be taught as a child is taught.[1] Learning is not magic; it is the use of data collected during use to improve future performance. That requires no "intelligence." Robert Dupchak's simple penny-matching machine used data about an opponent's behavior and appeared to "learn." Use of anthropomorphic terms obscures the actual mechanism.

Building programs that "learn" seems easier than analyzing the actual problem, but the programs may be untrustworthy. Programs that "learn" often exhibit the weaknesses of "hill-climbing"[l] algorithms; they can miss the best

j I cannot warrant the truth of this story; it was related to me as true, but I was not present for the events. I include it because it contains an important lesson.

k Current terminology in German.

l *Hill-climbing algorithms* are analogous to hikers who always walk uphill. They may end up at the top of a foothill far below the mountain peak.

solution. They may also err because of incomplete or biased experience. Learning can be viewed as a restricted form of statistical classification, mathematics that is well developed. Machine-learning algorithms are heuristic and may fail in unusual situations.

Robot Ethics

When people view computers as thinking or sentient beings, ethical issues arise. Ethicists traditionally asked if the use of some device would be ethical; Now, many people discuss our ethical obligations to AIs and whether AIs will treat us ethically. Sometimes ethicists posit situations in which AI must choose between two actions with unpleasant consequences, and ask what the device should do. Because people in the same situation would have the same issues, these dilemmas were discussed long before computers existed. Others discuss whether we are allowed to damage an AI. These questions distract us from the real question, "Is the machine trustworthy enough to be used?"

Wordplay

The AI research community exploits the way that words change meaning: the community's use of the word "robot" is an example. "Robot" began as a Czech word in Karel Čapek's play, *R. U. R.* (Rossum's Universal Robots). Čapek's robots were humanoids, almost indistinguishable from human beings, and acted like humans. If "robot" is used with this meaning, building robots is challenging. However, the word "robot" is now used in connection with vacuum cleaners, bomb-disposal devices, flying drones, and basic factory automation. Many claim to be building robots even though devices remotely like Karel Čapek's are nowhere in sight. This wordplay adds an aura of wizardry and distracts us from examining the actual mechanism to see if it is trustworthy. Today's "robots" are machines that can, and should, be evaluated as such. When discussing AI, it is important to demand precise definitions.

AI: Creating Illusions

Alan Perlis referred to AI researchers as "illusionists" because they try to create the illusion of intelligence. He argued they should be considered stage magicians rather than scientists. Dupchak and Weizenbaum demonstrated it is easy to create the illusion of intelligence.

Whenever developers talk about AI, ask questions. Although "AI" has no generally accepted definition, it may mean something specific to them.

We do not want computer systems that perform tricks; we need trustworthy tools. Trustworthy systems must be based on sound mathematics and science, not heuristics or illusionist's tricks.

Conclusion

Whenever developers talk about AI, ask questions. Although "AI" has no generally accepted definition, it may mean something specific to them. The term "AI" obscures the actual mechanism but, while it often hides sloppy and untrustworthy methods, it might be concealing a sound mechanism. An AI might be using sound logic with accurate information, or it could be applying statistical inference using data of doubtful provenance. It might be a well-structured algorithm that can be shown to work correctly, or it could be a set of heuristics with unknown limitations. We cannot trust a device unless we know how it works.

AI methods are least risky when it is acceptable to get an incorrect result or no result at all. If you are prepared to accept "I don't understand" or an irrelevant answer from a "personal assistant," AI is harmless. If the response is important, be hesitant about using AI.

Some AI programs almost always work and are dangerous because we learn to depend on them. A failure may go undetected; even if failures are detected, users may not be prepared to proceed without the device.

Do not be misled by demonstrations: they are often misleading because the demonstrator avoids any situations where the "AI" fails. Computers can do many things better than people. Humans have evolved through a sequence of slight improvements that need not lead to an optimal design. "Natural" methods have evolved to use our limited sensors and actuators. Modern computer systems use powerful sensors and remote actuators, and can apply mathematical methods that are not practical for humans. It seems very unlikely that human methods are the best methods for computers.

When Alan Turing rejected "Can machines think?" as unscientific, and described a different question to illustrate what he meant by "scientific," he was right but misled us. Researchers working on his "replacement question" are wasting their time and, very often, public resources. We don't need machines that simulate people. We need machines that do things that people can't do, won't do, or don't do well.

Instead of asking "Can a computer win Turing's imitation game?" we should be studying more specific questions such as "Can a computer system safely control the speed of a car when following another car?" There are many interesting, useful, and scientific questions about computer capabilities. "Can machines think?" and "Is this program intelligent?" are not among them.

Verifiable algorithms are preferable to heuristics. Devices that use heuristics to create the illusion of intelligence present a risk we should not accept. C

References

1. Turing, A.M. Computing machinery and intelligence. *Mind 59* (1950), 433–460.
2. Weizenbaum, J. Automating psychotherapy. ACM Forum Letter to the Editor. *Commun. ACM 17*, 7 (July 1974), 425; doi: 10.1145/361011.361081.
3. Weizenbaum, J. ELIZA—A computer program for the study of natural language communication between man and machine. *Commun. ACM 9*, 1 (Jan. 1966), 36–45; doi: 10.1145/365153.365168.

David Lorge Parnas works for Middle Road Software, Inc., in Ottawa, Canada. He is Professor Emeritus at McMaster University in Canada and the University of Limerick in Ireland.

Lillian Chik-Parnas, Nancy Leveson, Peter Denning, and Peter Neumann offered helpful suggestions about earlier drafts of this column.