DOI:10.1145/3567605

Peter J. Denning and John Arquilla

# The Profession of IT

# The Context Problem in Artificial Intelligence

*The artificial intelligence design challenge of teaming humans and machines is difficult because machines cannot read the context of use.*

DESPITE THE MANY wondrous uses of artificial intelligence (AI), examples of "fragile" AI are increasingly common. Facial recognition systems make serious errors when presented with images of people of color. Road sign recognizers mistake bird-stained stop signs for speed limit signs. Bias in training datasets skews to unfair decisions. Large-language models spew out fluent text that on closer inspection makes little sense. Adversaries successfully confuse AI systems by corrupting sensor data. Automatic missile defense systems have sometimes mistaken commercial aircraft for enemy warplanes. Military drones cannot reliably distinguish noncombatants in an operation.[3]

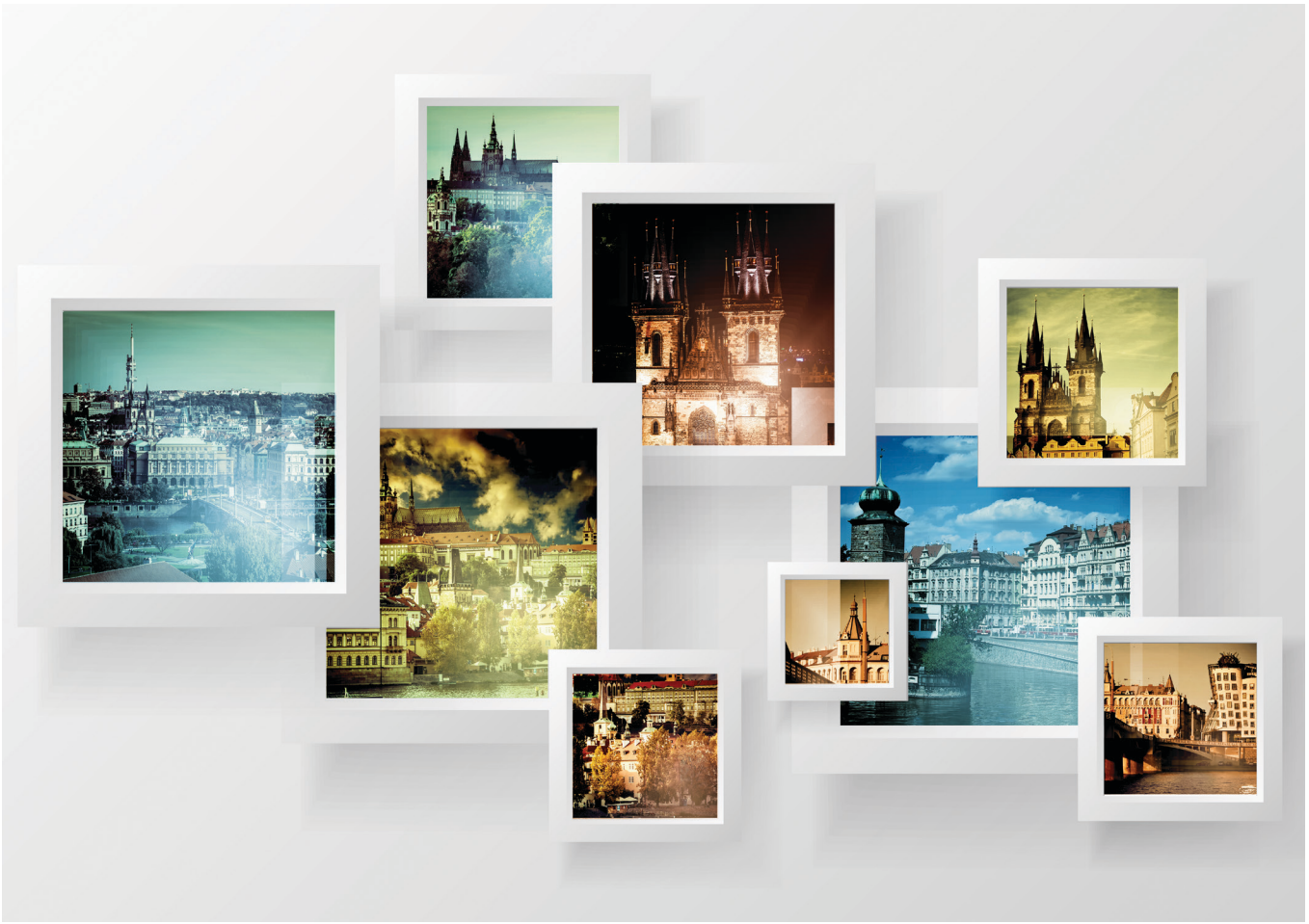
Such examples have made urgent the question: "When can you trust an AI?" This question has become a big challenge for professionals designing AI in response to client concerns about the safety and reliability of AI. A frequent feature of these failures is the machines were being used in a context different from what they were designed or trained for. In this column, we will discuss this context problem and strategies to mitigate it.

## A High-Stakes Example

The military desires to avoid civilian casualties when using drones to strike military targets. Drone guidance algorithms analyze satellite and other data seeking "signatures," or movement patterns that distinguish combatants from civilians. A human operator decides whether to strike based on the algorithm's recommendation. Even with a human in the loop, drone strikes frequently mistake civilians for enemy combatants. A horrible example occurred in August 2021 during the U.S. withdrawal from Afghanistan. The signature algorithm tagged a car approaching the airport as a likely suicide bomber and the human operator authorized the drone strike—killing all the car's occupants, an innocent family seeking to evacuate. The news reports commented the operator had a doubt about whether the car was a danger. Something in the context of the actual situation created a doubt. The algorithm could not detect that subtle change, and the human operator did not have time to sort it through in the very short decision window. Effectively, the algorithm's "judgment" determined the outcome.

**The idea of pairing machines with people to amplify human capabilities is not new.**

Could the algorithm be designed to avoid this kind of failure? Presumably, the designers encoded all the available knowledge about the context of use into the software and its training algorithms. But that was not



enough. The question is: Can the machine detect there has been a change in the environment and notify the operator? Can a machine read context in real time and adapt its recommendations?

This question has plagued AI pretty much since the beginning.

### First Appearance of the Problem in Early AI

In the late 1970s, AI research gave birth to rule-based expert systems, software that would perform at the level of a human expert. Over the years, some very competent expert systems were built, notably in medical diagnosis and machine repair, but none attained expert-level performance.[1] Dreyfus argued this failure is inherent because human experts do not use known rules to guide their actions, and in many cases their actions may not be describable by any rule. And even when human experts can articulate their rules, they cannot articulate how they tell when a rule is relevant to the situation at hand. Relevance is critical: expert action is situated in a context that the expert can sense but the machine cannot. Dreyfus's stand was very controversial because many believed that the problem was an insufficiency of rules and facts, not a fundamental flaw in their thinking about expertise.

AI has always been plagued with what has been called the "frame problem." AI systems tend to make mistakes when placed in a new context (new frame) from that in which they were programmed or trained.[4] No good solution to this problem has been found.

**There is plenty of room for competent AI-based systems.**

This old debate has resurfaced with the modern Artificial Neural Network (ANN). Some claim the ANN and its training algorithm, built around neurons that simulate brain structures, is not constrained by the assumption that its rules and facts are given externally. These networks have the ability to generate their own rules. Unfortunately, it appears that it is impossible to tell which internal connections (rules) matter in a given behavior of the ANN. Moreover, once the ANN is trained, its internal rules become fixed and do not adapt to a new situation of use. The training algorithm would have to be rerun with data from the new situation in the hope that the retrained network would respond better in the next context. The key part is the ANN cannot adapt fast enough because the training algorithms are too slow to meet real-time requirements.

Human machine teaming is a proposed solution to this dilemma. The idea is that humans are good at reading context to inform relevant and sound judgments, but not good at

fast calculation, whereas machines are good at calculation but not reading context. The context-free nature of machines is the consequence of their design. Anything a machine needs to "know" is encoded into local data stored for fast retrieval. This "locality of connection" confers blazing speed—computing machines can perform a trillion operations in the time it takes a human to do just one—yet it denies the machine the ability to sense human context. So if we have tasks that both require good human judgment and speed, why not pair them up?

A quintessential instance of this idea is Freestyle Chess, founded by then-World Champion Grandmaster Garry Kasparov in 1997 after the IBM Big Blue machine beat him. Kasparov's new game pitted teams of good chess players armed with good chess programs against each other. Doing so brought chess to a whole new level. Human-machine teams often won outright when playing against an unaided machine.

Douglas Engelbart famously founded the subfield of computing for augmented intelligence, in which machines enhance human intelligence but do not replace it. A recent illustration of Engelbart's principle was seen at a military exercise at Fort Benning in 2019, when a small unit, augmented by AI able to parse large amounts of sensor data quickly, defeated a much larger, conventional non-AI equipped team.[2]

### HCI Is Essential

The idea of pairing machines with people to amplify human capabilities is not new. The field of Human-Computer Interaction (HCI) emerged in the 1970s to provide principles for designing the interface. HCI has two broad objectives. First, enable humans to interact with machines as tools in the hands of a skilled operator, with no pretense of communicating with an intelligent entity. Second, design so that human psychology and cognitive bias do not cause the human operator to be misinformed or confused. In the worst case, a confused human operator cannot function, or rather "malfunctions," which can cost lives. The NASA "Cockpit of the Future" project, begun in the 1980s, illustrates a design meeting both these criteria.[a] For HMT to be successful, it must bring in HCI.

### The Context Problem

Context is the nub of all these problems. Defining what human context is has long been an occupation of philosophers. They refer to a "background of obviousness"—things we humans know and that are obvious to us but, since they have not been stated explicitly, cannot be captured as rules and are therefore not obvious to a machine. It is like ancestral knowledge passed down through the ages through community interactions, practices, stories, and beliefs without ever being explicitly stated. This background spawns and cultivates virtues such as empathy, compassion, caring, deciding relevance, social norms, ethical sensibility, and more. No one has come close to designing a machine with any of these virtues. Looked at this way, it is difficult to see how a machine could possibly master human context.

The usual explanation for the fragility of ANNs is "bias." A bias in the training data disposes the ANN to making mistakes when applied in new situations that are poorly represented in the training data. Another way to interpret this is the ANN cannot function well in a new context in which it has not been trained. The

a See https://go.nasa.gov/3e9ABEo

**Context is the nub of all these problems. Defining what human context is has long been an occupation of philosophers.**

common assumption that training in one context transfers to a new context is wrong. Given that ANNs are unable to read context of use during use, it is no surprise that the ANN cannot detect that it is being used improperly and warn the user. This places a responsibility on designers to thoroughly test an ANN in its context of use to be sure it performs as expected and can be trusted.

Our conclusion is that computing machines are useful not because in some ways they resemble humans, but because in most ways they are utterly different from humans. We do not choose a machine over our colleague because the machine is faster, we choose the machine because it can do what our colleague cannot possibly do at all.

## Pairing Principles

Let's put all this to work with four principles for designing trustworthy couplings of human and machines.

▸ *Design for the differences.* Design with the understanding that humans and machines are utterly different kinds of entities. Humans are context sensitive, machines context free. Humans and machines can pair up but, to use the baseball metaphor, they play different positions on the team. Assign tasks requiring awareness and sensibility of context to humans, and automatable tasks to the machine. This division of tasks is consistent with the goal of amplifying and augmenting human capabilities, not replacing them. This is a difficult design problem because in the complex and the often-high tempos of operation, it is easy for a human to misinterpret signs and signals from the machine. The long tradition of HCI gives much guidance on how to do this.

▸ *Design to overcome human cognitive blindness and bias.* It is wrong to conclude that AI machines are too fragile and quirky to be valuable when paired with humans. The machine can search a huge state space of possible futures in a very short time and reveal possibilities that no human might ever find because of inherent cognitive biases. This is what happened with the advanced program AlphaGo, which became so

**Computing machines are useful not because in some ways they resemble humans, but because in most ways they are utterly different from humans.**

proficient that it beat the world's Go champion; the pivotal move in a tournament game was considered absurd by human players. A similar conclusion was reached in the late 1990s when chess programs started beating grand masters. FreeStyle Chess turned the tables, enabling machine-aided humans to beat grand master computers.

The chess example is important, given that "chessbots" have supreme tactical skill in calculating sequences of moves, but they cannot attain human strategic thinking. But in partnering the machine with the human, the cognitive and motivated biases so common to the human (for example, seeing what one expects or wants to see) are eliminated by the relentless searching of the AI, and the strategic guidance of the human further actualizes the potential of the AI. The revelation of possibilities to which we are blind is not an act of machine creativity, but of superior searching.

▸ *Design to exploit simulations.* Exploit the power of the machine to run complex simulations and scenarios. Given a set of assumptions, the machine can show what futures are possible. Humans then decide whether that future is desirable and, if so, how to get there.

▸ *Test and evaluate relentlessly.* Do as we have always done with engineered systems. Test them thoroughly under the same conditions as they will operate in. Do not trust that a machine that worked well in one context will work well in another.

We prefer the term pairing to teaming. Pairing recognizes the best-designed AI uses computing for the tasks that need speed, and humans for judgment. But it should be noted that AI can also enhance judgment by providing a lens to look at problems that is less distorted by human cognitive biases. The failure of ANNs in new contexts powerfully reveals human cognitive biases in the selection of the training data.

## Conclusion

We reach four main conclusions. Neural networks are unlikely to perform at the level of human experts. Machines, such as neural networks, cannot sense or read context and must be thoroughly tested when transferred to a new operating context. The term "team" is misleading when applied to humans pairing with machines. The HCI field offers many design principles that can make human-machine pairing successful.

There is plenty of room for competent AI-based systems. In many applications, no more than that is needed. But when expert-level actions and decisions are required, the machines are not up to the task. There is nothing better than a human expert augmented with a competent machine. C


**References**

1. Dreyfus, H. *What Computers Cannot Do.* MIT Press, 1972; second edition 1992.
2. Freedberg, S.J., Jr. AI and robots crush foes in Army wargame. *Breaking Defense* (Dec. 19, 2019).
3. Jatho, E.W. and Kroll, J.A. Artificial intelligence: Too fragile to fight? *USNI Proceedings* (Feb. 2022); https://bit.ly/3CHa36N
4. Shannon, M. The Frame Problem. *The Stanford Encyclopedia of Philosophy* (Spring 2016 edition); https://stanford.io/3MehHIM



**Peter J. Denning** (pjd@nps.edu) is Distinguished Professor of Computer Science and Director of the Cebrowski Institute for Information Innovation at the Naval Postgraduate School in Monterey, CA, USA, is Editor of ACM *Ubiquity*, and is a past president of ACM. His most recent book is *Computational Thinking* (with Matti Tedre, MIT Press, 2019). The author's views expressed here are not necessarily those of his employer or the U.S. federal government.

**John Arquilla** (jarquilla@nps.edu) is Distinguished Professor Emeritus of Defense Analysis at the Naval Postgraduate School in Monterey, CA, USA, and is the author most recently of *Bitskrieg: The New Challenge of Cyberwarfare* (Polity, 2021).