

A Leap from Artificial to Intelligence

CARISSA SCHOENICK ET AL.'S article "Moving Beyond the Turing Test with the Allen AI Science Challenge" (Sept. 2017) got me thinking ... less about the article itself than about the many articles on artificial intelligence we see today. I am astonished that people who know what computers can do, and, especially, how they do it, still think we (humankind) will ever create a rational being, much less that the day is near.

I see no such sign. A program that can play winning chess or Go is not one. We all knew it would happen sooner or later. We are talking about a large but finite set of paths through a well-defined set. Clever algorithms? Sure. But such things are the work of engineers, not of the computer or its programs.

Siri is no more intelligent than a chess program. I was indeed surprised, the first time I tried it, by how my iPhone seemed to understand what I was saying, but it was illusory. Readers of *Communications* will have some notion of its basic components—something that parses sound waves from the microphone and something that looks up the resulting tokens—and if a sound token is within, say, 5% of the English word ... you know what must be happening. The code is clever, that is, cleverly designed, but just code.

Neither the chess program nor Siri has awareness or understanding. A game-playing program does not know what a "game" is, nor does it care if it wins or loses. Siri has no notion of what a "place" is or why anyone would want to go there.

By contrast, what we are doing—reading these words, asking maybe, "Hmm, what is intelligence?" is something no machine can do. We are considering ideas, asking, say, "Is this true?" and "Do I care?"

That which actually knows, cares, and chooses is the *spirit*, something every human being has. It is what distinguishes us from animals, and from computers. What makes us think we can create a being like ourselves? The

leap from artificial to intelligence could indeed be infinite.

Arthur Gardner, Scotts Summit, PA

Carissa Schoenick et al. (Sept. 2017) described an AI vs. eighth-grade science test competition run in 2015 by the Allen Institute for Artificial Intelligence. The top solution, as devised by Chaim Linhart, predicted the correctness of most answers through a combination of gradient-boosting models applied to a corpus. Schoenick et al. suggested that answering eighth-grade-level science-test questions would be more effective than Alan Turing's own historic approach as a test of machine intelligence. But could there be ways to extend the Turing Test even further, to, say, real-life scenarios, as in medicine?

Consider that electronic health records (EHRs) collected during a typical clinical-care scenario represent a largely untapped resource for studying diseases and associated co-occurring diseases at a national scale. EHRs are a rich data source, with thousands of data elements per patient, including structured elements and clinical notes, often spanning years. Because EHRs include sensitive personal information and are subject to confidentiality requirements, accessing such databases is a privilege research institutions grant only a few well-qualified medical experts.

All this motivated me to develop a simple computer program I call EM-RBots to generate a synthetic patient population of any size, including demographics, admissions, comorbidities, and laboratory values.² A synthetic patient has no confidentiality restrictions and thus can be used by anyone to practice machine learning algorithms. I was delighted to find one of my synthetic cohorts being used by other researchers to develop a novel neural network that performs better than the popular long short-term memory neural network.¹

In the EHR context, though a human physician can readily distinguish between synthetically generated and real live human patients, could a ma-

chine be given the intelligence to make such a determination on its own?

Health-care institutions increasingly must be able to identify authentically human patients' EHRs. In light of this trend, hackers might want to generate synthetic patient identities to exploit the data of real patients, hospitals, and insurance companies. For example, a hacker might want to falsely increase a particular hospital's congestive heart failure (CHF) 30-day readmission rate, a measure of quality. Medicare also uses it to evaluate U.S. hospitals—by posting into the hospital's database synthetic-patient identities associated with CHF readmissions.

Before synthetic patient identities become a public health problem, the legitimate EHR market might benefit from applying Turing Test-like techniques to ensure greater data reliability and diagnostic value. Any new techniques must thus consider patients' heterogeneity and are likely to have greater complexity than the Allen eighth-grade-science-test is able to grade.

References

1. Baytas, I., Xiao, C., Zhang, X., Wang, F., Jain, A., and Zhou, J. Patient subtyping via time-aware LSTM networks. In *Proceedings of the 23rd SIGKDD Conference on Knowledge Discovery and Data Mining* (Halifax, NS, Canada, Aug. 13–17). ACM Press, New York, 2017, 65–74.
2. Kartoun, U. A methodology to generate virtual patient repositories. *arXiv Computing Research Repository*, 2016; <https://arxiv.org/ftp/arxiv/papers/1608/1608.00570.pdf>

Uri Kartoun, Cambridge MA

I Am Not a Number

Solon Barocas's and danah boyd's Viewpoint "Engaging the Ethics of Data Science in Practice" (Nov. 2017) did well to focus on ethics in data science, as data science is an increasingly important part of everyone's life. Data, ethics, and data scientists are not new, but today's computational power magnifies their scale and resulting social and economic influence. However, by focusing narrowly on data scientists, Barocas and boyd missed several important sides of the ethics story.

The disjoint they identified between

data scientists and researchers who critique data science is a strawman. For example, Cathy O’Neil, whose book *Weapons of Math Destruction* Barocas and boyd turned to for examples of unethical algorithms, is not just a researcher who happens to focus on data ethics, as they described her, but is herself a data scientist. Researchers like O’Neil would thus seem to be part of the solution to the problem Barocas and boyd identified.

Barocas and boyd also did not mention capitalism—the economic force behind much of today’s data science—with its behavioral prediction and manipulative advertising. What if the answer to unethical algorithms is to not create them in the first place? If an algorithm yields biased results based on, say, ethnicity, as Barocas and boyd mentioned, then surely the answer would be to not develop or use it to begin with; that is, do no harm. But such an approach also means the algorithm writer cannot sell the algorithm, an option Barocas and boyd ignored. They said data scientists try “to make machines learn something useful, valuable...” yet did not identify who might find it useful or what kind of value might be derived. Selling an algorithm that helps “wrongly incarcerate” people has financial value for the data scientist who wrote it but negative social value for, and does major financial harm to, those it might help put in jail. Data scientists must do what “maximizes the...models’ performance” but for whom and to what end? Often the answer is simply to earn a profit.

If one can learn data science, one can should be able to use it ethically. More than once Barocas and boyd mentioned how data scientists “struggle” with their work but expressed no empathy for those who have been hurt by algorithms. They did say data scientists choose “an acceptable error rate,” though for some scenarios there is no acceptable error rate.

If data scientists do not use data science ethically, as Barocas and boyd wrote and as O’Neil has shown, they are indeed doing it improperly. What Barocas and boyd failed to suggest is such data scientists should not be doing data science at all.

Nathaniel Poor, Cambridge, MA

Why Whole Foods for Amazon

Michael A. Cusumano’s Viewpoint “Amazon and Whole Foods: Follow the Strategy (and the Money)” (Oct. 2017) looked to identify a financial strategy for Amazon.com’s June 2017 acquisition of Whole Foods, which Amazon must have seen as strategically advantageous because it paid far more than Whole Food’s market valuation at the time. However, the column ignored the crucial potential for cross-subsidization to harm competition in the grocery industry. The trade press has since reported sales of high-margin Amazon products, including Echo devices, at Whole Foods, along with deals expected to come later (such as Whole Foods discounts for customers who also purchase Amazon Prime). Using revenue from such products and services to lower the price for commodity, low-margin products like groceries is a classic anti-competitive strategy with dubious legal or ethical basis.

Andrew Oram, Arlington, MA

Address the Slacker Before the Hacker

Esther Shein’s news story “Hacker-Proof Coding” (Aug. 2017) deserves a clarification and a warning about assumptions. Software developers should recognize that the techniques Shein explored will locate code faults but need not be solely manual, expensive, or tedious and thus error-prone. Contrary to the sources Shein quoted, much of the process of looking for faults can be automated. Automated diagnosis of errors can be done for approximately 20% of what it now costs in terms of programmer time and financial expenditure, yielding immediate and significant ROI.

Regarding assumptions, software developers should also be aware that the threat to the world’s software systems is not just from hackers but also from slackers often within their own midst. Too many software developers simply do not put enough effort into understanding the context in which their code will operate. For example, consider what Zachary Tatlock of the University of Washington said to Shein, “... as software that verifies the beam power has not become too high ...” because software verifies only that some sensor output value is not too

high. Unfortunately, however, such verification software is not designed to verify first that the sensor is operating properly.

Moreover, software developers must learn to be more analytical about the specifications their software is being designed to obey. Making software conform to “some specification,” as Andrew Appel of Princeton University said to Shein, is foolish unless the specification is the right specification. In a large-system context, “right,” as Tatlock said, cannot be assured in advance. Developers must first create the code, then confirm it is consistent with all the other code with which it will eventually interoperate. Here, “confirm” means “consistent with the rules of logic, arithmetic, and semantics,” not just some code developer’s specification for only a piece of the ultimate system.

Jack Ring, Gilbert, AZ

Communications welcomes your opinion. To submit a Letter to the Editor, please limit yourself to 500 words or less, and send to letters@cacm.acm.org.

©2018 ACM 0001-0782/18/1

Coming Next Month in **COMMUNICATIONS**

The Next Phase in the Digital Revolution: Platforms, Automation, Growth, Employment

Elements of the Theory of Dynamic Networks

Titus: Introducing Containers to the Netflix Cloud

Views from the Top

The Theory of Dynamic Networks

Plus the latest news about quantum encryption, the value of data, and the continuing education of software.