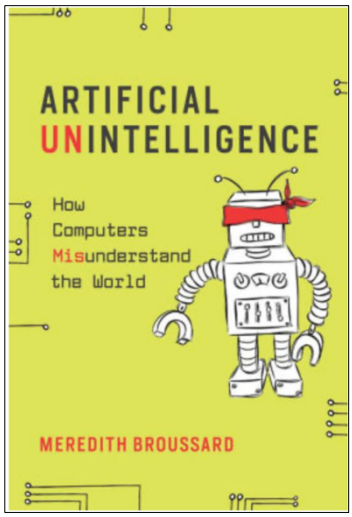
International Journal of Communication 15(2021), Book Review 765–767 1932–8036/2021BKR0009

Meredith Broussard, **Artificial Unintelligence: How Computers Misunderstand the World**, Cambridge, MA: MIT Press, 2018, 248 pp., $15.95 (paperback).

Reviewed by
Mary Heath
University of Illinois at Chicago



What if new technology isn't the answer to all of life's problems? In ***Artificial Unintelligence: How Computers Misunderstand the World***, author Meredith Broussard names this overreliance on technology "technochauvinism," and challenges her readers to question what types of problems computers are, and are not, effectively built to address (pp. 7–8). The author is an assistant professor in the Arthur L. Carter Journalism Institute at New York University, and a self-described "data journalist" (p. 6). Her work has been presented in the *Atlantic*, *Slate*, and the *Washington Post* among other sources. Immersed in discussion of technolibertarianism, analysis of the ideologies that bolster not only technochauvinism but tech culture as a whole, and a deep understanding of computer programming, Broussard uses *Artificial Unintelligence* to describe how a lack of understanding about how computers work leads to the application of technology to all types of issues, regardless of their appropriateness for the task.

Central to Broussard's argument is establishing a common understanding of what computer programming truly is, the formation and application of commands and algorithms which are based in math. With this in mind, *Artificial Unintelligence* questions the effectiveness of math (and therefore computers) to address the wide range of issues to which technochauvinists attempt to apply it. Fundamentally, since computer programming is entirely based on math, it is not the best way to approach all problems, especially social issues and problems that require adaptability rather than straightforward quantitative analysis. Although Broussard acknowledges that the limitations of technology are clear to many, she argues that the way that programming is discussed in daily life does not reflect this awareness.

In particular, Broussard notes that within technoculture "positive asymmetry" encourages the sharing of supportive comments, while people who critique the potential of new technologies are shunned, encouraging designers to ignore the possibly problematic implications of the technologies they are creating (p. 28). In addition, Broussard clarifies the distinction between general and narrow artificial intelligence (AI). General AI describes the science-fiction depiction of AI, which will result in computers having humanoid thinking processes and emotional reactions. In contrast, narrow AI refers to the way that current programmers use the term AI, describing a much more limited function of machine learning. The lack of verbal distinction between the technological imaginary of general AI and the reality of narrow AI has reinforced magical thinking about what computers can achieve. For example, Broussard notes that while the term "machine learning" implies a specific process to programmers, this language can be confusing to the everyman, to whom it implies a natural (humanlike) learning process, reinforcing the imaginary of general AI (p. 32).

Broussard delves further into the notion of how language and narratives surrounding technology impact real life by describing the practice of data journalism. Data journalism seeks to apply knowledge of algorithms and computational systems to the existing field of journalism and is premised on a strong understanding of where data originates from, how it is interpreted, and the impact and application of those interpretations. At its core, all data is dirty. This is true because not all factors being input can be properly quantified, data is often missing, and even existing data is socially constructed. People create data, not only through information categorization but also by literally creating the data that is used in training for machine learning. This is exemplified in chapter 3, where Broussard discusses how programmers used 30 million games played by human participants to train their program "Alpha Go" to determine algorithmically how to win at Go. Although the completed project is efficient in winning games, such AI programs would not be possible without the millions of hours of human labor used to create the data they are trained on (p. 36). This leads to numerous issues, since flawed data input can lead to flawed output. In chapter 3, Broussard provides an example of how people are needed to input, update, and maintain data by exploring the way that the lack of up-to-date data about textbook availability in lower-income school districts negatively impacts standardized testing scores. Specifically, she notes that even districts that did, in fact, have the textbooks that the tests are based on did not always have up-to-date records of those books, resulting in new books remaining unopened in closets, of no use to the students they were purchased to serve. Beyond the impact that everyday people have on the creation and use of data, programmers have a massive impact on data and its applications since computers act as proxies for the people who code them.

Broussard argues that the impact that programmers have on the real world necessitates increased diversity in mathematics and computational sciences. Chapter 6 covers the history of technology and mathematics culture, tracking the impact of notable personalities, such as Minsky and Turing, on the way that expertise is viewed both within and outside of the field. Broussard argues that our culture expects people with expertise in one area to be experts in all things. This misconception is bolstered by White male bias and the promotion of "genius" over hard work, both of which minimize the importance of social skills and reinforce a homogenous culture, leading to diversity dropout. The resulting culture, along with the positive asymmetry discussed above, leads to magical thinking that promotes technochauvinism. Broussard exemplifies this danger by describing how the push for self-driving cars is a public health crisis, endangering people's safety in the attempt to advance technology on principle.

Regarding the dangers of technochauvinism, Broussard also notes that this worldview leads to the reinforcement of social inequality and questions who technology truly makes the world better for. Touching on how algorithms inherently reinforce existing inequalities by reproducing the disparities existing in their input data, Broussard goes one step further to discuss how focusing economic resources on the production and adoption of new technology, rather than utilizing existing lower-technological systems (like books), is not only less efficient as a whole but also disproportionately disadvantages those without the money to integrate and maintain the new technologies. Thus, she encourages readers to recognize the limitations of programming and to integrate human-in-the-loop systems to design the best application of what computers and people are each best suited for. Noting that technology can fail and that even its successes are often attributable to false causality and "unreasonable effectiveness," *Artificial Unintelligence* argues that humans and computers working together, each playing to their own strengths, is more effective than computers or humans working alone and is an effective solution to avoiding the pitfalls of technochauvinism (p. 175).

Although I personally wish that Broussard had gone further in her critique of standardized testing and the quantification of education as a whole during chapter 5, overall, the examples used throughout the book are successful in revealing the large-scale problems inherent in attempts to employ programming to solve social issues. The author uses her experience as a data journalist to present these examples with skill and discusses the complications of computer programming in accessible language, presenting complex ideas in manageable bites.

Overall, *Artificial Unintelligence* is best suited to readers with some background in coding, although no expertise is necessary to grasp the broader concepts of the book. Furthermore, the author's useful suggestions on how to navigate the text expand its potential audience. The book is well written, includes grounded examples, and presents an accessible discussion of hardware, software, and real-life applications of programming to readers of all backgrounds. In particular, it is suggested to those with an interest in computer science and programming, journalists seeking to integrate data analysis into their work, and scholars who are interested in how computers and other technologies are relevant to Actor Network Theory, although the implications are relevant to anyone concerned with the impact of technology on social life.