

Digital Keywords |

*A Vocabulary of Information
Society and Culture*

Edited by
Benjamin Peters

Princeton University Press
Princeton and Oxford

Contents

Acknowledgments xi

Introduction

Benjamin Peters xiii

- 1 Activism**
Guobin Yang 1
- 2 Algorithm**
Tarleton Gillespie 18
- 3 Analog**
Jonathan Sterne 31
- 4 Archive**
Katherine D. Harris 45
- 5 Cloud**
John Durham Peters 54
- 6 Community**
Rosemary Avance 63
- 7 Culture**
Ted Striphas 70
- 8 Democracy**
Rasmus Kleis Nielsen 81
- 9 Digital**
Benjamin Peters 93
- 10 Event**
Julia Sonnevend 109
- 11 Flow**
Sandra Braman 118

- 12 Forum**
Hope Forsyth 132
 - 13 Gaming**
Saugata Bhaduri 140
 - 14 Geek**
Christina Dunbar-Hester 149
 - 15 Hacker**
Gabriella Coleman 158
 - 16 Information**
Bernard Geoghegan 173
 - 17 Internet**
Thomas Streeter 184
 - 18 Meme**
Limor Shifman 197
 - 19 Memory**
Steven Schrag 206
 - 20 Mirror**
Adam Fish 217
 - 21 Participation**
Christopher Kelty 227
 - 22 Personalization**
Stephanie Ricker Schulte 242
 - 23 Prototype**
Fred Turner 256
 - 24 Sharing**
Nicholas A. John 269
 - 25 Surrogate**
Jeffrey Drouin 278
- Appendix: Over Two Hundred Digital Keywords* 287
About the Contributors 291
Index 297

2

Algorithm

Tarleton Gillespie

In *Keywords*, Raymond Williams highlights how important terms change over time. But for many of the “digital keywords” here, just as important is the simultaneous use of a term by different communities, particularly inside and outside of technical professions, who seem often to share common words but speak different languages. Williams points to this concern too: “When we come to say ‘we just don’t speak the same language’ we mean something more general: that we have different immediate values or different kinds of valuation, or that we are aware, often intangibly, of different formations and distributions of energy and interest” (1976/1983, 11).

In the case of *algorithm*, the technical specialists, the social scientists, and the broader public are using the word in different ways. For software engineers, algorithms are often quite simple things; for the broader public they are seen as something unattainably complex. For social scientists, *algorithm* lures us away from the technical meaning, offering an inscrutable artifact that nevertheless has some elusive and explanatory power (Barocas, Hood, and Ziewitz 2013, 3). We find ourselves more ready to proclaim the impact of algorithms than to say what they are. This is not to say that critique requires a settled, singular meaning, or that technical meanings necessarily trump others. But we should be cognizant of the multiple meanings of *algorithm* as well as the discursive work the term performs.

To chase the etymology of the word is to chase a ghost. It is often said that the term *algorithm* was coined to honor the contributions of ninth-century Persian mathematician Muḥammad ibn Mūsā al-Khwārizmī, noted for having developed the fundamental techniques of algebra. It is probably more accurate to say that it developed from or with the word *algorism*, a formal term

for the Hindu-Arabic decimal number system, which was sometimes spelled *algorithm*, and which itself is said to derive from a French bastardization of a Latin bastardization of al-Khwārizmī's name, *Algoritmi*. Either way, it is something beyond irony that *algorithm*, which now drops its exotic flavor into Western discussions of the information society, honors an Arabic mathematician from the high court of Baghdad. The decimal number system he helped popularize also introduced the concept of zero, or *sifr* in Arabic. Perhaps it is fitting that al-Khwārizmī also has a crater on the moon named after him, a kind of astronomic zero. Like his crater and the zero concept he championed, the term *algorithm* will turn out to be important in part because it is vacant, a cypher, a ghostly placeholder upon which computational systems now stand.

Algorithm as a Trick

As we try to pinpoint the values that are enacted, or even embedded, in computational technology, it may in fact not be the algorithms that we need be most concerned about—if what we meant by *algorithm* was restricted to software engineers' use of the term. For the makers of algorithms, the term refers specifically to the logical series of steps for organizing and acting on a body of data to quickly achieve a desired outcome. MacCormick (2012), explaining algorithms to a general audience, calls them “tricks” (5), by which he means “tricks of the trade” more than tricks in the magical sense—or perhaps like magic, but as a magician understands it. An algorithm is a recipe composed in programmable steps; most of the “values” that concern us lie elsewhere in the technical systems and the work that produces them.

For its designers, the algorithm comes only after the generation of a “model.” The model is the formalization of a problem and its goal, articulated in computational terms. So the goal of *giving users the most relevant search results for their query* might be modeled as, or approximated into operationalized terms as, *efficiently calculating the combined values of preweighted objects in the index database, in order to improve the percentage likelihood that users click on one of the first five results*.¹ The complex social activity and the values it holds dear are translated into a functional interaction of variables, steps,

and indicators. What was a social judgment—“What’s relevant?”—gets modeled: posited and measurable relationships, actionable and strategic targets, and threshold indicators of success.

The algorithm, then, is merely the procedure for addressing the task as operationalized: steps for aggregating those assigned values efficiently, or making the matches rapidly, or identifying the strongest relationships according to some operationalized notion of “strong.” All is in the service of the model’s understanding of the data and what they represent, and the model’s goal and how it has been formalized. There may be many algorithms that would reach the same result inside a given model, just as bubble sorts and shell sorts will both alphabetize lists of words successfully. Engineers choose between them based on “technical” values such as speed, system load, perhaps their computational elegance. The embedded values that make a sociological difference are probably more about the problem being solved, the way it has been operationalized, the goal chosen, and the way that goal has been operationalized (Rieder 2012).

Of course, simple alphabetical sorting is a misleading example to use here. The algorithms we’re concerned about today are rarely designed to reach a single and certifiable answer, like a correctly alphabetized list. Most common algorithms produce no certifiably “correct” results at all but only turn out results based on many possible pathways. Algorithm designers are not pursuing correctness; they’re pursuing some threshold of operator or user satisfaction—understood in the model, perhaps, in terms of percent clicks on the top results; or percentage of correctly identified human faces from digital images.

Contemporary algorithms, especially those involved in some form of machine learning, are also “trained” on a corpus of existing data. These data have been in some way certified, either by the designers or by past user practices: this photo is of a human face, this photo is not; this search result has been selected by many users in response to this query, this one has not. The algorithm is then run on these data so that it may “learn” to pair queries and results found satisfactory in the past, or to distinguish images with faces from images without.

The values and assumptions that go into the selection and preparation of these training data may be of much more importance to

our sociological concerns than the algorithm that's learning from them. For example, the training data must be a reasonable approximation of the data that algorithm will operate on in the wild. The most common problem in algorithm design is that the training data turn out not to match the data being operated on in the wild, in some consequential way. Sometimes new phenomena emerge that the training data simply did not include and could not have anticipated; just as often, something important was overlooked as irrelevant, or was scrubbed from the training data in preparation for the development of the algorithm. Imagine a recognition algorithm trained on a corpus of selfies, but the photo archive came from an online service that is used disproportionately by people of particular races. The algorithm designed may later prove less accurate with a more diverse corpus of photos, and may therefore seem to have deeply problematic implications.²

Furthermore, improving an algorithm is rarely about redesigning it. Rather, designers “tune” an array of parameters and thresholds, each of which represents a tiny assessment or distinction. In search, this might mean the weight given to a word based on where it appears in a webpage, or assigned when two words appear in proximity, or given to words that are categorically equivalent to the query term. These thresholds can be dialed up or down in the algorithm's calculation of which webpage has a score high enough to warrant ranking it among the results returned to the user.

Finally, these exhaustively trained and finely tuned algorithms are instantiated inside of what we might call an application. For software engineers, the algorithm is the conceptual sequence of steps, which should be expressible in any computer language, or in human or logical language. They are then instantiated in code, running on servers somewhere, attended to by other helper applications (Geiger 2014), and triggered when a query comes in or an image is scanned. These applications may embody values as well, outside of their reliance on a particular algorithm.

To inquire into the implications of algorithms, if we mean what software engineers mean by the term, could only mean something so picky as investigating the political implications of using a bubble sort or a shell sort—and perhaps missing the bigger questions, like why alphabetical in the first place, or why train on this particular

data set. Perhaps there are lively insights to be had about the implications of different algorithms in this strict technical sense,³ but by and large we in fact mean something else when we talk about an algorithm having “social implications.”

Algorithm as Synecdoche

While it is important to understand the technical specificity of the term, *algorithm* has now achieved some purchase in the broader public discourse about information technologies, where it is typically used as an abbreviation for everything described above, combined: algorithm, model, target goal, data, training data, application, hardware. As Goffey puts it, “Algorithms act, but they do so as part of an ill-defined network of actions upon actions” (2008, 19). It is this ill-defined network to which our more common use of the term refers. And this technical assemblage stands in for, and often obscures, the people involved at every point: people debating the models, cleaning the training data, designing the algorithms, tuning the parameters, deciding on which algorithms to depend on in which context. “These algorithmic systems are not standalone little boxes, but massive, networked ones with hundreds of hands reaching into them, tweaking and tuning, swapping out parts and experimenting with new arrangements. . . . We need to examine the logic that guides the hands” (Seaver 2013). Perhaps *algorithm* is coming to serve as the name for a particular kind of sociotechnical ensemble, one of a family of systems for knowledge production or decision making: in this one, people, representations, and information are rendered as data, are put into systematic/mathematical relationships with each other, and then are assigned value based on calculated assessments about them.

But what is gained and lost by using *algorithm* this way? Calling the complex sociotechnical assemblage an *algorithm* avoids the need for the kind of expertise that could parse and understand the different elements; a reporter may not need to know the relationships between model, training data, thresholds, and application in order to call into question the impact of that “algorithm” in a specific instance. It also acknowledges that, when designed well, an algorithm is meant to function seamlessly as a tool; perhaps it

can, in practice, be understood as a singular entity. Even algorithm designers, in their own discourse, shift between the more precise meaning and this broader use.

On the other hand, this conflation risks obscuring the ways in which political values may slip in elsewhere than at what designers call the *algorithm*. This helps account for the way many algorithm designers seem initially surprised by the interest of sociologists in what they do—because they may not see the values in their “algorithms” (precisely understood) that we see in their algorithms (broadly understood), because questions of value are very much bracketed in the early decisions about how to operationalize a social activity into a model, and lost in the minuscule, mathematical moments of assigning scores and tuning thresholds.

In our own scholarship, this kind of synecdoche is perhaps unavoidable. Unexamined, it reifies the very processes that constitute it. It is too easy to treat it as a singular artifact, when in the cases we’re most interested in, it’s rarely one tool, but many tools functioning together, sometimes different tools for different users, so complex that in some cases even their designers can no longer comprehend them.⁴ It also tends to erase the people involved, downplay their role, and distance them from accountability. In the end, whether this synecdoche is acceptable depends on our intellectual aims. Calling all these social and technical elements the *algorithm* may give us a handle with which to grip what we want to closely interrogate; at the same time it can produce a “mystified abstraction” (Striphas 2012) that, for other research questions, it might be better to demystify.

Algorithm as Talisman

The information industries often invoke the term *algorithm* to the public as well. To call a service or process an *algorithm* is to lend it a set of associations: mathematical, logical, impartial, consistent. Algorithms seem to have a “disposition towards objectivity” (Hillis, Petit, and Jarrett 2013, 37); this objectivity is regularly performed as a feature of algorithmic systems (Gillespie 2014). Conclusions described as having been generated by an algorithm wear a powerful legitimacy, much the way statistical data bolster scientific

claims. It is a different kind of legitimacy from one that rests on the subjective expertise of an editor or a consultant, though it is important not to assume that it trumps such claims in all cases. A market prediction that is “algorithmic” is different from a prediction that comes from expert brokers highly respected for their expertise and acumen; a claim about an emergent social norm in a community generated by an algorithm is different from one generated ethnographically. Each makes its own play for legitimacy, and implies its own framework for what legitimacy is (quantification or interpretation, mechanical distance or human closeness) (see [community](#)). But in the context of nearly a century of celebration of the statistical production of knowledge and long-standing trust in automated calculation over human judgment, the algorithmic does enjoy a particular cultural authority.

More than that, the term offers the corporate owner a powerful talisman to ward off criticism, when companies must justify themselves and their services to their audience, explain away errors and unwanted outcomes, and justify and defend the increasingly significant roles they play in public life (Gillespie 2012a). When critics say, “Facebook’s algorithm,” they often mean Facebook and the choices it makes, some of which are made in code. But information services can point to “the algorithm” as having been responsible for particular results or conclusions, as a way to distance those results from the providers (Morozov 2014, 142). The term generates an entity that is somehow separate, like the assembly line inside the factory, that can be praised as efficient or blamed for mistakes.

The term *algorithm* is also quite often used as a stand-in for its designer or corporate owner. This may be another way of making the earlier point, that the singular term stands for a complex sociotechnical assemblage: *Facebook’s algorithm* really means *Facebook*, and *Facebook* really means the people, things, priorities, infrastructures, aims, and discourses that animate the site. But it may also be a political economic conflation: this is Facebook acting through its algorithm, intervening in an algorithmic way, building a business precisely on its ability to construct complex models of social/expressive activity, train on an immense corpus of data, tune countless parameters, and reach formalized goals extremely efficiently. Facebook as a company often behaves algorithmically.

Maybe saying “Facebook’s algorithm” and really meaning the choices made by Facebook the company is a way to assign accountability (Diakopoulos 2013; Ziewitz 2011). It makes the algorithm theirs in a powerful way, reducing the distance some providers put between “them” (their aims, their business model, their footprint, their responsibility) and “the algorithm” (as somehow separate from all that). On the other hand, conflating the algorithmic mechanism and the corporate owner may obscure the ways these two entities are not always aligned. It is crucial that we distinguish between things done by the algorithmic system and things done in other ways, such as the deletion of obscene images from a content platform, which is sometimes performed algorithmically and sometimes manually (Gillespie 2012b). It is crucial to note slippage between a provider’s financial or political aims and the way the algorithmic system actually functions. And conflating algorithmic mechanism and corporate owner misses how some algorithmic approaches are common to multiple stakeholders, circulate among practitioners in specific ways, and embody a tactic that exceeds any one implementation.

Algorithm as Committed to Procedure

In recent scholarship, *algorithm* increasingly appears not as a noun but as an adjective. To talk about “algorithmic identity” (Cheney-Lippold 2011), “algorithmic regulation” (O’Reilly 2013), “algorithmic power” (Bucher 2012), “algorithmic ideology” (Mager 2012), “algorithmic culture” (Striphas 2010), or the “algorithmic turn” (Uricchio 2011) is to highlight a social phenomenon that is driven by and committed to algorithmic systems—which include not just algorithms themselves, but also the computational networks in which they function, the people who design and operate them, the data and users on which they act, and the institutions that provide these services.

What we are really concerned with when we invoke the “algorithmic” here is not algorithms per se, but the insertion of procedure into human knowledge and social experience. What makes something algorithmic is that it is produced by or related to an information system committed (both functionally and ideologically)

to the computational generation of knowledge or decisions. This requires the formalization of social facts into measurable data, and the “clarification” (Cheney-Lippold 2011) of social phenomena into computational models that operationalize both problem and solution. These often stand in as proxies for human judgment or action, meant to simulate it as nearly as possible. But the “algorithmic” intervenes in terms of step-by-step procedures that one (computer or human) can enact on this formalized information such that it can be computed. This process is automated so that it can happen instantly, repetitively, and across many contexts, away from the guiding hand of its implementers.

This is not the same as suggesting that knowledge is produced exclusively by a machine abstracted from human agency or intervention. Information systems are always swarming with people; we just can’t always see them (Downey 2014; Kushner 2013). And an assembly line might be just as “algorithmic” in this sense of the word, or at least the parallels are important to consider. What is central is the commitment to procedure, and the way procedure distances its human operators from both the point of contact with others and the mantle of responsibility for the intervention they make. It is a principled commitment to the “if/then” logic of computation.

Yet what does *algorithmic* refer to, exactly? To put it another way, what is it that is *not* algorithmic? What kind of “regulation” is being condemned as insufficient when Tim O’Reilly calls for “algorithmic regulation”? It would be all too easy to invoke the algorithmic as simply the opposite of what is done subjectively or by hand, or of what can be accomplished only with persistent human oversight, or of what is beholden to and limited by context. To do so would draw too stark a contrast between the algorithm and something either irretrievably subjective (if we are glorifying the algorithmic) or warmly human (if we’re condemning it). If “algorithmic” market predictions or search results are produced by a complex assemblage of people, machines, and procedures, what makes their particular arrangement feel different from other ways of generating information, which are also produced by a complex assemblage of people, machines, and procedures? It is imperative that we look more closely at those practices that precede or stand in contrast to those we posit as algorithmic, and recognize how they too combine the

procedural and the subjective, the machinic and the human, the measured and the ineffable. And it is crucial that we continue to examine algorithmic systems ethnographically, to explore how the systemic and the ad hoc coexist and are managed within them.

To highlight their automaticity and mathematical quality, then, is not to contrast algorithms to human judgment. It is to recognize them as part of mechanisms that introduce and privilege quantification, proceduralization, and automation in human endeavors. Our concern for the politics of algorithms is an extension of worries about Taylorism and the automation of industrial labor; about actuarial accounting, the census, and the quantification of knowledge about people and populations; and about management theory and the dominion of bureaucracy. At the same time, we sometimes wish for more “algorithmic” interventions when the ones we face are discriminatory, nepotistic, and fraught with error; sometimes procedure is truly democratic.

We rarely get to watch algorithms work; but picture watching complex traffic patterns from a high vantage point: it is clear that this “algorithmic” system privileges the imposition of procedure, and—to even participate in such a complex social interaction—users must in many ways accept it as a kind of provisional tyranny. The elements can be known only in operational terms; every possible interaction within the system must be anticipated; and stakeholders often point to the system-ness of the system to explain success and explain away failure. The system struggles with the tension between the operationalized aims and the way humanity inevitably undermines, alters, or exceeds those aims. The system is designed and overseen by powerful actors, though they appear only at specific moments of crisis. And it’s not clear how to organize such complex behavior in any other way and still have it be functional and fair. Commitment to the system and the complex scale at which it is expected to function makes us beholden to the algorithmic procedures that must manage it. From this vantage point, algorithms are merely the latest instantiation of the modern tension between ad hoc human sociality and procedural systemization—but one that is now powerfully installed as the beating heart of the network technologies we surround ourselves with and increasingly depend upon.

See in this volume: community, culture, digital, information, personalization, prototype

See in Williams: bureaucracy, determine, expert, hegemony, industry, institution, jargon, management, mechanical, pragmatic, standards, technology

Notes

- 1 This parallels Kowalski's well-known definition of an algorithm as "logic + control": "An algorithm can be regarded as consisting of a logic component, which specifies the knowledge to be used in solving problems, and a control component, which determines the problem-solving strategies by means of which that knowledge is used. The logic component determines the meaning of the algorithm whereas the control component only affects its efficiency" (Kowalski 1979, 424). I prefer to use "model" because I want to reserve "logic" for the underlying premise of the entire algorithmic system and its deployment.
- 2 This may help explain Google's racially charged image labeling blunder in 2015. See Dougherty 2015.
- 3 See Kockelman 2013 for a dense but superb example.
- 4 See Christian 2012.

References

- Barocas, Solon, Sophie Hood, and Malte Ziewitz. 2013. "Governing Algorithms: A Provocation Piece." <http://governingalgorithms.org/resources/provocation-piece/>.
- Bucher, Taina. 2012. "Want to Be on the Top? Algorithmic Power and the Threat of Invisibility on Facebook." *New Media & Society* 14(7): 1164–80.
- Cheney-Lippold, John. 2011. "A New Algorithmic Identity: Soft Biopolitics and the Modulation of Control?" *Theory, Culture & Society* 28(6): 164–81.
- Christian, Brian. 2012. "The A/B Test: Inside the Technology That's Changing the Rules of Business." *Wired*, April 25. http://www.wired.com/2012/04/ff_abtesting/.
- Diakopoulos, Nicholas. 2013. "Algorithmic Accountability Reporting: On the Investigation of Black Boxes." A Tow/Knight Brief. Tow Center for Digital Journalism, Columbia Journalism School. <http://towcenter.org/algorithmic-accountability-2/>.
- Dougherty, Conor. 2015. "Google Photos Mistakenly Labels Black People 'Gorillas'" *Bits Blog, New York Times*, July 1. <http://bits.blogs.nytimes.com/2015/07/01/google-photos-mistakenly-labels-black-people-gorillas/>.
- Downey, Gregory J. 2014. "Making Media Work: Time, Space, Identity, and Labor in the Analysis of Information and Communication Infrastructures."

- In *Media Technologies: Essays on Communication, Materiality, and Society*, edited by Tarleton Gillespie, Pablo J. Boczkowski, and Kirsten A. Foot, 141–66. Cambridge, MA: MIT Press.
- Geiger, R. Stuart. 2014. “Bots, Bespoke, Code and the Materiality of Software Platforms.” *Information, Communication & Society* 17(3): 342–56.
- Gillespie, Tarleton. 2012a. “Can an Algorithm Be Wrong?” *Limn* 1(2). <http://escholarship.org/uc/item/0jk9k4hj>.
- . 2012b. “The Dirty Job of Keeping Facebook Clean.” *Culture Digitally*, February 22. <http://culturedigitally.org/2012/02/the-dirty-job-of-keeping-facebook-clean/>.
- . 2014. “The Relevance of Algorithms.” In *Media Technologies: Essays on Communication, Materiality, and Society*, edited by Tarleton Gillespie, Pablo J. Boczkowski, and Kirsten A. Foot, 167–93. Cambridge, MA: MIT Press.
- Goffey, Andrew. 2008. “Algorithm.” In *Software Studies: A Lexicon*, edited by Matthew Fuller. Cambridge, MA: MIT Press.
- Hillis, Ken, Michael Petit, and Kylie Jarrett. 2013. *Google and the Culture of Search*. New York: Routledge.
- Kockelman, Paul. 2013. “The Anthropology of an Equation. Sieves, Spam Filters, Agentive Algorithms, and Ontologies of Transformation?” *HAU: Journal of Ethnographic Theory* 3(3): 33–61.
- Kowalski, Robert. 1979. “Algorithm = Logic + Control.” *Communications of the ACM* 22(7): 424–36.
- Kushner, Scott. 2013. “The Freelance Translation Machine: Algorithmic Culture and the Invisible Industry?” *New Media & Society* 15(8): 1241–58.
- MacCormick, John. 2012. *Nine Algorithms That Changed the Future*. Princeton, NJ: Princeton University Press.
- Mager, Astrid. 2012. “Algorithmic Ideology: How Capitalist Society Shapes Search Engines.” *Information, Communication & Society* 15(5): 769–87.
- Morozov, Evgeny. 2014. *To Save Everything, Click Here: The Folly of Technological Solutionism*. New York: PublicAffairs.
- O’Reilly, Tim. 2013. “Open Data and Algorithmic Regulation.” In *Beyond Transparency: Open Data and the Future of Civic Innovation*, edited by Lauren Goldstein and Lauren Dyson. San Francisco, CA: Code for America Press. <http://beyondtransparency.org/chapters/part-5/open-data-and-algorithmic-regulation/>.
- Rieder, Bernhard. 2012. “What Is in PageRank? A Historical and Conceptual Investigation of a Recursive Status Index.” *Computational Culture* 2. http://computationalculture.net/article/what_is_in_pagerank.
- Seaver, Nick. 2013. “Knowing Algorithms.” Paper presented at Media in Transition 8, Cambridge, MA. <http://nickseaver.net/papers/seaverMIT8.pdf>.
- Striphas, Ted. 2010. “How to Have Culture in an Algorithmic Age.” *The Late Age of Print*, June 14. <http://www.thelateageofprint.org/2010/06/14/how-to-have-culture-in-an-algorithmic-age/>.
- . 2012. “What Is an Algorithm?” *Culture Digitally*, February 1. <http://culturedigitally.org/2012/02/what-is-an-algorithm/>.
- Uricchio, William. 2011. “The Algorithmic Turn: Photosynth, Augmented Reality and the Changing Implications of the Image.” *Visual Studies* 26(1): 25–35.

- Williams, Raymond. 1976/1983. *Keywords: A Vocabulary of Culture and Society*. 2nd ed. Oxford: Oxford University Press.
- Ziewitz, Malte. 2011. "How to Think about an Algorithm: Notes from a Not Quite Random Walk." Discussion paper for Symposium on Knowledge Machines between Freedom and Control, September 29. http://ziewitz.org/papers/ziewitz_algorithm.pdf.