In “Algorithms, Performativity and Governability,” Lucas Introna asks us to consider the “situated ‘doing’” of algorithms, not so much what they are as what they perform—what they effectively enable and become—amid the mutually-becoming socio-technical contexts of their application. So, for instance, plagiarism detection algorithms don’t so much function as advertised (to detect plagiarism) as much as they instantiate a specific and very limited (detectable) version of what plagiarism is, a version mobilized within the unfolding landscape of commodification that today so successfully if insidiously frames education. Our focus is directed toward the “computational reductionism” that would at once produce and reduce the “problem” of plagiarism in student writing assignments to a question of alphanumeric character strings repeated (or not) across a vast, growing, and proprietary corpus of such writings. Lucas shows how this has dire implications that are both narrow—punishing cut-and-paste while ignoring other plagiary—and broad, as educators and others unwittingly accept/promote the commodification of education. The project of governability stumbles, it seems, because “The issue is not the details of the algorithm in isolation,” but rather the whole complex of doings or becoming. What to do? Google is the big question of course, and Lucas’s draft hints that our most productive course of action would not be to demand the regulation/revelation of its forever-changing search algorithms but instead to demand that Google “be explicit” about its business model and “give users the ability to manage their relationship with” the corporation.

I’m not sure I understand exactly what this last means yet in practical terms, but I am certainly persuaded by Lucas’s analysis, which—it seems to me—is at once an “embrace [of] ambiguity” (“Algorithms: A provocation piece” ¶ 7) and a corresponding displacement of specificity (“Algorithms: A provocation piece” ¶ 12) such that we forgo our concern for algorithmic particulars and learn instead to specify the socio-technical contexts of their application. Algorithms in general retain their mystery, while particular algorithms remain trade secrets. Let the specialists handle all that, I guess. What the rest of us can shoot for is a more thorough understanding of the production/reduction of problems within the logic of computation. How do phenomena become available for algorithmic analysis in the first place? What are the limitations and affordances of that availability? And in whose interests—eventual as well as immediate—are those limitations treated as inconsequential while those affordances become productive? As a Deleuzian might put it, “To submit a phenomenon to computation is to striate otherwise-smooth details, analog details, to push them upwards toward the sovereign, to make only high-level control available to the user, and then only those aspects of control that are deemed appropriate by the sovereign.” I am an English professor and no Deleuzian, so
let me simply say that the logic of computation requires that phenomena be reduced to data, sliced and diced into fields and tables, or—in the case of search—processed (“encoded”) as character strings.

If I can for a moment pursue a line of thinking that perhaps only an English professor could care about, we might well wonder whether the production/reduction of linguistic phenomena as/into character strings—in student papers but also online generally for algorithmic purposes—isn’t having unnoticed intellectual effects. One way to describe the professionalization of literary study over the last 200 years is as a series of “detailed engagements with the classical inheritance of rhetoric and philology,” really a prolonged and spasmodic struggle in which rhetoric keeps losing out and philology keeps gaining prestige. Banging ideas into character strings is simply the latest and perhaps final (hollow?) victory for a philological perspective, since online “hits”—words and phrases and sources found algorithmically by search, by term-extraction analyses, or other forms of text mining—are exclusively instances with particular character strings detectable within them or indexing them, not instances with a particular logic, say, or exhibiting particular rhetorical figures, emotive appeals, or styles of argument. Rhetoric resists computation, while philology is endemic to it, if reduced to mere detectability. The upshot: Google may be making us all stupid, as Nicholas Carr has it (the jury is still out on that one, I think), but Google is certainly making us all into wordsmiths of a very particular, very narrow sort, even if I can’t describe to you exactly how.

That rhetoric as it is traditionally understood should resist computation is deeply ironic, because data are nothing if not rhetorical. That is, data necessarily exist prior to argument, prior to any potentially persuasive gesture or operation. Data are the “given” upon which algorithms operate. (In saying so I’m admittedly appealing to etymology, that is, to philology.) Compare data and facts. Facts are facts, true by dint of being factual. Data, by contrast, aren’t truth-oriented; data can be good or bad, better or worse, incomplete and insufficient. Data are at some level always already mobilized, even if the ultimate function(s) of their mobilization remain uncertain. Here I am adding a corollary of sorts to “Algorithms: A provocation piece” (¶ 30-33), which asks, “Is there any such thing as algorithmic neutrality or impartiality?” and “How does one ascertain whether an algorithm is ‘good’ or ‘bad,’ ‘fair’ or unfair,’ ‘just’ or ‘unjust’?” It may be that these are important questions to ask about algorithms, but certainly they need to be asked about data first. What makes data “good,” both in operational terms and in ethical ones? The data used by Turnitin.com isn’t just “good” for Turnitin.com, it’s better and better, because every student essay added makes the database more valuable—because larger—in the hunt for repeated character strings. Students required to submit essays aren’t just made subject to Turnitin.com’s algorithm, that is, they actually enhance the algorithm that interpellates them by enlarging the encoded corpus of its operation. Of course we all do the same thing for Google, I should note, since we know that Google uses our personal search histories partly to enhance its algorithm.

So by all means let us consider the agencies and values implicit in algorithms according to their socio-technical performativity. For starters we’ll need to ask not about algorithms at all but rather about the data upon which they work. We need to ask about
the collection, the completeness, and commensurability of data, about the ownership of data, its accessibility, longevity, and protection. I really want to know about Bluffdale, Utah, where the NSA is building its mega data storage facility. No, not mega-, I mean its exa-, zetta-, yotta- data storage facility.

---


iv I’m drawing on Daniel Rosenberg’s chapter on the semantics of data in the edited collection “*Raw Data*” *Is an Oxymoron* (MIT 2013).