ALGORITHMIC ACCOUNTABILITY
Journalistic investigation of computational power structures

Nicholas Diakopoulos

Every day automated algorithms make decisions that can amplify the power of businesses and governments. Yet as algorithms come to regulate more aspects of our lives, the contours of their power can remain difficult to grasp. This paper studies the notion of algorithmic accountability reporting as a mechanism for elucidating and articulating the power structures, biases, and influences that computational artifacts exercise in society. A framework for algorithmic power based on autonomous decision-making is proffered and motivates specific questions about algorithmic influence. Five cases of algorithmic accountability reporting involving the use of reverse engineering methods in journalism are then studied and analyzed to provide insight into the method and its application in a journalism context. The applicability of transparency policies for algorithms is discussed alongside challenges to implementing algorithmic accountability as a broadly viable investigative method.

KEYWORDS accountability reporting; algorithms; computational journalism; data journalism; robot journalists; transparency

Introduction

We are now living in a world where algorithms, and the data that feed them, adjudicate a large array of decisions in our lives: not just search engines and personalized online news systems, but educational evaluations, the operation of markets and political campaigns, the design of urban public spaces, and even how social services like welfare and public safety are managed. But algorithms can arguably make mistakes and operate with biases. The opacity of technically complex algorithms operating at scale make them difficult to scrutinize, leading to a lack of clarity for the public in terms of how they exercise their power and influence.

Journalists are beginning to adapt their traditional watchdogging and accountability functions to this new wellspring of power in society. They are investigating algorithms in order to characterize their power and delineate their mistakes and biases in a process of what I call "algorithmic accountability reporting." Examples span everything from relatively straightforward comparisons and visualizations of statistical models of unemployment correction (Ingraham 2014), to sophisticated reverse engineering investigations of price discrimination online (Valentino-DeVries, Singer-Vine, and Soltani 2012).

Routledge
Taylor & Francis Group

Digital Journalism, 2015
Vol. 3, No. 3, 398–415, http://dx.doi.org/10.1080/21670811.2014.976411

In this paper, I study the broad question of how algorithms exert their power and are worthy of scrutiny by computational journalists. I explore approaches such as transparency and reverse engineering and how they may be useful for articulating algorithmic power. I analyze five case studies of journalistic investigations of algorithms and describe the challenges and opportunities they illustrate for doing algorithmic accountability work, including identifying newsworthy algorithms, sampling the input–output relationships of those algorithms to study correlations, and ultimately finding a story. This work contributes both (1) a theoretical lens positing various atomic algorithmic decisions which suggest a number of leading questions that can inform the investigation of algorithms and the development of transparency policies for algorithms, and (2) an initial assessment and analysis of how algorithmic accountability is being employed in practice, including its various limitations. I further discuss challenges to employing this mode of reporting, including human resources, legality, and ethics, and also look ahead to how journalists themselves may employ transparency in their own use of algorithms.

Related Work

Here I discuss relevant research in computational journalism, the study of the bias of algorithms, and attempts to investigate algorithms in the literature.

Computational journalism was initially conceived as the application of computing technology to enable journalism across information tasks such as information gathering, organization and sensemaking, storytelling, and dissemination (Diakopoulos 2010), as well as activities relating to data-driven investigations (Cohen, Hamilton, and Turner 2011). Computational journalism is often presented as tool-oriented (Lewis and Usher 2013), with literature describing the development of new tools and techniques in application areas like social media (Diakopoulos, De Choudhury, and Naaman 2012; Diakopoulos, Naaman, and Kivran-Swaine 2010; Schifferes et al. 2014; Stavelin 2013; Zubiaga, Ji, and Knight 2013), data visualization (Gao et al. 2014; Hullman, Diakopoulos, and Adar 2013), and audience understanding (Diakopoulos and Zubiaga 2014; Lee, Lewis, and Powers 2014). A parallel track of research on computational journalism has looked at the sociology of computing in the newsroom and how work practices are adapting (Anderson 2012; Karlsen and Stavelin 2013). In this paper, I proffer a new branch of research in computational journalism that inverts the typical tool-orientation and foregrounds journalism by making computation its *object*. Algorithmic accountability reporting thus seeks to articulate the power structures, biases, and influences that computational artifacts play in society.

Researchers have been aware of and have critiqued the biases embedded in computational systems for many years (Friedman and Nissenbaum 1996). More recent research has considered how values (Fleischmann and Wallace 2010) or even ideologies (Mager 2012) manifest themselves in computational models, as well as how such biases are observed in media systems (Bozdag 2013; Gillespie 2014). This paper explores and studies a diffusion and adaptation of such critiques of algorithms into journalism, including the use of reverse engineering techniques.

Reverse engineering has long been used in domains as diverse as archeology, architecture, forensics, and the military, not to mention software (Eilam 2005; Mancas 2013).

More recently there have been a number of studies in the computing literature that take a reverse engineering approach toward understanding "big data" systems and algorithms, such as in online reviews (Mukherjee et al. 2013), autocomplete systems (Baker and Potts 2013), online pricing systems (Mikians et al. 2012), search personalization (Hannak et al. 2013), online advertising (Guha, Cheng, and Franci 2010), and public health prediction (Lazer et al. 2014), which inform my understanding of the method and its limitations for journalistic investigations. This paper studies a number of examples of the technique in journalism and delineates the challenges and opportunities of employing the method as a broader strategy to enable algorithmic accountability reporting.

Algorithmic Power

An algorithm can be defined as a series of steps undertaken in order to solve a particular problem or accomplish a defined outcome. Here I consider algorithms that operate via digital computers due to their prevalence and ability to effect large numbers of people through scale.

There are myriad ways in which algorithms interact with and potentially problematize public life, including how they necessitate the datafication of the world, create complex feedback loops with social data, or encourage the creation of calculated publics (Gillespie 2014). Here I focus on the underlying and perhaps intrinsic crux of algorithmic power: autonomous decision-making. Algorithmic decisions can be based on heuristics and rules, or calculations over massive amounts of data. Rules may be articulated directly by programmers, or be dynamic and flexible based on machine learning of data. Sometimes a human operator maintains agency and makes the final decision in a process, but even in this case the algorithm biases the operator's attention toward a subset of information.

We can begin to assess algorithmic power by analyzing the atomic decisions that algorithms make, including *prioritization*, *classification*, *association*, and *filtering*. Sometimes these decisions are chained in order to form higher-level decisions and information transformations, such as summarization, which uses prioritization and then filtering operations to consolidate information while maintaining the interpretability of that information.

Prioritization

Prioritization serves to emphasize or bring attention to certain things at the expense of others, such as when a search engine prioritizes and ranks the most relevant search results. Among other uses, prioritization algorithms can enable more efficient management of many social services, such as fire-code inspections, policing and recidivism, immigration enforcement, or welfare management (Flowers 2013; Kalhan 2013; Perry and McInnis 2013), where assigning risk and orienting official attention can provide efficiency gains. But such risk ranking raises interesting questions: is that risk being assigned fairly and with freedom from malice, abuse, or discrimination?

Embedded in every prioritization algorithm are criteria that are computed and used to define a ranking through a sorting procedure. These criteria embed a set of choices and value-propositions, which may be political or otherwise biased, that determine what gets pushed to the top. Sometimes these criteria are not public, making it difficult to understand the weight of different factors contributing to the ranking.

Classification

Classification decisions involve categorizing a particular entity as a constituent of a given class by looking at any number of that entity's features. Google's Content ID is an example of an algorithm that makes classification decisions that feed into filtering decisions ("How Content ID Works" 2013). It scans all videos uploaded to YouTube and classifies them according to whether they have any copyrighted music playing during the video. If the algorithm classifies a video as an infringer it can automatically remove (i.e., filter) that video from the site as well as trigger other types of responses from the uploader or copyright holder.

In addition to the potential for uncertainty and mistakes, classification algorithms can also have biases. In a supervised machine-learning algorithm, training data is used to teach the algorithm how to separate classes. That training data is often gathered from people who inspect thousands of examples and tag each instance according to its category. The algorithm learns how to classify based on the definitions and criteria humans used to produce the training data, thus potentially introducing human bias into the classifier.

In general, there are two kinds of mistakes a classification algorithm can make—*false positives* and *false negatives*. For Content ID, a false positive is a video classified as "infringing" when it is actually "fair use." A false negative, is a video classified as "fair use" when it is in fact "infringing." Classification algorithms can be tuned to make fewer of either of those mistakes, but as false positives are tuned down, false negatives will increase, and vice versa. Tuned all the way toward false positives, the algorithm will mark a lot of fair use videos as infringing; tuned the other way it will miss a lot of infringing videos altogether. Tuning can privilege different stakeholders in a decision, implying an essential value judgment by the designer of such an algorithm in terms of how false positive and false negative errors are balanced (Kraemer, van Overveld, and Peterson 2011).

Association

Association decisions mark relationships between entities and draw their power through both semantics and connotative ability. For example, IBM's InfoSphere Identity Insight is a system that builds up context around people (the entities in this example) and then associates them. It is used by various governmental social service management agencies to reduce fraud and help make decisions about resource allocation. An IBM use-case for the system highlights the power of associative algorithms ("Outsmarting the Social Services Fraudster" 2013). The scenario involves the assessment of a potential foster parent, Johnson Smith. InfoSphere associates him, through a shared address and phone number, with his brother, a convicted felon. The report then renders judgment: "Based on this investigation, approving Johnson Smith as a foster parent is not recommended." In this scenario the social worker might deny a person the chance to be a foster parent because he or she was associated with a felon in the family via the algorithm.

Association algorithms are also built on criteria that define the association, including the similarity definition of how precisely two things must match to be considered associated with each other. When the similarity reaches a particular threshold value, the two things are said to have that association. Association decisions thus also suffer the same kinds of false positive and false negative mistakes as classification decisions.

Filtering

Filtering involves including or excluding information according to various rules or criteria. Inputs to filtering algorithms often take prioritizing, classification, or association decisions into account. In news personalization apps like Flipboard, news is filtered in and out according to how that news has been categorized, associated to the person's interests, and prioritized for that person.

Filtering decisions exert their power by either over-emphasizing or censoring certain information. The notion of a "filter bubble" is largely predicated on the idea that by only exposing people to information that they already agree with (by overemphasizing it) it amplifies biases and hampers people's development of diverse and healthy perspectives (Pariser 2011). Outright censorship is also an issue in some online information regimes such as the Chinese social network Weibo, which uses computer systems to constantly scan, read, and remove any objectionable content on the platform.

Algorithmic Accountability

Above I have articulated several ways that algorithms exert power though decisions they make in prioritizing, classifying, associating, and filtering information. Through these descriptions, it should also be clear that there are a number of human influences embedded into algorithms, such as criteria choices, training data, semantics, and interpretation. Algorithmic accountability must therefore consider algorithms as objects of human creation and take into account *intent*, including that of any group or institutional processes that may have influenced their design, as well as the *agency* of human actors in interpreting the output of algorithms in the course of making higher-level decisions. In the next section, I turn to an examination of transparency and how it may be useful for algorithmic accountability. Understanding the weaknesses of a transparency approach then motivates this paper's study of algorithmic accountability by journalists using reverse engineering methods.

Transparency

Transparency has recently gained purchase among journalists seeking to build public trust, becoming a new pillar of journalism ethics identified by McBride and

Rosenstiel (2013). Transparency can be a useful lever to bring to bear on algorithmic power when there is sufficient motive on the part of the algorithm's creator to disclose information and reduce information asymmetry.

Public relations concerns or competitive dynamics can incentivize the release of information to the public. Google, for instance, publishes a biannual transparency report showing how often it removes information or discloses it to governments. In other cases, the government imposes targeted transparency policies that compel disclosure (Fung, Graham, and Weil 2009). Such policies can improve public safety, the quality of services provided to the public, or have bearing on issues of discrimination or corruption that might persist if the information were not public. Transparency policies like restaurant inspection scores or automobile safety tests have been quite effective, for instance.

In other cases, algorithm operators may have ulterior goals which conflict with a desire for transparency. Oftentimes corporations limit how transparent they are, since exposing too many details (trade secrets) of their proprietary systems may undermine their competitive advantage, hurt their reputation and ability to do business, or leave the system open to gaming and manipulation. Trade secrets are a core impediment to understanding automated authority like algorithms since they, by definition, seek to hide information for competitive advantage (Pasquale 2011). Moreover, corporations are unlikely to be transparent about their systems if that information hurts their reputation or ability to do business. Finally, gaming and manipulation are issues that can undermine the efficacy of a system.

In the case of governmental use of algorithms, there is often a tension between transparency and national security. Despite policy such as the Federal Agency Data Mining Reporting Act of 2007 (42 USC § 2000ee–3. Federal Agency Data Mining Reporting 2007), which compels disclosure about a range of data-mining activities in the federal government, the leaked documents from Edward Snowden reveal that such policy is not effective when pitted against national security concerns. Furthermore, even in cases where national security is not an issue, the corporate concern for trade secrecy can bleed into the government's use of algorithms. Exemption 4 to the US federal Freedom of Information Act (FOIA) covers trade secrets and allows the federal government to deny requests for transparency concerning any third-party software integrated into its systems.

Transparency is far from a complete solution to balancing algorithmic power. When corporations or governments are not legally or otherwise incentivized to disclose information about their algorithms, we might consider a different, more adversarial approach employing reverse engineering.

A Study of Algorithmic Accountability Through Reverse Engineering

While transparency faces a number of challenges as an effective check on algorithmic power, an alternative approach is emerging based on the idea of reverse engineering. In this section, I first explain the method of reverse engineering, describe a study of the journalistic use of the technique, delineate five case studies where journalists have reverse engineered an algorithm, and synthesize the processes used by journalists in such investigations.

Reverse Engineering

Reverse engineering is the process of articulating the specifications of a system through a rigorous examination drawing on domain knowledge, observation, and deduction to unearth a model of how that system works. It is "the process of extracting the knowledge or design blueprints from anything man-made" (Eilam 2005, 3).

Some algorithmic power may be exerted intentionally, while other aspects might be incidental. The inadvertent variety benefits from reverse engineering's ability to characterize unintended side effects. Because the process focuses on the system's performance in-use, it can tease out consequences that might not be apparent even if you spoke directly to the designers of the algorithm. On the other hand, talking to a system's designers can also uncover useful information: design decisions, descriptions of the objectives, constraints, and business rules embedded in the system, major changes that have happened over time, as well as implementation details that might be relevant (Chikofsky 1990; Singh 2013). Thus, the journalistic adaptation of reverse engineering will naturally include reporting methods such as interviews or document reviews in conjunction with reverse engineering analysis.

Algorithms are often described as black boxes, their complexity and technical opacity hiding and obfuscating their inner workings. At the same time, algorithms must always have an input and output, two openings that can be manipulated to help shed light on the algorithm's functioning. It is not essential to understand the code of an algorithm to begin surmising something about how the algorithm operates in practice.

Study Methodology

The overall goal of this study was to understand the opportunities and limitations of a reverse engineering approach to investigating algorithms. In-depth interviews were conducted with four journalists who had worked on or edited stories involving the reverse engineering of algorithms in a news context, including Michael Keller (*The Daily Beast*), Scott Klein (ProPublica), Jeremy Singer-Vine (*Wall Street Journal*), and Rob Barry (*Wall Street Journal*). The author also engaged in participant observation and gained first-hand experience of developing a news story using reverse engineering. The five journalistic stories are presented in the next section as a series of case studies that inform the findings. Three additional investigative journalists (Sheila Coronel, Chase Davis, and Alyssa Katz) were then interviewed to better contextualize the findings in terms of investigative journalism. The interview protocol was semi-structured and covered questions regarding the genesis of the reverse engineering stories they were involved with, other stories where they had considered using the method, what the biggest challenges were, and how the method related to more traditional forms of investigative journalism. Interviews were analyzed qualitatively and in context with the journalistic output, including primary articles as well as any extant methodological articles explaining how the story was accomplished. The sample for the study is small as a result of the novelty of the technique and its narrow adoption, but even an analysis of these few cases begins to provide valuable insight into the challenges and opportunities of the method.

Case Studies in Reverse Engineering

Autocompletions on Google and Bing.. The Google autocomplete FAQ reads, “We exclude a narrow class of search queries related to pornography, violence, hate speech, and copyright infringement.” Bing makes similar claims about filtering spam and detecting adult or offensive content. Such editorial choices set the stage for broadly specifying censorship criteria. But what exactly are the boundaries of that censorship, and how do they differ among search engines?

To assess these questions, I gathered autosuggest results from hundreds of queries related to sex and violence in an effort to find those that were blocked (Diakopoulos 2013b). A list of 110 sex-related keywords was drawn from carefully crafted academic sources as well as the slang Urban Dictionary as inputs to the algorithm (Diakopoulos 2013a). I then looked to see which inputs resulted in zero output—suggesting a blocked word. While many of the most obvious words were outright blocked—like “ass” and “tits”—a number of the search terms were not.

In this case some transparency by the services through their FAQs and blogs suggested a hypothesis and tip as to what types of input the algorithm might be sensitive to (i.e., pornography and violence). Moreover, the algorithms themselves, both their inputs and outputs, were observable and accessible through application programming interfaces (APIs), which made it straightforward to collect a range of observations about the input–output relationship.

Autocorrections on the iPhone. Another example of surfacing editorial criteria in algorithms comes from Michael Keller, who at *The Daily Beast* dove into the iPhone spelling correction feature to see which words, like “abortion” or “rape,” the phone would not correct if they were typed incorrectly (Keller 2013).

Michael’s first attempt to sample this phenomenon was an API on the iPhone, which he used to identify words from a large dictionary that were not getting corrected, essentially pruning down the space of inputs to see what the algorithm “paid attention” to. He noticed that some of the words the API did not correct *were* getting corrected when they were typed directly on an iPhone. In order to mimic the real user experience he had to run an iPhone simulator on a number of computers, scripting it to act like a human typing in the word and then clicking the word to see if spelling corrections were presented.

Sometimes algorithms expose inputs and make it possible to record outputs, but those outputs are then further transformed and edited by downstream algorithms used to produce the user interface. What really matters to the end-user is the composition of these algorithms, not just the algorithm accessible via an API, but also how the user-interface algorithm interacts with that API to render the output that a user actually experiences. Understanding the context of how an algorithm’s output is transformed for human consumption is thus an important aspect to reporting on an algorithm’s consequences.

Targeting political emails. ProPublica’s Message Machine tried to reverse engineer how the Obama campaign in 2012 was using targeting information to adapt and personalize email messages for different recipients (Larson and Shaw 2012). In addition to collecting the emails from end-users, ProPublica asked participants to fill out a survey with basic demographic information, where they lived, and their campaign donation and volunteer history. These survey answers then served as a stand-in for the input to the algorithm they were trying to dissect. In this case, the output was observable—crowdsourced from thousands of people—but the types of inputs used by the targeting algorithm were hidden behind the campaign wall. Instead, ProPublica tried to determine what types of inputs the campaign's targeting algorithm was paying attention to based only on the outputs collected and a crowdsourced proxy for the inputs.

In one instance the analysis was wrong, as Scott Klein, an editor on the project explained to me: "We slipped and we said that 'in such and such an example they are targeting by age.'" After the campaign was over they found out that in fact the campaign was not targeting by age, but by donation history, a correlated variable. But correlation does not imply causation, nor intent on the part of the algorithm designer. Relying on correlation to make claims about what inputs an algorithm is using is thus error prone and demands additional reporting to help answer the question of "why?"

Price discrimination in online commerce. In 2012, the *Wall Street Journal* began probing e-commerce platforms to identify instances of potential price discrimination—the provision of different prices to different people (Valentino-DeVries, Singer-Vine, and Soltani 2012). By polling different websites it was able to spot several vendors, such as Staples, Home Depot, Rosetta Stone, and Orbitz, that were adjusting prices dynamically based on different factors like user geography, browser history, or mobile-browser use. In the case of Staples, it found that the input most strongly correlated to price was the distance to a rival's store, explaining about 90 percent of the pricing pattern.

To get the story the *Wall Street Journal* had to simulate visiting the various sites from different computers and browsers in different geographies (Singer-Vine, Valentino-DeVries, and Soltani 2012). Various archetype users and user profiles were built using cookies to see how those user profiles might impact the prices recorded. The journalists had to painstakingly construct those profiles to simulate inputs to the algorithm, and then looked to see if any of the variables in the profiles led to significant differences in output (prices).

Using reverse engineering on the scale of the Web surfaces several challenges, underscored both by the *Wall Street Journal* story and by academic efforts to reverse engineer personalization in Web search (Hannak et al. 2013). One of the issues is that sites like Staples might be using A/B testing to assess different tweaks to their interface. In other words, they are already running experiments on their sites, and to a reverse engineer it might look like noise, or just confusing irregularities. "While we try to experiment on algorithms, they are experimenting on us," observes Nick Seaver (2013, 6). Algorithms may be unstable and change over time, or have randomness built in to them, which makes understanding patterns in their input–output relationship much more challenging. Other tactics such as parallelization or analysis of temporal drift may be necessary in order to control for a highly dynamic algorithm.

Executive stock trading plans. Executives and corporate leaders sometimes use preset trading plans to avoid accusations of insider trading. The algorithmic plans can be triggered by any number of different parameters, like specific dates, stock prices, or announcements from competitors. The only catch is that the plans cannot be based on inside information. When an executive makes a trade, he or she files a form with the US Securities and Exchange Commission (SEC). The *Wall Street Journal* collected millions of these forms in an attempt to use reverse engineering to see if any of the plans were "opportunistic"—if they appeared to be taking advantage of market timing to increase profits (Pulliam and Rob 2012).

In this case, the output was observable since the prices of all trades were known. What the *Wall Street Journal* was interested in was reverse engineering how timing information was being used by different plans as an input. Essentially the *Wall Street Journal* had a sampled input–output relationship for each executive's plan specified by the documents filed with the SEC. However, what it did not know was any of the other inputs that could have also been feeding into these plans. Even though trade forms must be filed, the details of the plans themselves are hidden, leaving the reverse engineer to guess what inputs the algorithm was likely using. Perhaps competitor or sector prices are also inputs to some plans, requiring consideration of each variable in turn to assess whether there were correlations suggesting a connection. This case underscores the challenge with trying to understand *which* inputs an algorithm pays attention to. There is a huge space of potential inputs, some of which are observable and some of which are not.

Analysis of Cases: Toward a Theory and Methodology

I analyzed the cases presented above, together with the interviews of key informants and a review of algorithm reverse engineering in the literature (Baker and Potts 2013; Guha, Cheng, and Franci 2010; Lazer et al. 2014; Mikians et al. 2012; Mukherjee et al. 2013; Hannak et al. 2013; Sweeney 2013). I first present a model of different reverse engineering scenarios based on the visibility and accessibility of the algorithm. Then I identify processes and challenges to investigating algorithms which can be broadly characterized as identifying a newsworthy target, sampling the algorithm, and finding the story.

Theory. Figure 1 depicts two idealized black-box scenarios of interest to journalists reverse engineering algorithms by examining the input–output relationship. Each represents an extreme on a spectrum of observability of inputs and outputs to the algorithm. The first scenario, in Figure 1A, corresponds to an ability to fully observe all of an algorithm's inputs and outputs. This is the case for algorithms accessible via an online API. The cases of autocompletion, autocorrection, and price discrimination exemplify this scenario, though with varying degrees of difficulty in constructing and recording the inputs and outputs.

Figure 1B depicts a different scenario in which only the outputs of the algorithm are visible. The message machine case is a good example of this scenario in which the inputs are crowdsourced. This is a common case that data journalists encounter: a large

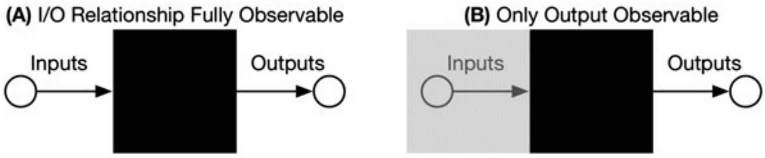
(A) I/O Relationship Fully Observable
Inputs
Outputs
(B) Only Output Observable
Inputs
Outputs

FIGURE 1
Two black-box scenarios with varying levels of observability

output dataset is available but there is limited information about how that data was transformed algorithmically. Interviews and document investigation can be especially important here and crowdsourcing is a way to cope with this issue by collecting data that might be used as inputs to the algorithm.

Sometimes inputs can be partially observable but not controllable; for instance, when an algorithm is driven off public data but it is unclear exactly what aspects of that data serve as inputs into the algorithm, such as in the cases of the executive stock trading plans or message machine. In general, the variable observability of the inputs and outputs is a limitation and challenge to the use of reverse engineering in practice. There are many algorithms that are used behind an organizational barrier that makes them difficult to prod. In such cases, partial observability (e.g., of inputs or outputs) through FOIA, document leaks, scraping, or crowdsourcing can still lead to some interesting results.

Identification. The interviews elucidated a number of questions relating to the newsworthiness of algorithm stories. "You need to ID algorithms that are very much non-hypothetical and direct in their impact," Alyssa Katz told me. Identifying algorithms to scrutinize thus involves asking questions like: What are the consequences and impact of that algorithm for the public, how significant are those consequences, and how many people might be affected by or perceive an effect by the algorithm? Does the algorithm have the potential for discrimination? Do errors from the algorithm create risks that negatively impact the public or individuals? While newsworthiness criteria for algorithms are still not well-defined, these are but a few examples of questions that might lead to newsworthy investigations of algorithms.

Sampling. A challenge in the process of reverse engineering is to choose how to sample the input–output relationship of the algorithm in some meaningful way. Sometimes the algorithm is out in the open and there are APIs that can be sampled, whereas other times inputs are obfuscated. Figuring out how to observe or simulate those inputs is a key part of a practical investigation involving reverse engineering. Reporting techniques and talking to sources can help uncover what inputs are being fed into an algorithm, but when trade secrets obscure the process we can be reduced to guessing, such as in the Message Machine or executive stock trading plans examples. Figuring out *what* the algorithm pays attention to as input becomes as intriguing a question as how the algorithm transforms input into output.

Given a vast potential sampling space, sampling decisions are often driven by hypotheses or potentially newsworthy outcomes. For the autocomplete story, the sampling strategy followed from a question of legality that might lead to a newsworthy story. If sex-related queries with "child" led to child pornography this would be a legal violation and a newsworthy story. There are, of course, tradeoffs between what you *can* sample and what you *would like* to sample in order to answer your question. Sampling an algorithm is not just about getting *any* valid sample either. The sample must simulate the reality of importance to your audience. This was a key difficulty for the autocorrection story, which eventually used a simulation of the iPhone with scripts that mimic how a human uses the phone. My experience analyzing autocompletions had a similar issue —the API results did not perfectly line up with what the user experiences. The Google API returns 20 results, but only shows 4 or 10 in the user interface (UI) depending on how preferences are set. Data returned from the API but that never appears in the UI is less significant since users will never encounter it in their daily usage.

In some cases, a dataset may be "found" in which someone else has already sampled an input–output relationship. Or you may not have any control of inputs because those inputs are actually individual people that you are unable or not ethically willing to simulate. Such datasets can still be useful, but the method is more powerful when the sampling strategy can be defined in a way to help directly answer the question at hand.

Finding the story. Once the input–output relationship of a black box is mapped out, the challenge becomes a data-driven expedition to find a news story. Has the algorithm made a bad decision or broken an expectation for how we think it should be operating? If there is a break with expectation, what is driving that—a bug, an incidental programming decision, or a deep-seated design intent? Expectations may be statistically based, or built on an understanding of social and legal norms. Looking at the false positives and false negatives can provide ideas about how and where the algorithm is failing, and lead to interesting stories.

In the price discrimination case, the first filter used for narrowing in on e-commerce sites was a statistical one: the variance of prices returned from a site for a given item across a variety of geographies. If any non-random variance was observed the site was marked for a more rigorous and in-depth analysis. Similarly, for the executive trading plans story, a sophisticated data-mining technique involving clustering and Monte Carlo simulation was used to find newsworthy cases and identify trading plans that fell outside of the norms of expectation.

In the autocompletions story, I used social and legal norms to help zero in on stories inside the collected data. Both Google and Bing had publicly expressed a desire to filter suggestions relating to pornography. Taking that a step further, child pornography is indeed a violation of the legal code, so searching for instances of that became a starting point for filtering the data I had collected. Knowing where the algorithm violates the designers' expectations (e.g., it lets through child pornography when the stated intent is not to do so), or where it may have unintended side effects can both make for interesting stories.

Reporting is still a key part of finding a story in a reverse engineering analysis. For every site that was flagged as a statistical hit, the price discrimination team did a much more comprehensive and custom analysis. Knowing what makes something a story is perhaps less about a filter for statistical, social, or legal deviance than it is about understanding the context of the phenomenon, including historical, cultural, and social expectations related to the issue—all things with which traditional reporting and investigation can help. Reaching out for interviews can still be valuable since information about the larger goals and objectives of the algorithms can help better situate a reverse engineering analysis.

Discussion

The study of algorithmic accountability reporting and reverse engineering described above exposes a number of challenges to incorporating the method into practice, including issues of human resources, legality, ethics, and the role that transparency might still effectively play.

Challenges in Human Resources, Legality, and Ethics

Developing the human resources to do algorithmic accountability reporting will take dedicated efforts to teach the computational thinking, programming, and technical skills needed to make sense of algorithmic decisions. The number of computational journalists with the technical skills to do a deep investigation of algorithms is still limited. Teaming computationally literate reporters with tech-savvy computer scientists in interdisciplinary "trading zones" (Lewis and Usher 2014) might be one method for doing more algorithmic accountability reporting. Another way would be to train journalists in more computational techniques. More applied experience with the technique is essential. "It's a lot of testing or trial and error, and it's hard to teach in any uniform way," noted Jeremy Singer-Vine in his interview.

More work is also needed to explore the legal ramifications of algorithmic accountability through reverse engineering by journalists. In the United States the Digital Millennium Copyright Act is one statute that poses issues, in addition to the anti-reverse engineering clauses that corporations typically add to their End User License Agreements (Eilam 2005). Even more severe is a law like the Computer Fraud and Abuse Act (18 USC § 1030. Fraud and Related Activity in Connection with Computers 2011). The need for qualified legal advice and potential for harsh sanctions for reverse engineering online sites suggests that non-professional journalists may find it more difficult to do algorithmic accountability investigations.

There are ethical questions that arise in the context of studying algorithms which also demand more research. In particular, we need to ask about the possible ramifications of publishing details of how certain algorithms work. Would publishing such information negatively affect any individuals? By publishing details of how an algorithm functions, specifically information about what inputs it pays attention to, how it uses various criteria in a ranking, or what criteria it uses to censor, how might that allow the algorithm to be manipulated or circumvented? And who stands to benefit or suffer disadvantage from that manipulation?

Transparency and the Journalistic Use of Algorithms

As previously noted, transparency as a mechanism for algorithmic accountability suffers from the issues of trade secrecy and manipulation. This creates a tension with journalism since, on the one hand, journalistic organizations are competitive corporations like any other, but on the other hand, have newly emerging ethical ideals promoting transparency (McBride and Rosenstiel 2013). As news organizations also come to employ algorithms in the shaping of the news they report, whether that be in finding new stories in massive datasets or presenting stories interactively, journalistic standards for transparency of algorithms will need to be developed. Well-trodden transparency policies in other domains do offer some opportunity to reflect on how such policy might be adapted for algorithms (Fung, Graham, and Weil 2009). For instance, targeted transparency policies might indicate the boundaries of disclosure (e.g., the factors or metrics of the algorithm), the frequency of their disclosure, and the user experience for communicating that information.

The case studies and their analysis above suggest several informational dimensions that might be disclosed in a standard transparency policy for algorithms, possibly for use by newsrooms themselves as well. These might include: the (1) the criteria used to prioritize, rank, emphasize, or editorialize things in the algorithm, including their definitions, operationalizations, and possibly even alternatives; (2) what data act as inputs to the algorithm—what it "pays attention" to, and what other parameters are used to initiate the algorithm; (3) the accuracy including the false positive and false negative rate of errors made in classification (with respect to some agreed-upon ground truth), including the rationale for how the balance point is set between those errors; (4) descriptions of training data and its potential bias, including the evolution and dynamics of the algorithm as it learns from data; and (5) the definitions, operationalizations, or thresholds used by similarity or classification algorithms. The legal and ethical perspectives alluded to above provide an overarching context for how these dimensions might variably be implemented or meet with resistance.

Another challenge to a transparency policy is to develop an effective user experience for transparency information. Ideally, the disclosed information needs to integrate into the decisions that the end-user would like to make based on such information. Some have argued for source code transparency in algorithms (O'Neil 2014), and while that may be helpful for specialists, it does not provide for an effective user experience for the public since they may not have adequate technical expertise to make meaningful choices based on such information. Furthermore, examining source code introduces a complication related to versioning: is the source code in operation the same that you are looking at and have access to, or could there be differences?

Journalistic innovation in algorithmic transparency is already emerging. Take, for instance, the *New York Times* 4th Down Bot, which is exemplary in its transparency (Burke and Quealy 2013). The bot uses a model built on data collected from NFL games going back to the year 2000. For every fourth down in a game, it uses that model to decide whether the coach should ideally "go for it," "punt," or "go for a field goal." How the bot sees the world is clearly articulated (Burke and Quealy 2013). It pays attention to the yard line on the fourth down as well as how many minutes are left in the game. Those are the inputs to the algorithm. It also defines two criteria that inform its predictions: expected points and win percentage. The model's limitations are clearly delineated—it cannot do overtime properly, for instance. And the bias of the bot is explained too: it is less conservative than the average NFL coach.

Two things that the bot could be more transparent about are its uncertainty—how *sure* it is in its recommendations—and its accuracy. Moreover, the information is "buried" in an article format, but might be made more salient to the end-user through innovations in information design (e.g., an information box that clearly answers the key questions above). Nonetheless, the 4th Down Bot already represents a fairly robust example of journalistic norms adapting to algorithmic technologies.

Conclusions

This paper has identified algorithmic power as something worthy of scrutiny by computational journalists interested in accountability reporting. I have offered a basis for understanding algorithmic power in terms of the types of decisions algorithms make in prioritizing, classifying, associating, and filtering information. Furthermore, I have presented five case studies, which contribute to delineating algorithmic accountability methods in practice, including challenges and considerations about the variable observability of input–output relationships as well as identifying, sampling, and finding newsworthy stories about algorithms. The case studies show that reverse engineering the input–output relationship of an algorithm can elucidate significant aspects of algorithms such as censorship. Finally, I have discussed challenges to the further application of algorithmic accountability reporting, and shown how transparency might be used to effectively adhere to journalistic norms in the use of newsroom algorithms.

ACKNOWLEDGEMENTS

A number of people have been generous with their time in providing critical feedback on this work, including the anonymous reviewers of this article. Scott Klein, Cathy O'Neil, and Cliff Stein deserve special praise for publicly critiquing the work on a panel at Columbia University. I am also grateful to those whom I interviewed about their experience (Rob Barry, Michael Keller, Scott Klein, and Jeremy Singer-Vine) and who helped me contextualize the findings (Sheila Coronel, Chase Davis, and Alyssa Katz). The Tow Center for Digital Journalism was also very supportive of the work.

REFERENCES

Anderson, C. W. 2012. "Towards a Sociology of Computational and Algorithmic Journalism." *New Media & Society*, December. doi:10.1177/1461444812465137. http://nms.sagepub.com/cgi/doi/10.1177/1461444812465137.

Baker, Paul, and Amanda Potts. 2013. "'Why Do White People Have Thin Lips?' Google and the Perpetuation of Stereotypes via Auto-complete Search Forms." *Critical Discourse Studies* 10 (2): 187–204.

Bozdag, Engin. 2013. "Bias in Algorithmic Filtering and Personalization." *Ethics and Information Technology* 15 (3): 209–227.

Burke, Brian, and Kevin Quealy. 2013. "How Coaches and the NYT 4th down Bot Compare." *New York Times*. http://www.nytimes.com/newsgraphics/2013/11/28/fourth-downs/post.html

Chikofsky, Elliot. 1990. "Reverse Engineering and Design Recovery: A Taxonomy." *IEEE Software* 7 (1): 13–17.

Cohen, Sarah, James T. Hamilton, and Fred Turner. 2011. "Computational Journalism." *Communications of the ACM* 54 (10): 66–71.

Diakopoulos, Nicholas. 2010. "A Functional Roadmap for Innovation in Computational Journalism." http://www.nickdiakopoulos.com/wp-content/uploads/2007/05/CJ_Whitepaper_Diakopoulos.pdf.

Diakopoulos, Nicholas. 2013a. "Sex, Violence, and Autocomplete Algorithms: Methods and Context." http://www.nickdiakopoulos.com/2013/08/01/sex-violence-and-autocomplete-algorithms-methods-and-context/.

Diakopoulos, Nicholas. 2013b. "Sex, Violence, and Autocomplete Algorithms." *Slate*, August. http://www.slate.com/articles/technology/future_tense/2013/08/words_banned_from_bing_and_google_s_autocomplete_algorithms.html

Diakopoulos, Nicholas, and Arkaitz Zubiaga. 2014. "Newsworthiness and Network Gatekeeping on Twitter: The Role of Social Deviance." *International Conference on Weblogs and Social Media (ICWSM)*, Ann Arbor, MI.

Diakopoulos, Nicholas, Mor Naaman, and Funda Kivran-Swaine. 2010. "Diamonds in the Rough: Social Media Visual Analytics for Journalistic Inquiry." In *Proceedings of the Symposium on Visual Analytics Science and Technology (VAST)*, 115–122. Salt Lake City, UT. http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5652922&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D5652922.

Diakopoulos, Nicholas, Munmun De Choudhury, and Mor Naaman. 2012. "Finding and Assessing Social Media Information Sources in the Context of Journalism." *Conference on Human Factors in Computing Systems (CHI)*, Austin, TX.

Eilam, Eldad. 2005. *Reversing: Secrets of Reverse Engineering*. Indianapolis, IN: Wiley.

Fleischmann, Kenneth R., and William A. Wallace. 2010. "Value Conflicts in Computational Modeling." *Computer* 43 (7): 57–63.

Flowers, Michael. 2013. "Beyond Open Data: The Data-driven City." In *Beyond Transparency: Open Data and the Future of Civic Innovation*, edited by Brett Goldstein. Code for America Press. http://beyondtransparency.org/chapters/part-4/beyond-open-data-the-data-driven-city/.

Friedman, Batya, and Helen Nissenbaum. 1996. "Bias in Computer Systems." *ACM Transactions on Information Systems* 14 (3): 330–347.

Fung, Archon, Mary Graham, and David Weil. 2009. *Full Disclosure: The Perils and Promise of Transparency*. New York: Cambridge University Press.

Gao Tong, Jessica Hullman, Eytan Adar, Brent Hecht, and Nicholas Diakopoulos. 2014. "NewsViews: An Automated Pipeline for Creating Custom Geovisualizations for News." *Proceeding of Conference on Human Factors in Computing Systems (CHI)*, Toronto.

Gillespie, Tarleton. 2014. "The Relevance of Algorithms." In *Media Technologies: Essays on Communication, Materiality, and Society*, edited by Tarleton Gillespie, Pablo Boczkowski, and Kirsten Foot. Cambridge, MA: MIT Press.

Guha, Saikat, Bin Cheng, and Paul Franci. 2010. "Challenges in Measuring Online Advertising Systems." Internet Measurement Conference (IMC), Melbourne.

Hannak, Aniko, Piotr Sapiezynski, Arash Molavi Kakhki, Balachander Krishnamurthy, David Lazer, Alan Mislove, and Christo Wilson. 2013. "Measuring Personalization of Web Search." Proceeding of World Wide Web Conference (WWW), Rio de Janeiro.

"How Content ID Works." 2013. https://support.google.com/youtube/answer/2797370?hl=en.

Hullman, Jessica, Nicholas Diakopoulos, and Eytan Adar. 2013. "Contextifier: Automatic Generation of Annotated Stock Visualizations." *Conference on Human Factors in Computing Systems (CHI)*, Paris.

Ingraham, Christopher. 2014. "Jobs Preview: Pay Less Attention to the Sausage, and More to How It's Made." *Washington Post*, April. http://www.washingtonpost.com/blogs/wonkblog/wp/2014/04/03/jobs-preview-pay-less-attention-to-the-sausage-and-more-to-how-its-made/.

Kalhan, Anil. 2013. "Immigration Policing and Federalism through the Lens of Technology, Surveillance, and Privacy." *Ohio State Law Journal* 74: 1105–1165.

Karlsen, Joakim, and Eirik Stavelin. 2013. "Computational Journalism in Norwegian Newsrooms." *Journalism Practice* 8 (1): 34–48.

Keller, Michael. 2013. "The Apple 'Kill List': What Your IPhone Doesn't Want You to Type." *The Daily Beast*, July.

Kraemer, Felicitas, Kees van Overveld, and Martin Peterson. 2011. "Is There an Ethics of Algorithms?" *Ethics and Information Technology* 13 (3): 251–260.

Larson, Jeff, and Al Shaw. 2012. "Message Machine: Reverse Engineering the 2012 Campaign." *ProPublica*, July.

Lazer, David, Ryan Kennedy, Gary King, and Alessandro Vespignani. 2014. "The Parable of Google Flu: Traps in Big Data Analysis." *Science* 343 (6176): 1203–1205.

Lee, Angela, Seth Lewis, and Matthew Powers. 2014. "Audience Clicks and News Placement: A Study of Time-lagged Influence in Online Journalism." *Communication Research* 41 (4): 505–530.

Lewis, Seth C, and Nikki Usher. 2013. "Open Source and Journalism: Toward New Frameworks for Imagining News Innovation." *Media, Culture and Society* 35 (5): 602–619.

Lewis, Seth C, and Nikki Usher. 2014. "Code, Collaboration, and the Future of Journalism: A Case Study of the Hacks/Hackers Global Network." *Digital Journalism* 2 (3): 383–393. doi:10.1080/21670811.2014.895504.

Mager, Astrid. 2012. "Algorithmic Ideology: How Capitalist Society Shapes Search Engines." *Information, Communication & Society* 15 (5): 769–787.

Mancas, Christian. 2013. "Should Reverse Engineering Remain a Computer Science Cinderella?" *Information Technology & Software Engineering*. http://omicsgroup.org/journals/should-reverse-engineering-remain-a-computer-science-cinderella-2165-7866.S5-e001.php?aid=12682.

McBride, Kelly, and Tom Rosenstiel, eds. 2013. *The New Ethics of Journalism*. Thousand Oaks, CA: CQ Press.

Mikians, Jakub, László Gyarmati, Vijay Erramilli, and Nikolaos Laoutaris. 2012. "Detecting Price and Search Discrimination on the Internet." *Workshop on Hot Topics in Networks*: 79–84. http://dl.acm.org/citation.cfm?id=2390245.

Mukherjee, Arjun, Vivek Venkataraman, Bing Liu, and Natalie Glance. 2013. "What Yelp Fake Review Filter Might Be Doing?" *International Conference on Weblogs and Social Media (ICWSM)*, Boston, MA.

O'Neil, Cathy. 2014. "An Attempt to FOIL Request the Source Code of the Value-added Model." http://mathbabe.org/2014/03/07/an-attempt-to-foil-request-the-source-code-of-the-value-added-model/.

"Outsmarting the Social Services Fraudster." 2013. IBM White Paper.

Pariser, Eli. 2011. *The Filter Bubble: How the New Personalized Web is Changing What We Read and How We Think*. New York: Penguin Press.

Pasquale, Frank. 2011. "Restoring Transparency to Automated Authority." *Journal on Telecommunications & High Technology Law* 9: 235–256.

Perry, Walter, and Brian McInnis. 2013. *Predictive Policing: The Role of Crime Forecasting in Law Enforcement Operations*. Santa Monica, CA: RAND.

Pulliam, Susan, and Barry Rob. 2012. "Executives' Good Luck in Trading Own Stock." *Wall Street Journal*, November. http://online.wsj.com/news/articles/SB10000872396390444100404577641463717344l7.

Schifferes, Steve, Nic Newman, Neil Thurman, David Corney, Ayse Göker, and Carlos Martin. 2014. "Identifying and Verifying News through Social Media: Developing a User-Centred Tool for Professional Journalists." *Digital Journalism* 2 (3): 406–418.

Seaver, Nick. 2013. "Knowing Algorithms." *Media in Transition* 8: 1–12.

Singer-Vine, Jeremy, Jennifer Valentino-DeVries, and Ashkan Soltani. 2012. "How the Journal Tested Prices and Deals Online." *Wall Street Journal*. http://blogs.wsj.com/digits/2012/12/23/how-the-journal-tested-prices-and-deals-online/.

Singh, Ramandeep. 2013. "A Review of Reverse Engineering Theories and Tools." *International Journal of Engineering Science Invention* 2 (1): 35–38.

Stavelin, Eirik. 2013. "The Pursuit of Newsworthiness on Twitter." *Norsk Informatikkonferance (NIK)*: 1–12.

Sweeney, Latanya. 2013. "Discrimination in Online Ad Delivery." *Communications of the ACM (CACM)* 56 (5): 44–54.

18 USC § 1030. Fraud and Related Activity in Connection with Computers. 2011. http://www.law.cornell.edu/uscode/text/18/1030.

42 USC § 2000ee–3. Federal Agency Data Mining Reporting. 2007. http://www.law.cornell.edu/uscode/text/42/2000ee-3.

Valentino-DeVries, Jennifer, Jeremy Singer-Vine, and Ashkan Soltani. 2012. "Websites Vary Prices, Deals Based on Users' Information." *Wall Street Journal*. http://online.wsj.com/news/articles/SB10001424127887323777204578189391813881534.

Zubiaga, Arkaitz, Heng Ji, and Kevin Knight. 2013. "Curating and Contextualizing Twitter Stories to Assist with Social Newsgathering." *International Conference on Intelligent User Interfaces (IUI)*, Santa Monica, CA.

Nicholas Diakopoulos, College of Journalism, University of Maryland, USA. E-mail: nad@umd.edu. Web: http://www.nickdiakopoulos.com