5-10-2023

When AI Moves Downstream

Frances S. Grodzinsky
Sacred Heart University

Keith W. Miller
University of Missouri-St. Louis

Marty J. Wolf
Bemidji State University

Recommended Citation

Grodzinky, F. S., Miller, K. W., & Wolf, M. J. (2023). When AI moves downstream. *International Conference on Computer Ethics, 1*(1), 1-9.

WHEN AI MOVES DOWNSTREAM

Frances S. Grodzinsky
School of Computer Science and Engineering,
Sacred Heart University, Fairfield, Connecticut, USA,

Keith W. Miller
College of Education and the Department of Computer Science,
University of Missouri, St. Louis, Missouri, USA

Marty J. Wolf
Department of Mathematics and Computer Science,
Bemidji State University, Bemidji, Minnesota, USA

KEYWORDS: AI, ethics, downstream uses, responsibility attribution. large language models

ABSTRACT

After computing professionals design, develop, and deploy software, what is their responsibility for subsequent uses of that software "downstream" by others? Furthermore, does it matter ethically if the software in question is considered to be artificial intelligent (AI)? The authors have previously developed a model to explore downstream accountability, called the Software Responsibility Attribution System (SRAS). In this paper, we explore three recent publications relevant to downstream accountability, and focus particularly on examples of AI software. Based on our understanding of the three papers, we suggest refinements of SRAS.

INTRODUCTION

In "On the Responsibility for Uses of Downstream Software" (2019) Wolf et al. explored the degree to, and ways in which, computing professionals are responsible for the downstream use of the software they develop. This analysis is based on the nature of the software itself, not on the nature of the downstream use. "Downstream use" refers to how a piece of software is used by others after its release.

The authors adapted a mechanism developed by Floridi (2016). In this work, Floridi shifted the question of responsibility away from the intentions of developers per se and onto the impact that their Distributed Moral Actions have on moral patients. Wolf et al. take this in a slightly different direction and make an argument that there are features of software that can be used as guides to better distinguish situations where a software developer might share in responsibility for the software's downstream use, from those in which the software developer likely does not share in that responsibility. The features of that Software Responsibility Attribution System (SRAS)—as we will call it here—that are significant include: closeness to the hardware, risk, sensitivity of data, degree of control over or knowledge of the future population of users, and the nature of the software (general vs. special purpose). A subsequent paper, Grodzinsky et al. (2020), offers some evidence that these features and their impact on responsibility assessment are consistent with some sources in the literature.

Since that time, artificial intelligence (AI), understood broadly, has been increasingly deployed in many different application areas. In this paper, we will re-examine the SRAS using critical work on AI. That analysis will lead to adjustments to the SRAS in the context of AI applications.

THE SRAS

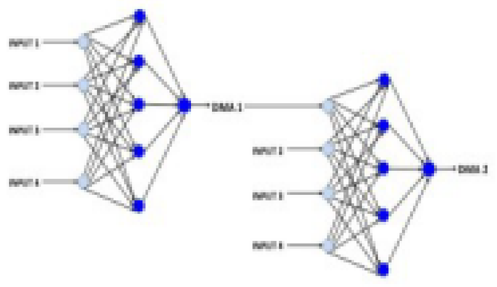
The SRAS is built upon the assumption that not all software gives software developers the same sort of control over how others might use their software after its release. This assumption resulted in SRAS being a two-model system of responsibility: the Fixed History Model (FHM) and the Chained History Model (CHM). (Figure 1)

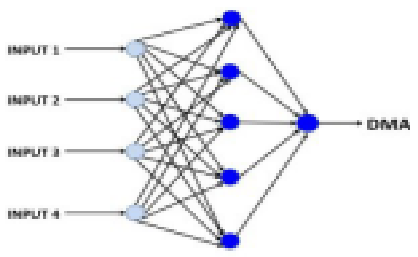
When it is appropriate to apply, the Fixed History Model illustrates that a developer has limited ethical responsibility for downstream uses. For example, a software developer who produces a compiler for an established programming language has little responsibility for programs that happen to use that compiler. (In applying SRAS, we assume that the software, the compiler in this example, is well constructed and does a reasonably good job of producing its intended effect, efficient, error-free machine code in the compiler's case.)

When the FHM is not appropriate (based on the features identified in SRAS), the CHM would be used to emphasize that the software developer has at least some responsibility for the ethical impact—both good and bad—that the software in question produces downstream. In this case, responsibility *does* reach back when the software is used in a subsequent product. For example, imagine software that takes as input a known set of DNA samples and a relatively tiny (and perhaps mixed) sample of unknown origin, and outputs information on the likelihood that the tiny unknown sample includes DNA that "matches" one of more of the known sample inputs. The sensitivity of the data involved, and the special purpose of this matching software compels us to emphasize the responsibility of the developers for any downstream use.

Next, we consider AI software against the six software features of SRAS to give a cursory analysis of whether AI developer responsibility for its consequences ought to be evaluated according to one model or the other. For this analysis, it is important to distinguish at least two roles: the developer who designs and implements the AI and the developer who trains the AI to perform a particular function. For some applications, these two roles are inseparable. In situations where they are not, we are referring to the first type of developer.

- Closeness to the hardware: AI is not used to manage hardware resources efficiently, so this feature does not come to bear on AI.
- Risk: AI is inherently risky in that it changes as it acquires new information. This feature suggests use of the CHM,
- Sensitivity of data: As the developer has control over the choice of data used to train the AI and how to scrub this data, this feature suggests the CHM.
- Degree of control over or knowledge of the future population of users: As there is typically very little control over who might acquire an AI (other than perhaps price), the SRAS suggests the use of the CHM.
- The nature of the software (general vs. special purpose): The nature of the AI we are considering here suggests the use of the FHM due to its general-purpose nature.





Fixed History

Model

Chained History Model

Figure 1: The SRAS models

In the remainder of the paper, we consider whether there is more than meets the eye with respect to features of AI that come to bear on the assignment of ethical responsibility to its developers.

“MORAL ENTANGLEMENTS” AND “THE RESPONSIBILITY GAP”

An important characteristic of the SRAS is its reliance on characteristics of the software to help determine if the fixed or chained model is more appropriate. While the binary simplicity of SRAS (pick one of two) is appealing, it may also be simplistic for some situations. We do not want to abandon the SRAS, but we do want to expand it, integrating ideas from recently published papers that describe issues of responsibility for the consequences of software development. One such paper is "Mind the Gap: Autonomous Systems, the Responsibility Gap, and Moral Entanglement" (Goetze 2022). The "responsibility gap" Goetze identifies is exactly the problem that the SRAS was designed to bridge, a philosophical "distance" between initial work on software and its eventual deployment, and how responsibility and accountability can extend across that gap. The SRAS bridges the gap by insisting on a traceable link from software authors to their artifact used elsewhere when the chained model is appropriate. Goetze takes a different approach, characterized by two phrases: "vicarious responsibility" and "moral entanglement."

Goetze suggests that declaring a programmer to be "responsible" or "not responsible" when their software is used downstream may be imprecise. Instead, Goetze encourages us to think of the relationship between a software developer and the software as something analogous to parents and their children, or to citizens and their county. When a developer's software downstream causes harm, the developer can feel shame and accountable for that harm, without being directly blameworthy, both in a legal or ethical sense.

As we integrate this idea into the SRAS, we need to emphasize that when the CHM is judged to be appropriate, the responsibility may be more than vicarious, and the accountability of the developer more than mere moral entanglement. When someone writes software explicitly designed to remotely take control of a heart pacemaker, without proper authorization, and someone else uses that software in a new program that is used to assassinate people after hijacking the pacemaker, the original software developer bears direct, moral culpability for both the original software, and the ensuing harms. However, we also see the possibility of someone working on open source machine learning software that someone else uses in a clearly harmful way. Depending on the details of the original software and the subsequent use, we can envision a designation that includes the CHM, but also labels the original developer's accountability to be vicarious responsibility. The developer may have regrets and shame at being morally entangled with the new software and its harms, but we can also envision not declaring that the original developer was directly accountable for the subsequent harms.

Goetze discusses examples including autonomous systems that learn. We are reluctant to make a blanket claim that whenever learning systems are deployed, the developers can convincingly claim that their responsibility extends only to vicarious responsibility. We insist that developers have a duty to carefully anticipate problems with such systems before releasing them into society, and to mitigate the risks that such systems might have. If the developers do not accomplish due diligence to reduce those risks, we think their moral accountability is more direct than the phrase "vicarious responsibility" implies. However, we also recognize the possibility of cases in which vicarious responsibility is the appropriate designation.

THE INFLUENCE OF FREE SOFTWARE ON RESPONSIBILITY

Widder et al. (2022) studied the ways that developers of an AI-enabled open source deepfake project reasoned about the ethics of their work. This study identified a connection between Free Software (FS) values and the developers' attitudes about the ethical uses of deepfake software. This connection raises a question as to whether the SRAS feature list ought to include the type of license applied to the software, especially when that software includes features identified as AI. Our initial position is that the SRAS characteristic "Degree of Control Over or Knowledge of the Future Users" implicitly includes licensing considerations.

Widder et al. (2022) identified several responsibility related concerns that were reported by developers working on a particular open source project that was licensed under the GNU General Public License (GPL). Developers noted the decentralized control, the legal requirements of the GPL, and the inevitability of technology as reasons that they might not bear much responsibility for the downstream uses of the software that they are developing.
Different open source projects have different models for managing projects. Some use a distributed model for decision making, others use a more centralized model where major decisions rest in the hands of a few project leaders. Furthermore, the ability to fork a project keeps pressure on decision makers to be mindful of the desires of the developers working on a project. While actually forking a project is rather straightforward from a code base perspective, the social cost that comes with potentially splitting the developer base is high. Yet, Widder et al. found at least a suggestion that some leaders of the project that they studied saw this "decentralized control" as a reason for not including features that might prevent, or at least make more difficult,
some of the harms that might be the result of downstream use.

The argument developers made that stem from the legal requirements of the GPL was captured by Widder et al.: "…the open source status of their project (a choice they made) prohibits them from controlling downstream uses" (2022 p 2037). Once a project incorporates any GPL software, there is a legal obligation to license the new software with the GPL. Widder et al. point out that the GPL becomes norm setting in that there is a perception that any changes to the licensing arrangement would kill the project (2022 p 2037).

Widder et al. also found that individual developers did not see their role in the project as particularly significant. "We see that our participants view their own role in developing Deepfake software as insignificant in the context of the wider progress of mutually interchangeable alternatives" (2022 p 2038). These developers would have us imagine that if they didn't work on this project someone else would, or that some other, similar project would take its place. Thus, they bear no responsibility for the downstream use of the software—an argument that the FHM is better suited for all open source software.

The final argument against assigning open source developers' responsibility for the downstream uses of their software that Widder et al. identified in their work is again more general. That is the notion that software is merely a tool and "and that the ethics of any particular use case is solely up to the user" (2022 p 2038). This argument can be taken more generally to mean that no developer of any technology is responsible for its use—essentially arguing that using something like the CHM for determining responsibility is never in play.

Thus, there is at least the suggestion that some open source developers working on AI projects see either general arguments about technology or about the GPL and the resulting software development model as ways to absolve developers of responsibility for the downstream use of their software. Most of these arguments are upended by other developers that Widder et al. interviewed. They identified strategies taken by GitHub (a large software development hosting platform), individual developers, and project leaders, as well as attitudes held by some developers. These observations all point to the fact that at least some developers working on open source projects not only have a sense of responsibility for the downstream uses of the software they develop but are also actively engaged in activities that make harmful uses of the software less socially acceptable and more challenging to achieve. Widder et al. also suggest six strategies for open source developers and others to engage in to take better care of their downstream responsibilities.

With respect to the proposal that the software development model (open source/proprietary) ought to be added to the list of features included in SRAS, we offer this analysis. Under the proprietary model of software development there are clearly more tools that a developer can use to control who gets to use the software, tools not available to open source developers. Setting the price high or creating enforceable contract terms regarding the allowable uses of the software are in play. When these control mechanisms are used there is an argument that the developer ought to be less responsible for the downstream use—one ought to tend toward the FHM in
responsibility analysis. Certainly, when a developer does not engage in practices that attempt to limit downstream harm, that failure to act becomes an ethical failure. SRAS is used to help analyze situations where there are no clear ethical failures.

Open source developers have less control over downstream users. This is especially clear for developers whose code is licensed under the GPL. This lack of control increases their ethical responsibility even in the absence of ethical failures.

While the software development model does have an impact on responsibility assignment, our analysis here suggests that impact does not have aspects that cause a different analysis than one gains from considering the SRAS characteristic Degree of Control Over or Knowledge of the Future Users. Our decision is against explicitly including the software development model in the features used in SRAS.

LARGE LANGUAGE MODELS

Attribution of responsibility comes to light in the article by Bender et al. (2021), that looks at the ethics surrounding the use of large language models (LLMs) in natural language processing (NLP). They enumerate risks and harms that stem from LLMs. There are at least three phases of NLP that they consider: the creation of large training sets, the training of the AI models, and the application of the models. The features of SRAS are mostly present in the harms identified by Bender et al. However, the complexity of these systems suggests that Sensitivity of Data feature in the SRAS needs to be revisited. In the case of LLMs, no one piece of data may be particularly sensitive, but information carried by the entire collection of data can lead to harm (Hand, 2018).

Bender et al. investigate how language models change with size. As the number of parameters and the size of the training data sets increase, the potential risks increase. The authors consider environmental risks, financial risks, and risks associated with training sets; they also discuss ways to mitigate these risks. In 2021, Bender et al. explored the issue of "How big is too big"? (Bender 2021: 610). In this case study, the developer who designs and implements the AI and the developer who trains the AI to perform a particular function are usually one in the same. AI developer responsibility for the consequences of the use of data can best be analyzed by the CHM.

It could be argued that GPT-n is a general-purpose black box application where there is very little, if any, transparency, and no collection of private data; therefore, the FHM would work. However, Bender et al.'s article demonstrates that there are problems of data sensitivity *other than privacy* that could be harmful to marginalized communities. The CHM allows us to focus more on transparency of the GPT algorithms at the development stage to better understand responsibility and accountability for the data sets selected and

the potential biases that may harm users downstream. For example, the developers of GPT-2, created their training sets from internet data scraped from users in the United States who are primarily men and between the ages of 18-29 (Bender et al., 613). The training set for GPT-3 filtered a version of Common Crawl used to develop GPT-2. These were deliberate developer choices. In addition, once the filtering parameters are set, it is hard to know exactly what else is filtered. In accepting web text as representative, developers perpetuate dominant viewpoints and exclude the language of marginalized communities (Bender et al., 614). This reflects a worse scenario than the one suggested by the old adage, "Garbage In, Garbage Out." With AI it is "Mundane In, Downstream Damage Out."
The CHM places developers in the chain of responsibility for the data sets that drive GPT. Developers should be aware of the harm that uncurated and unanalyzed large data sets gathered from web crawlers like Common Crawl might cause. These training sets will not guarantee diversity and might encode unwanted, or worse, harmful, biases. Bender et al. advocate for the need for developers to curate and document language model training data, consciously and carefully deciding what text to put into the training sets. Abdicating responsibility to web crawlers and opting for scale at the cost of misinformation and perpetuated biases and stereotypes would cause harm to users downstream and quite likely people who have been already marginalized. We agree with their advocacy for a more transparent, justice-oriented method of data collection for GPT-4 and similar applications that use training sets.

CONCLUSIONS

In its original form, the SRAS model for attributing responsibility for downstream use had the elegance of simplicity, but that simplicity could mask some complications that can be important, especially for some particular types of software. Those complications seem to be captured in this recent Twitter post by OpenAI's CEO Sam Altman (2023):

> we had a significant issue in ChatGPT due to a bug in an open source library, for which a fix has now been released and we have just finished validating.
>
> a small percentage of users were able to see the titles of other users' conversation history. we feel
>
> awful about this.

We see the tension of Goetze's "vicarious responsibility" and "moral entanglement." Altman feels "awful" but not responsible. Altman seems to suggest that the responsibility for the harm here does not lie with OpenAI, a downstream user of open source software, but with the open source developers. His observation reflects some of the tension among developers identified by Widder et al. (2022) surrounding responsibility in open source projects. Finally, as an LLM, ChatGPT carries with it the moral concerns raised by Bender et al. (2021). It is clear from the quote and each article that work still needs to be done on how to consistently attribute downstream responsibility, how the SRAS can help in that attribution, and how to improve it.

From Goetze, we apply the ideas of "vicarious responsibility" and "moral entanglement." Although these terms should not be allowed to serve as a blanket abandonment of responsibility for developers, the ideas can remind us that the ties that bind us to accountability for downstream use can require nuanced analysis. Even though developers might be justified in having regrets about what has been done downstream with software they helped produce, that does not automatically mean that they have a direct responsibility for the harm. SRAS gives us several characteristics to help determine the strength of the link from downstream to upstream; Goetze's work suggests ways to consider the nature of the link.

Widder et al. (2022) and Gualdi et al. (2021) both explore the interaction of open source protocols and

licenses, and how they might affect our judgement of responsibility for downstream uses. Indeed, some advocates of the GPL make explicit claims that they are obligated to *not* consider the ethical import of subsequent uses. We contend that this abdication of responsibility is not ethical acceptable. While the GPL license may discourage any legal link between the original developers and subsequent developers, we maintain that a strong ethical link still exists, and should be acknowledged. Our judgement is that the SRAS need not be changed in order to better accommodate open source, GPL licensed software,
Bender et al. (2021) focus on the importance of the data used to train AI models, and the ethical responsibilities associated with the collection and use of that data, as well as the dissemination of the resulting systems. We enthusiastically agree with Bender et al. that there are significant ethical risks and responsibilities associated with data sets used to train AI. We think the SRAS can be used in a directly analogous fashion for data as it is used for programming. An insight of Bender et al. is that the selection and use of data when fed to AI models is a sort of programming, even though specific commands are not applied.

We anticipate continued research and discussions about the ethical responsibilities for downstream use of software and data. We support ways to better understand and track these responsibilities in order to increase accountability and transparency. We oppose any notion that sophisticated systems somehow automatically insulate software developers from their responsibilities; they do not.

REFERENCES:

Altman, Sam. (2023) https://twitter.com/sama/status/1638635717462200320

Bender, Emily M., Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. (2021). On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?. In Conference on Fairness, Accountability, and Transparency (FAccT '21), March 3–10, 2021, Virtual Event, Canada. ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3442188.3445922

Floridi, L. (2016) Faultless responsibility: on the nature and allocation of moral responsibility for distributed moral actions. Philosophical Transactions. http://rsta.royalsocietypublishing.org/

Goetze, Trystan S. (2022). Mind the Gap: Autonomous Systems, the Responsibility Gap, and Moral Entanglement. In 2022 ACM Conference on Fairness, Accountability, and Transparency (FAccT '22). Association for Computing Machinery, New York, NY, USA, 390–400. https://doi.org/10.1145/3531146.3533106

Grandinetti, Justin. (2021). "Examining embedded apparatuses of AI in Facebook and TikTok." Ai & Society (2021): 1-14.

Grodzinsky, F. S., Wolf, M. J., Miller, K. (2020) "On Using Models for Downstream Responsibility." Paradigm Shifts in ICT Ethics Proceedings of the ETHICOMP 2020. Ed: Jorge Pelegrín Borondo, Mario Arias Oliva, Kiyoshi Murata. Spain: Universidad de La Rioja. Pp.364- 365.

Gualdi, Francesco, and Antonio Cordella.(2021). "Artificial intelligence and decision-making: The question of accountability." In Proceedings of the 54th Hawaii International Conference on System Sciences, p. 2297.

Hand, David J. "Aspects of data ethics in a changing world: Where are we now?" *Big Data* 6, no. 3 (2018): 176-190.

Kang, Hyunjin, and Chen Lou.(2022). "AI agency vs. human agency: understanding human–AI interactions on TikTok and their implications for user engagement." Journal of Computer Mediated Communication 27, no. 5 (2022): zmac014.

Prince, Anya, Reproductive Health Surveillance (July 29, 2022). Boston College Law Review, Forthcoming, U Iowa Legal Studies Research Paper No. 2022-36, Available at SSRN: https://ssrn.com/abstract=4176557

Widder, David Gray, Dawn Nafus, Laura Dabbish, and James Herbsleb. (2022). Limits and Possibilities for "Ethical AI" in Open Source: A Study of Deepfakes. In 2022 ACM Conference on Fairness, Accountability, and Transparency (FAccT '22), June 21–24, 2022, Seoul, Republic of Korea. ACM, New York, NY, USA 2035-2046. https://doi.org/10.1145/3531146.3533779

Wolf, M. J., Miller, K. W., & Grodzinsky, F. S. (2019). On the responsibility for uses of downstream software. In D. Wittkower (Ed.), 2019 Computer Ethics - Philosophical Enquiry (CEPE) Proceedings, (14 pp.). doi: 10.25884/7576-wd27 Retrieved from https://digitalcommons.odu.edu/cepe_proceedings/vol2019/iss1/3

Yi Zhang, Mengjia Wu, George Yijun Tian, Guangquan Zhang, Jie Lu. (2021). Ethics and privacy of artificial intelligence: Understandings from bibliometrics, Knowledge-Based Systems, Volume 222, 2021, 106994. (https://www.sciencedirect.com/science/article/pii/S0950705121002574