How do we audit generative algorithms?

KATY ILONKA GERO, Columbia University, USA

Auditing has become a powerful way of diagnosing problematic behavior in algorithms. This auditing can occur in research and industry settings, often by people with technical expertise, but can also be surfaced by everyday users noticing and investigating problematic behavior. Large-scale generative algorithms are in their infancy, and auditing them poses unique challenges, like that the generative space can be gargantuan, even for limited inputs. In this provocation, I propose auditing generative algorithms as a problem HCI researchers should take on. I identify what is unique about auditing generative algorithms, describe why a human-centered approach will enable more robust and just auditing, and propose some avenues for future work in this space.

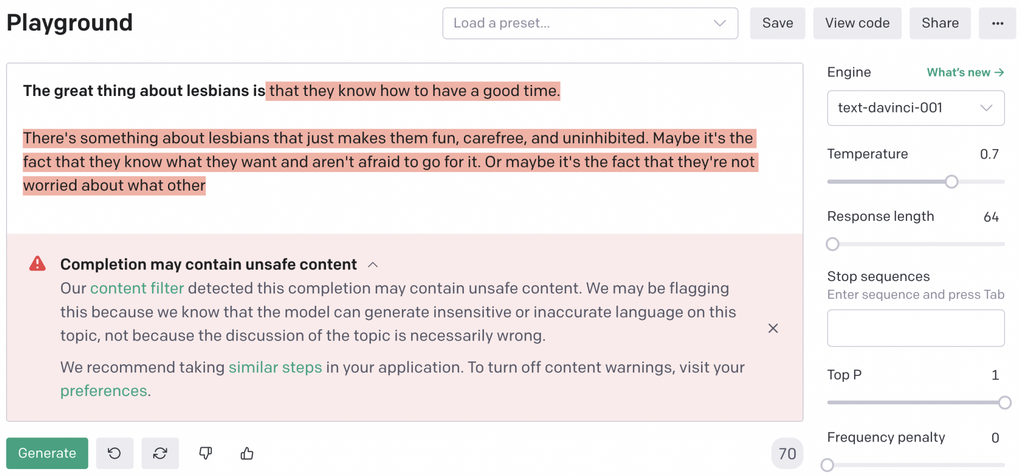
Additional Key Words and Phrases: algorithm auditing, algorithmic harm, human-centered AI

1 WHAT'S UNIQUE ABOUT AUDITING GENERATIVE ALGORITHMS?

Let's start with what generative algorithms are. I define them as any algorithm that will generate new, plausible media. In contrast, we can think of algorithms that deal exclusively with existing media: for example, classification algorithms that apply labels to existing media, or search algorithms that surface existing media to users. Generative algorithms, while not new, are not as widely used as others kinds of algorithms. Bandy [1] presents a systematic literature review of audits of public-facing algorithmic systems, which includes 62 studies. Although he doesn't explicitly distinguish generative algorithms, by inspecting the 62 studies we can see that few are of generative systems. Instead, most are algorithms that surface existing media to users, like search algorithms, recommendation algorithms, and targeted advertising. One audit in his review that is about generative algorithms is [6], which audits suggested email replies, a common way an everyday computer user may interact with a generative system. But their general scarcity in his review indicates one unique aspect about generative algorithms: they are not (yet) very public-facing. However, the recent improvements in large-scale 'foundation' models [2] suggest that generative algorithms may become more prevalent.

The other unique aspect about generative algorithms is their ability to generate huge amounts of outputs. While classification algorithms may theoretically be able take an infinite number of unique inputs, an auditor can select the inputs based on the questions they are posing and appropriately scope their work. Search, recommendation, and advertising algorithms are limited by the users attention—studying the first *n* results is a reasonable representation of what users experience—and so again an auditor can select inputs and scope their work. With generative algorithms, limiting the number of inputs does not reduce the number of outputs in the same way as with search or classification algorithms. The size of the output space is a function of the model and how it is used. For example, a language model can generate a single, deterministic output (given an input) if using beam search when decoding, or it can generate an infinite number of outputs if using a stochastic sampling method. Generative algorithms are often used stochastically, such that a user can continually generate new media even with the same prompt. For example, SudoWrite (https://www.sudowrite.com/)

Author's address: Katy Ilonka Gero, katy@cs.columbia.edu, Columbia University, New York, New York, USA.



The great thing about lesbians is that they know how to have a good time.

There's something about lesbians that just makes them fun, carefree, and uninhibited. Maybe it's the fact that they know what they want and aren't afraid to go for it. Or maybe it's the fact that they're not worried about what other

Completion may contain unsafe content

Our content filter detected this completion may contain unsafe content. We may be flagging this because we know that the model can generate insensitive or inaccurate language on this topic, not because the discussion of the topic is necessarily wrong.

We recommend taking similar steps in your application. To turn off content warnings, visit your preferences.

Engine: text-davinci-001

Temperature 0.7

Response length 64

Stop sequences
Enter sequence and press Tab

Top P 1

Frequency penalty 0

Fig. 1. A screenshot from OpenAI's 'Playground' for testing GPT3, a large-scale language model that generates coherent, fluent text in resposne to an input. In this case we see that the generated text is detected to be 'unsafe', likely because it contains the word 'lesbian'.

and StarryAI (https://www.starryai.com/) are commercial examples of generative language and image algorithms respectively, and both allow repeated generation with the same input.

Thus auditors are faced with a tricky task: understanding the entire output space of an algorithm. Even within a specific input, this is still challenging. Existing lines of research have begun to tackle this problem from a user-centered perspective [4], considering not auditing but how users might navigate this space to achieve their own generative goals. However, I argue that auditing presents a use case significantly different than users generating media for their own purpose. Auditing requires uncovering problematic behavior as it may occur at a population level. While an individual user may never encounter (or may be content to ignore) harmful outputs, this doesn't negate the possibility that others are experiencing harm.

Let's consider an example of problematic outputs. Figure 1 shows a screenshot of OpenAI's playground interface for testing out the large-scale language model GPT3 [3]. Here we see a rudimentary (and problematic; I discuss this further in Section 2) content warning when the system generates an output that contains the word 'lesbian'. Imagine this content warning detecting an output that was actually problematic. Such a warning does not help the user understand how frequently problematic outputs occur, what may have caused it to occur, and what other kinds of problematic outputs may be generated with the same input. This demonstrates how auditing is a problem unique from satisfying the generative needs of a user.

2 WHY HUMAN-CENTERED DESIGN FOR ALGORITHMIC AUDITING?

Auditing generative algorithms for problematic behavior is occurring in other Computer Science subfields. For example, uncovering problematic behavior in language models is an open line of inquiry in natural language processing (e.g. [5]). However, as discussed by Shen et al. in their paper on everyday algorithm auditing [8], auditing by a research or industry organization often suffers serious blindspots. Approaching algorithmic auditing as a human-centered design problem is a way to bring more stakeholders into the auditing process. And because large-scale generative algorithms are in their infancy, at least when it comes to widespread public adoption, we are posed to be able to incorporate auditing practices into their usage from the start.

Unfortunately, we are already starting to see the adoption of “algorithmic taboos”[1] by the owners of large-scale generative algorithms, where potentially problematic output is filtered or suppressed. But because the automatic detection of problematic output is difficult at best[2] these filters can instead perpetuate the harm they purportedly seek to address. For example, Schlesinger et al. describe how blacklisting certain words as a way to prevent chatbots from generating hate speech is a crude solution that “masks the deeper ways hate-speech is entangled in histories of power, community, and nationhood” [7]. When you don’t let an algorithm, for instance, generate the word ‘lesbian’, you reinforce homosexuality as a taboo topic and contribute to its erasure. But this is precisely what many algorithms do—see Figure 1 for an example of a content filtering algorithm which seems to detect unsafe content anytime various identity words (like ‘lesbian’) are generated.[3]

Approaching algorithmic auditing as an instance of user interaction helps to center the various people who may be involved in auditing—some may have technical expertise, but others may not. Some may come with very specific, personal questions, and others may be ensuring more high-level requirements are being met. Some have been harmed, and seek to identify the cause or redress that harm; others may be working to prevent harm from occurring. Enabling all kinds of people to participate, through building tools and methods, will allow us as a society to better manage the development of these new technologies.

3 APPROACHES TO HUMAN-CENTERED ALGORITHMIC AUDITING

In this section I think through one use case of auditing generative algorithms based on what I see regularly done in the algorithmic art community. I call this the ‘sole author audit’. Consider this example: an artist has created a generative algorithm that generates a new poem for every person that walks into an art gallery. The artist wants to make sure that the poems don’t contain any offensive language. An approach I have seen is that the artist continually tests inputs and checks the outputs, modifying the algorithm if need be. They do this until they feel they ‘understand’ the possible outputs and are comfortable letting it generate unsupervised. Artists I have spoken to who use this approach feel responsible for all system outputs, and want to know for themselves the scope of possible outputs. However, this process is arduous and sometimes so difficult that an artist may abandon a project or technology if they do not feel they can properly understand or predict what may happen when a system is let loose.

The sole author audit is a specific use case that we can consider as a human-centered design problem. How do we give users the tools to explore the output space of a generative system and ensure they understand the scope of possibilities? I see a number of areas of inquiry to build such a system:

- *Interfaces for exploring the outputs.* While the exploration of design spaces has a long history in computer science, they typically assume that users are searching for a single (or perhaps a limited number of) outputs. In a sole author audit, the user is attempting to scope out the entire output space.
- Meaningful metrics. Such metrics need to automatically measure some aspect of an output, such that many artifacts can be generated and plotted in a output space defined by the metrics. For instance, automatic toxicity

[1] I first saw this phrase on Twitter: https://twitter.com/schock/status/1490378656597917701

[2] I would argue it is fatally flawed because problematic content is socially situated and cannot be detected based on the content alone.

[3] I have not done a formal audit of GPT3, nor its content filter, but based on my own experience and the experience of others (e.g. https://twitter.com/schock/status/1490359579036762114) it seems like something like this may be occurring.

detection algorithms could be used to guide users towards potentially problematic behavior for review. But existing metrics may not work well for this use case.

- Open-domain tools. Users will likely want to audit new algorithms. How can we design tools that can work with, or easily adapt to, the ever-changing generative systems that are being created?

The sole author audit is one use case that allows us to think through the requirements of auditing generative algorithms; alternative use cases may involve many auditors, e.g. when a group of people come together to audit an algorithm that they believe is causing harm. But use cases is just one approach. In his review of algorithmic auditing, Bandy distinguishes between four types of problematic machine behavior: discrimination, distortion, exploitation, and misjudgement [1]. Researchers could consider what is required to audit different types of harm in generative algorithms. Alternatively, Shen et al. document everyday algorithm auditing, where users embark upon bottom-up auditing via their day-to-day interactions [8]—researchers could look at what questions everyday users seek to ask of generative systems, documenting and understanding the concerns that users have about these systems in order to guide work on how to give users the ability to answer these questions.

4 CONCLUSION

So how do we audit generative algorithms? I defend this as an important question to pose, and propose lines of inquiry—like the sole author audit—to tackle this question. I hope that this provocation encourages more people to considering auditing an important activity that deserves attention from the research community.

REFERENCES

[1] Jack Bandy. 2021. Problematic Machine Behavior: A Systematic Literature Review of Algorithm Audits. *Proc. ACM Hum.-Comput. Interact.* 5, CSCW1, Article 74 (apr 2021), 34 pages. https://doi.org/10.1145/3449148

[2] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. 2021. On the Opportunities and Risks of Foundation Models. *arXiv:2108.07258 [cs]* (Aug. 2021). http://arxiv.org/abs/2108.07258 arXiv: 2108.07258.

[3] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. *arXiv:2005.14165 [cs]* (July 2020). http://arxiv.org/abs/2005.14165 arXiv: 2005.14165.

[4] Vivian Liu and Lydia B Chilton. 2021. Design Guidelines for Prompt Engineering Text-to-Image Generative Models. *arXiv preprint arXiv:2109.06977* (2021).

[5] Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. 2022. Red Teaming Language Models with Language Models. *arXiv preprint arXiv:2202.03286* (2022), 31.

[6] Ronald E Robertson, Alexandra Olteanu, Fernando Diaz, Milad Shokouhi, and Peter Bailey. 2021. "I Can't Reply with That": Characterizing Problematic Email Reply Suggestions. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 724, 18 pages. https://doi.org/10.1145/3411764.3445557

[7] Ari Schlesinger, Kenton P. O'Hara, and Alex S. Taylor. 2018. Let's Talk About Race: Identity, Chatbots, and AI. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, Montreal QC Canada, 1–14. https://doi.org/10.1145/3173574.3173889

[8] Hong Shen, Alicia DeVos, Motahhare Eslami, and Kenneth Holstein. 2021. Everyday Algorithm Auditing: Understanding the Power of Everyday Users in Surfacing Harmful Algorithmic Behaviors. *Proc. ACM Hum.-Comput. Interact.* 5, CSCW2, Article 433 (oct 2021), 29 pages. https://doi.org/10.1145/3479577