If Software is My Copilot, Who Programmed My Software?

— by Bradley M. Kuhn, Policy Fellow, Software Freedom Conservancy

Preamble

The bulk of the essay that follows was written between June–August 2021 for submission for FSF's deadline on 23 August 2021. Much has changed since then. Most notably, from June 2021 until the publication date of this essay — in collaboration with my colleagues at Software Freedom Conservancy — we at Software Freedom Conservancy worked to launch an effort in the copyleft community to address the urgent issue presented for copyleft by GitHub Copilot specifically and AI-assisted software development generally. In our first step in this process, Software Freedom Conservancy chartered a committee of experts who will study this issue and make recommendations to help drive the public discussion. The essay below focuses on framing the problem and the largest concerns the Free Software community should consider.

The essay as written below is the first expression of my thoughts — written in the immediate wake GitHub's announcement. It seeks to begin a complex conversation for the Free Software community, as no definitive answers were available then, nor are they available now.

The Free Software Foundation has pledged a US$500 reward to all papers they republish. When I receive those funds, I will donate that amount to Speak Out, Act, Reclaim (SOAR) (formerly called "Victims Refuse Silence").

Introduction

Software freedom is our goal. Copyleft is a strategy to reach that goal. That tenet is oft forgotten by activists. Copyleft is even abused to advance proprietary goals. We too often see concern about the future of copyleft overshadow the necessary fundamental question: does a particular behavior or trend — and the inevitable outcomes of those behaviors and trends — increase or decrease users' rights to copy, share, modify, and reinstall modified versions of their software? That question remains paramount as we face new challenges.

Introduced first by Microsoft's GitHub in their Copilot product, computer-assisted software authorship by way of machine learning models presents a formidable challenge to software freedom's future. Yet, we can, in fact, imagine a software freedom utopia that embodies this technology. Imagine that all software authors have access to the global archive of machine learning models — and they are fullly reproducible. Everyone has equal rights to fork these models, train them further with their own datasets, provided that they must release new models (and the input code) freely in the global archive. All code produced by these models is also made freely available under copyleft. All code that builds the models, all historical input sets, and all trained models are all also made available to everyone under copyleft licenses.

While activists might quibble about minor details to optimize imagined utopia, this thought experiment shows computer-assisted software authorship does not inherently negate software freedom. Rather, the rules, requirements, and policies that apply will determine whether software freedom is respected. To paraphrase Hamlet: there is nothing either good or bad, but the policy makes it so.

What's the Worse That Could Happen?

> [They are] not a good [person] who, without a protest, allows wrong to be committed . . . with the means which [they] help to supply.
>
> — *John Stewart Mill, University of St. Andrews, 1 February 1867*

Obviously, ignoring machine learning for computer-assisted software authorship will not usher in this software freedom utopia. Copyleft activists cannot stand idly by in this situation, but we must temper our attention by considering the likelihood of *dystopian* and problematic outcomes, and the options available to prevent them.

In response to Copilot's announcement, pundits speculated, without evidence, a prevailing feeling of "Free Software had a good run, but I guess that's over now". Such predictions seem consistent with the well-documented overoptimism of artificial intelligence success. Rapid replacement of traditional software development methodologies seem unlikely. As such, we should not overestimate the likelihood that these new systems will **both** accelerate proprietary software development, **while** we simultaneously fail to prevent copylefted software from enabling that activity. The former *may not* come to pass, so we should not unduly fret about the latter, lest we misdirect resources. In short, AI is usually slow-moving, and produces incremental change far more often than it produces radical change. The problem is thus not imminent nor the damage irreversible. However, we must respond deliberately with all due celerity — and begin that work immediately.

Currently, there are two factors that influence the timing of our response. First, if GitHub's Copilot becomes a non-beta product available to the programming public, that would indicate necessity of an urgent response. Microsoft and GitHub are unlikely to share their product plans, so we cannot know for sure when this will occur. However, in the seven months since the first beta was made available, we've consistently heard anecdotally that more and more developers (particularly, FOSS developers!) have received beta invitations. Based on these (admittedly incomplete) facts, we must assume that a move from private beta to public deployment is imminent in 2022. This indicates some urgency of the problem.

Second, we *already know* that some of our worst fears are definitely true. Namely, that Microsoft and GitHub used copylefted software as part of Copilot's training set.

Copilot was trained on "billions of lines of public code ... written by others". While GitHub has refused requests to release even a list of repositories included in the training set, the use of the word "public" indicates that only software with source-available licenses (even if not FOSS licenses) were input into Copilot. Furthermore, GitHub admits that during training, the system encountered a copy of the GPL more than 700,000 times. This effectively confirms that copylefted public code appears in the training set.

When questioned, former GNOME developer and (at the time of writing) GitHub CEO, Nat Friedman, declared publicly "(1) training ML systems on public data is fair use (2) the output belongs to the operator". Friedman himself, as well as Microsoft and GitHub's other executives and lawyers, ignored Software Freedom Conservancy's requests for clarification and/or evidence supporting these statements.

Meanwhile, GitHub continues to improve this system, trained only on publicly source-available software, and seeks to market it to new users, including those who otherwise use FOSS development tools. Users continue to report gaining access to the beta and are noticing improvements. Microsoft and GitHub's public position is meanwhile clear: they claim to have no copyleft obligations for training the model, the model itself, and deploying the service. They also believe there are no licensing obligations for the output.

While Friedman ignored the community's requests publicly, we inquired privately with Friedman (then GitHub CEO) and other Microsoft and GitHub representatives in June 2021, asking for solid legal references for GitHub's public legal positions of (1) and (2) above. They provided none, and reiterated, without evidence, that they believed the model does not contain copies of the software, and output produced by Copilot can be licensed under any license. We further asked if there are no licensing concerns on either side, why did Microsoft not also train the system on their large proprietary codebases such as Office? They had no immediate answer. Microsoft and GitHub promised to get back to us, but have not.

This secrecy and non-cooperativeness is expected from a proprietary software company and its subsidiary, but leaves us only with speculative conclusions to inform a strategy for copyleft here. We can reliably guess that the companies *will* claim "fair use" as their primary justification for creating the model and offering the service, and will argue that both the output and the trained model are *not* "work[s] based on the Program" (GPLv2) nor do they "copy from or adapt all or part of the work[s] in a fashion requiring copyright permission" (GPLv3/AGPLv3). Furthermore, we can reliably conclude, given the continuing product promotion, that the companies have at least a medium-term commitment to Copilot.

In short, they have already hunkered down for a protracted disagreement. Their positions are now incumbent — using their resources and power to successfully charge copyleft activists to "prove them wrong". But we do not have to accept their unsubstantiated arguments at face value. In fact, these areas are so substantially novel that almost every issue has no definitive answers, but we must nevertheless begin to formulate our position and our response to Microsoft and GitHub's assault on copyleft.

Trained Models, Fair Use, and Copyright Infringement

Consider GitHub's claim that "training ML systems on public data is fair use". We have not found any case of note — at least in the USA — that truly contemplates that question. The *only* legal case in the USA to look near this question is Authors Guild v. Google, Inc., 804 F.3d 202 (2d Cir. 2015). The Supreme Court denied certiorari on this case; it is not legal precedent in all jurisdictions where Microsoft and GitHub operate.

Even more, that case considered a fact pattern centered around *search*, not *authorship of new/derived works*. Google had made copies of entire copyrighted books, not for the purpose of displaying them, but so users could (1) run search queries, and (2) see a "snippet" of the search hits (i.e., to see the search hit in context). The Second Circuit held Google's copying of the books was "fair use" because searching and providing context added value exceeding what a user could obtain from their own copies, and Google's product *did not* substitute the market for the books.

The analogous fact pattern for code is obvious: GitHub could offer a *search* tool that assists users in finding key public repositories (and specific lines of code within those repositories) that seemed to solve tasks of interest. Developers could then easily utilize those codebases in the usual, license-compliant ways. The actual Copilot fact pattern is not this one.

Meanwhile, the Authors Guild case begins and *ends* the list of major cases regarding machine learning systems and "fair use". We should simply ignore GitHub's risible claim that the "fair use question" on machine learning is settled.

Perhaps most importantly, in the USA, "fair use" is an affirmative defense to answer copyright infringement. In concrete terms, that means — particularly in cases where the circumstances are novel — a copyright holder brings an infringement lawsuit and *then* the alleged infringer shows in court that their actions met the relevant factors for "fair use" sufficiently. Frankly, we refuse to do these companies' job for them. Copyleft activists need not *tell Microsoft and GitHub why this isn't "fair use"*, rather, *they* need to tell *us* why training the model with copylefted code *is* "fair use" and prove that the trained model itself is not a "work based on" the GPL'd software.

GitHub has meanwhile artfully avoided the question of whether the trained model is a "work based on" the input. We contend that it probably is. However, given that “fair use” is an affirmative defense to copyright infringement, they are obviously anticipating a claim that the trained model is, in fact, a “work based on” the inputs to the model. Why else would they even bring up “fair use”, rather than simply say their use is fully non-infringing? Anyway, we have no way to even explore these questions authoritatively without examining the model, fully affixed in its tangible medium. We don’t expect GitHub to produce that unless compelled by a third party.

Indeed, discussion of these questions outside of a courtroom is moot. For this novel and contentious fact pattern, only a court decision can settle the matter adequately. As a strategic matter, copyleft activists should keep their own counsel about what we anticipate in the opposition’s “fair use” and/or non-infringement defenses, and the counter-arguments that we plan.

Copilot Users Should Worry

GitHub’s position does a great disservice to Copilot users. Their claim that “the output belongs to the operator” creates a false sense of legal justification. Users have already shown that Copilot can generate a substantial amount of unique, GPL’d code, and then (rather ironically, given GitHub’s claim that they removed the text of the GPL from the training set) *also* suggest a license that is non-copyleft. Friedman’s statement surely does not qualify as an indemnity for Copilot users who might face GPL enforcement actions. Users almost surely must construct their own “fair use” or “not copyrightable” defenses for Copilot’s output.

The length and detail of what Copilot can generate for users seems unbounded. The glaring example above appears primia facie to be copyright infringement; we expect further such problems. Consider the sheer amount that a fully functional and successful Copilot would generate. Surely, AI researchers seek the ability for Copilot to “figure out” that you are trying to solve some specific task when programming. The better Copilot gets at handing ready-made solutions to its users, the more likely it becomes that its output may offer the user copylefted software.

Copilot leaves copyleft compliance as an exercise for the user. Users likely face growing liability that only increases as Copilot improves. Users currently have no methods besides serendipity and educated guesses to know whether Copilot’s output is copyrighted by someone else. Proprietary software companies such as Synopsys provide so-called “scanning tools” — that can search your proprietary codebase and find hidden copylefted software. However, the Free Software tools for that job are in their infancy and unlikely to develop quickly, since historically those who want those tools are companies that primarily develop proprietary software and seek to avoid copylefted software.

We recommend users who wish to avoid infringing the copyrights of others simply avoid Copilot.

On Copyleft Maximalism and Unilateral Capitulation

Draconian copyright law generally horrifies software freedom activists for good reason. Nearly all copyleft activists would prefer a true, multilateral rewriting of copyright rules that prioritized the interest of the general public and software rights. Copyleft exists primarily *because of* the long-standing political non-viability of a copyright law reboot. Nothing has changed in this regard; if anything, changing legislation has become an even more expensive lobbying proposition than it was at copyleft's advent. Copyleft activists should expect, indefinitely, for proprietary software companies and media oligarchs to control copyright legislation.

Fortunately, copyleft was designed specifically for this eventuality. Activists have called copyleft the "judo move" of software freedom, since copyleft uses the powerful copyright force (invented primarily by our opposition) against itself. That realization leads to a painful, but pragmatically necessary, awkwardness.

The issues herein — from training of machine learning models, to the copyright questions about those models, to the derivation questions about their output — are novel copyright questions. As software freedom activists, we are uniquely qualified to invent an ideal copyright structure for these technologies. But, without a path to promulgate such replacement copyright rules into the incumbent system, that exercise is futile. Furthermore, systems outside of copyright — including but not limited to EULAs, business agreements and patents — have long been used to proprietarize software without the need of copyright. Reality of facts on the ground dictate that we not concede the only wedge we have to compel software freedom; that wedge *is* copyleft.

Meanwhile, proprietary software companies regularly exploit any unilateral concessions on weakening of copyleft that FOSS projects make, while continuing to pursue copyright maximalism for their works. Particularly in novel areas, we must assume a copyleft maximalist approach — until courts or the legislature disarm all mechanisms to control users' rights with regard to software. That adversarial process will frustrate us, but ultimately by choosing copyright as our primary tool, we already chose the courts as our battleground for contentious issues.

We all surely have our opinions about how copyleft should operate in these novel situations. We have even expressed some such opinions herein. But, ultimately, strong copyleft licenses do not defer the "what's covered?" question to one individual or organization. The "judo" power comes from strong copyleft reaching to all of what copyright governs. When those issues are novel — and companies flaunt that novel manipulation of copylefted works — only a court can answer definitively.

A Community-Led Response

While these companies will likely not succeed in their efforts to disarm copyleft, they have nevertheless attacked the entire copyleft infrastructure. We must mount an effective response.

Software Freedom Conservancy has spent the last six months in deep internal discussions about this novel threat to the very efficacy of copyleft. We have a few ideas — a mix of short-term, medium-term and long-term strategies to address the problem. However, we recognize that a community approach is needed — at least for this problem. Thus, putting first things first, we realized that we should gather the best minds in the software freedom community with direct experience in copyleft theory and practice. We have convened these individuals to a committee specifically chartered by Software Freedom Conservancy to — as quickly as reasonably possible – publish a series of recommendations to the community on how we should respond to both the immediate threat to copyleft found in Copilot, and (long-term) analyze the more general threat that AI-assisted programming techniques pose to the strategy of copyleft.

Finally, as much as can be done during the pandemic using FOSS tools available, we will attempt to convene public discussions as much as possible. We will contemporaneously publish the committee's minutes publicly. If you'd like to get involved today in public discussions about this issue, please join the mailing we launched for this topic. ■