Analyzing the Effectiveness of Copyright Law on AI Output in Software Engineering and Future Developments

Surendra Jammishetti, sjammish@ucsc.edu

***Abstract*—TODO: need to do this later**

***Index Terms*—AI, Software Engineering, Copyright**

I. Introduction

THE emergence of AI tools has, no doubt, changed many aspects of how work is approached in many professions, but the greatest impact has been on software engineering. New tools like Github Copilot, an AI assisted code generation tool thats usable in many modern code editors today, boasts a 2x speedup of feature completions and a large boost in productivity for its users [1]. However accurate those claims may be, theres no doubt that these tools are being used widely by the developer population, and they arent going away any time soon.

However the code generated by these tools must come from somewhere. Many models source their training data from the public internet, but not everything in public access allows for commercial / derivative use. Such models ignore this step, jumping straight towards using this data with any disregard for copyright protections in place. While copyright has been slow in the past, there is no doubt the legal systems around the world will regulate and form precedent for these utilites; Inevitably placing their users under risk.

For those concerned with the legality of the software they produce, the usage of AI-generated code should be avoided as their trustworthiness is dubious and they will become succeptible to copyright law in the near future.

II. Background

To help understand the context that this paper resides in, below are lay-person explanations of AI training, Software Licensure, AI tools, and more.

A. Data Acquisition

The way AI development labs gather data for their models is mainly through web scrapers. Web scrapers are programs that access a immense number of websites, downloading and orgranizing the all the data they find [2]. The gathered data is then used to train whichever model they are interested in.

The modern way for a website to prevent a webscraper from downloading its contents is a file called a ‘robots.txt‘, that any honest web scraper would look for first, and obey the rules inside [3]. The problem with this ”trust me” approach is that a dishonest, or even badly programmed, web scraper can come along and look past this file and do whatever they please[3].

B. Software Licenses

While it may come as a suprise to the average person, even code nowadays has some kind of protection against copying / un-permitted usage. The mechanism for this is called a ”Software License”[4]. The code that these licenses protect is called ”source code”. There exist many kinds of licenses, mainly meant for open use towards the public, but there are a two that are crucial to understand.

- Copy Left
  A ”Copy Left” license allows the general public to view and do whatever they would like with the source code, but they enforce that any derivative must also use the same license. In effect this is a strictly anti-commercial license as its quite hard for a buisness to use Copy Left Licensed software as anything they produce using that code must have the same Copy Left license, and as a result live in the public domain. [4]
- Proprietary
  A ”Proprietary” license are the most restrictive, as they prevent any viewer / user from copying, modifying, redistrbuting, etc [4]. They are the defaco type of license for any commercial code, as they protect code the best, and are legally viable [5].

C. AI-Powered Tools

The most popular AI tool for developers is Github Copilot, which helps developers as they write code. Its most important capability, in the context of this work, is its power to automatically generate code when you ask. The same capability exists for Claude, OpenAI’s models, and others, where they have the capacity to take a description of what the user wants, and can write code to perform that task [6]. For example, If we want to make our own snake game, we could trivially ask a model to program the snake game.

D. Copyright

III. AI and Code Generation

IV. Tracking

V. Discussion and Summary

Acknowledgment

The authors would like to thank...

REFERENCES

[1] E. Kalliamvakou, *Research: Quantifying github copilot's impact on developer productivity and happiness*, May 2024. [Online]. Available: https://github.blog/news-insights/research/research-quantifying-github-copilots-impact-on-developer-productivity-and-happiness/.

[2] Miquido. "What is ai data scraping? "[Online]. Available: https://www.miquido.com/ai-glossary/what-is-ai-data-scraping/.

[3] *What is a robots.txt*, 2024. [Online]. Available: https://www.cloudflare.com/learning/bots/what-is-robots-txt/.

[4] P. Odence. "Five types of software licenses you need to understand: Black duck blog. "[Online]. Available: https://www.blackduck.com/blog/5-types-of-software-licenses-you-need-to-understand.html.

[5] H. L. Review, *Google llc v. oracle america, inc.* Nov. 2021. [Online]. Available: https://harvardlawreview.org/print/vol-135/google-llc-v-oracle-america-inc/.

[6] M. S, *How to use github copilot: Setting up and learning various useful ai coding methods*, Jun. 2023. [Online]. Available: https://www.hostinger.com/tutorials/how-to-use-github-copilot.