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 **github**. 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 **Miquido**. 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 **robots**. 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**robots**.

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”**Blackduck**. 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. **Blackduck**
- Proprietary
  A ”Proprietary” license are the most restrictive, as they prevent any viewer / user from copying, modifying, redistrbuting, etc **Blackduck**. They are the defaco type of license for any commercial code, as they protect code the best, and are legally viable **harvard**.

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 **s˙2023**. For example, If we want to make our own snake game, we could trivially ask a model to program the snake game.

D. Copyright

Copyright is a mechanism used to protect created works from direct copying but not a sufficiently derivative use or one that gives adequate credit to the original author **stokes2021**. There has been one landmark case regarding copyright and programming, Google INC v. Oracle America, Inc, which regards the copying of generic and widely known API code by Google from Oracle **harvard**. API in short means that its code that acts as a abstracted layer, such that the user doesnt have to think about whats going on underneath. Think of it like me making my own laptop, but still keeping around the QWERTY keyboard so the users of my laptop dont have to learn a whole new keyboard layout. The ruling stated that its alright for google to copy this code because they were doing it in the interest of the general public and thought it wasnt possible to copyright something so generic, leading to google winning the case.

III. AI and Code Generation

AI has been getting better and better at generating code **codesignal**, due to the vasty increasing training data these models are harvesting from the web **lacour˙2024**. Especially with many popular code repositories being open online, its fairly straightforward to assume that they are being used for training. While its impossible to know exactly whats being used as training data and what isnt by these large companies, they arent making it any easier with their lack of clarity and reassurance **willison˙2023**. Therefore, it can be reasonably assumed that licensed code that is public purview is also being used as training data.

The issue then arises with the licensure of the code ingested by the model. Whether its under a ”Proprietary” license or a ”Copy Left” license, the end user could be in trouble.

A Proprietary license would bar any derivative usage, putting the user in deep trouble if found out, whether they’re trying to publicize their code or not.

A Copy Left, while less troublesome, an entity trying to privatize their code would fail to uphold the conditions of the license, which demands that any derivative works also be in the public domain.

We havent even gotten to the fact that many licenses require attribution to the original author, which would be completely lost in this process of ai ingestion and generation.

IV. Detecting the Origins of Generated Code

While it may seem impossible, If there was a way to figure out where the code generated by an AI came from, any users of the generated code could be in huge trouble depending on the licensure of the source code. Such technology doesnt exist today but is tending towards that direction, as seen in this study: **ma2024**. They dive deep into the possibility of detecting whether or not code generated by a model can be traced back / verified to be in some dataset.

Obviously this hasn’t been attempted on mainstream models, but with their findings being so fruitful, its only a matter of time before new research builds off this foundation to see if generated code has its roots on public code. Logically then, source code authors could audit whether or not others have indirectly used their source code, via this AI ingestion and generation proxy.

V. Forward Compliance

VI. Discussion and Summary

Acknowledgment

The authors would like to thank...