SheppardMullin

Solving Open Source Problems With AI Code Generators - Legal issues and Solutions

PART 1 - LEGAL ISSUES

By: James Gatto

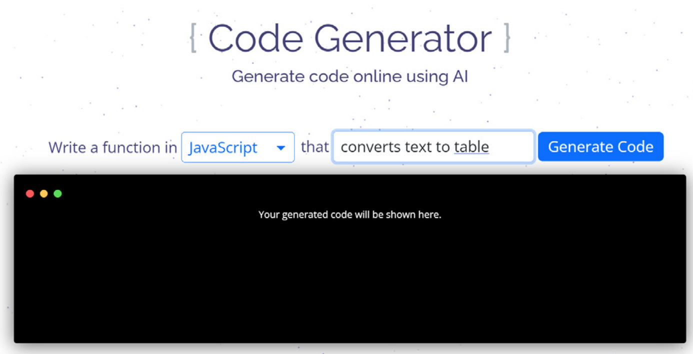
AI-based code generators are a powerful application of generative AI. These tools leverage AI to assist code developers by using AI models to auto-complete or suggest code based on developer inputs or tests. These tools raise at least three types of potential legal issues:

- Does training AI models using open source code constitute infringement or, even if the use is licensed, does doing so require compliance with conditions or restrictions of the open source licenses?
- Does using the output of an AI code generator subject the developer to infringement claims?
- Does use of AI-generated code by developers creating a new software application require the application to be licensed under an open source license and its source code to be made available?

This article will address these legal issues and discuss some practical solutions to abate these problems. Part 1 of the article covers the legal issues. Part 2 will cover solutions.

AI Code Generators

These tools can greatly simplify and expedite the code development process. The AI models used are typically trained on billions of lines of code, mostly publicly available open source code. Based on a developer request and existing code, the tool can generate suggested code ranging from snippets of code to fully coded functions. This is done in real time in a matter of seconds. These tools are easy to use and work with many programming languages. A simple example is shown below.



Training AI Code Generator Models

The training data for AI code generator models is typically based on huge repositories of open source code. Many people think that because the code used is open source it can be freely used with no legal problem. After all, the point of open source is to freely permit its use. And true open source licenses do not discriminate against the use to which open source software is put.[1]

Open Source License Basics

Open source software is typically free to use but that freedom is based on a license that accompanies the software. Most open source licenses permit the user to copy, modify and redistribute the open source code. However, these freedoms come with conditions. These conditions vary by license and can range from simple compliance obligations to more onerous, substantive requirements.

Examples of sample compliance obligations include maintaining copyright notices, providing attribution and including the license terms with any redistribution. The more substantive provisions can include the requirement that any software that includes or is derived from the open source software must be licensed under the terms of the open source license and the source code for that software must be made freely available. These conditions are often referred to as "tainting" of the software. The licenses with these permissions are often called "restrictive" open source licenses.

For commercial developers, who desire to develop proprietary software that can be licensed for a fee under a proprietary license, tainting is a huge problem. The value of software is severely diminished if the developer must license it under an open source license and make the source code available. The reason is that the open source license gives recipients the right to copy, modify and redistribute that software at no charge.

Whether simple compliance or more substantive obligations, failure to comply with those terms can result in legal problems. Failure to comply can be deemed a breach of contract. Or it can result in termination of the license and loss of right to use the open source software. Continued use after termination can give rise to claims for copyright infringement.

Open Source Legal Issues With AI Code generators

Does training AI models using open source code constitute infringement or, even if the use is licensed, does doing so require compliance with conditions or restrictions of the open source licenses?

Training AI models using open source code alone does not likely constitute infringement.[2] As explained above, typically open source licenses do not impose restrictions on the use of the open source code. However, legal problems can arise if the open source license compliance obligations are not satisfied.

A recent lawsuit against CoPilot, an AI code generator alleges that in training the model using open source code, the tool stripped copyright notices and license terms from the the code in violation of the licenses. It alleges that the output of CoPilot copies the code (or portions of it) yet does not include the copyright information or attribution notices or satisfy other compliance obligations. The legal claims include breach of contract for violation of license terms, violation of the DMCA Section 1202 for removing copyright management information (CMI) and various other

---

1 For example, one of the fundamental tenets of the criteria for open source is the" license must not restrict anyone from making use of the program in a specific field of endeavor." The Open Source Definition, https://opensource.org/osd/

2 Depending on how the open source code is obtained, other issues may arise. For example, some open source code repository platforms have terms of use that cover use of their platform. Violation of those terms could present certain problems, but those issues are beyond the scope of this paper.

claims. Section 1202 prohibits intentionally removing or altering any CMI or distributing works knowing that CMI has been removed or altered.

Training AI models and stripping out CMI might be a violation of the DMCA Section 1202. And it may constitute breach of contract for failure to comply with the relevant open source license terms. However, each of these issues will be fact specific. One of the facts depends on the specific license terms. For example, some open source licenses require the compliance obligations be met if the open source code is *redistributed*.[3] Arguably, if Company A downloads open source code and uses it to train its own models, on its own servers, at that point it is not yet a redistribution by the company. If Company A's AI code generator *outputs that code* in response to a user request, that likely becomes a redistribution.

Another factual issue relates to how Company A trains its model. If the model includes the open source code, likely it may need to maintain the CMI in that code. However, if the model is generated by learning information about the code and later creates new code based on this information, the issues may be different. In this case, the model itself may not include a copy of the code. Some of the AI code generators claim they do not look up copies of code to generate their output.[4]

The bottom line is that it is necessary to consider the license terms and the method of training and using the model to assess whether any legal violation has occurred by training an AI model with open source code.

Does using the output of an AI code generator subject the developer to infringement claims?

Because open source licenses permit copying, modifying and redistributing the open source code, outputting the code from the AI tool alone may not be an infringement. However, if the code is output and license compliance obligations are not satisfied, that may be breach of contract. Under some open source licenses, such breach may result in termination of the license. Continued use after termination may constitute infringement.

Does use of AI-generated code by developers creating a new software application require the application to be licensed under an open source license and its source code to be made available?

If the code output from an AI code generator is covered by a restrictive open source license, use of that code in another program taints that program. As explained above, this requires the program as a whole to be licensed under the same terms as the restrictive open source license and requires the source code for the entire program to be made available. This means recipients can copy, modify and redistribute the program for free. This is not an ideal solution if the desire is to build proprietary software that can be licensed for a fee.

Solving the Open Source Problems with AI-Based Code Generators

The trio of problems addressed above seem insurmountable to some people. Many companies are banning the use of AI-code generators by their developers to avoid tainting issues and to minimize the likelihood of getting dragged into a lawsuit for infringement. This is a legally safe option but prevents developers from obtaining the benefits of AI code generators. Fortunately, there are a number of practical solutions that can mitigate these risks and enable developers to safely use AI code generators.

3 For example, some versions of the BSD license state: "*Redistributions* in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution."

4 For example, the CoPilot website states that it "generates new code in a probabilistic way, and the probability that they produce the same code as a snippet that occurred in training is low. The models do not contain a database of code, and they do not 'look up' snippets. Our latest internal research shows that about 1% of the time, a suggestion may contain some code snippets longer than ~150 characters that matches the training set."

In part 2 of this article, I will discuss some of these solutions. As a preview, these solutions can include:

- filters to prevent the output of problematic code
- code referencing tools to flag problematic output
- code scanning tools to assist developers with open source compliance.

Conclusion

Many companies have banned the use of AI code generators by its developers due to the legal risks and uncertainty resulting from the use of open source code to train the models. In my view, these issues are manageable by using various known solutions. Use of these solutions can significantly mitigate the risk and uncertainty of using AI-code generators. Developers, companies and their in-house counsel struggling with how to manage legal risks with AI code generators will definitely want to learn about these solutions.

For Part 2 of this article and more details on generative AI, please contact:

James Gatto
Blockchain and Fintech Team Co-leader
bio
202.747.1945
jgatto@sheppardmullin.com

Sheppard Mullin's Blockchain Technology and Fintech team helps clients develop innovative and comprehensive legal strategies to take advantage of what may be the most disruptive and transformative technology since the Internet. We focus on advising clients on how to meet their business objectives, without incurring unnecessary legal risk. Our team includes attorneys with diverse legal backgrounds who collectively understand the vast array of legal issues with and ramifications of blockchain technology and digital currencies. More Information