SheppardMullin

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

By: James Gatto

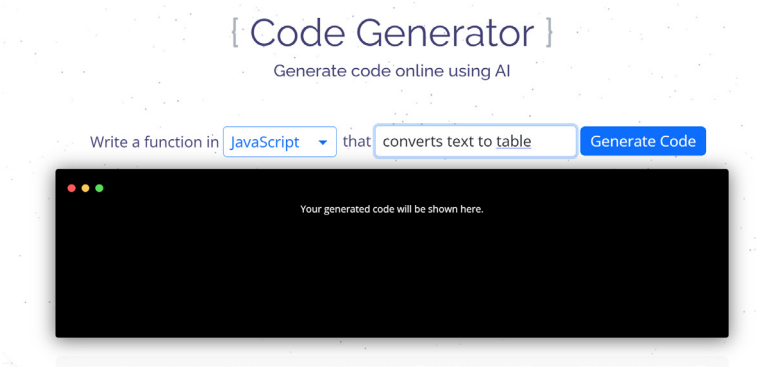
AI-based code generators are a powerful application of generative AI. These tools leverage AI to assist code developers by using AI models to auto-complete or suggest code based on developer inputs or tests. These tools raise at least three types of potential legal issues:

- 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.

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 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

[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.

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 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.

---

[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."

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.

Potential Solutions to Mitigate Open Source Legal Risks with AI Code Generators

- 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.

In apparent recognition of the potential open source legal issues, some of the leading AI code generators have optional features. To mitigate legal risk, some companies are mandating, as part of their AI policies, that employees use these features.

Below, I summarize some AI code generator tools, how they describe the operation of their tool and features they offer to help companies manage certain open source issues.

Copilot

The Copilot website states the following about its operation:

> GitHub Copilot's suggestions[5] are all generated through AI. GitHub Copilot 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. Previous research showed that many of these cases happen when GitHub Copilot is unable to glean sufficient context from the code you are writing, or when there is a common, perhaps even universal, solution to the problem.

Copilot Solutions

Filter

Copilot offers a solution that includes a filter "to help detect and suppress GitHub Copilot suggestions which contain code that matches public code on GitHub." Use of this filter is optional for individuals accessing Copilot via an individual account.[6] But use of the filter can be mandated by an Enterprise administrator for access via a business account.[7]

---

[5] Copilot refers to the outputs of its AI code generator as "suggestions."

[6] Copilot for Individual users have the choice to enable that filter during setup on their individual accounts.

[7] For Copilot for Business users, the Enterprise administrator controls how the filter is applied. They can control suggestions for all organizations or defer control to individual organization administrators. These organization administrators can turn the filter on or off during setup (assuming their Enterprise administrator has deferred control) for the users in their organization.

When the filter is enabled, Copilot checks code suggestions against public open source code on GitHub. If there is a match, the suggestion is not shown. Assuming this works as intended, this filters out known open source code and prevents it from being shown to the developer user.

Reference Tool

Copilot has another tool [announced] called a "reference." This tool provides a "reference" for suggestions that resemble public code on GitHub so the developer user can make "a more informed decision about whether and how to use that code." This means, if the suggestion contains known open source code the developer (or counsel) can determine the relevant license and assess the potential risks of using that code. Most notably, as addressed in Part 1 of this article, this assessment can ensure there is no tainting of proprietary software and/or reveal any compliance obligations.

CodeWhisperer

According to its website:

> As a generative AI, CodeWhisperer creates new code based on what it has learned from the code that it was trained on and the context that you provided as prior code and comments. While CodeWhisperer is not designed to reproduce code that it was trained on, it is possible that on rare occasions it will generate code that closely matches particular code snippets in the training data.

Filter

According to its website, CodeWhisperer also can flag or filter code suggestions that resemble open-source training data and provide the associated open-source project's repository URL and license so developer users or counsel can more easily review and assess them.

Reference

CodeWhisperer also offers a reference tracker that detects whether a code suggestion is similar to particular CodeWhisperer open-source training data. If CodeWhisperer detects that its output matches particular open-source training data, the built-in reference tracker will notify you with a reference to the license type and a URL for the open-source project. The reference tracker can flag such suggestions or optionally filter them out.

In addition, all references can be logged for later review later (e.g., by counsel) to enable a developer to adopt the suggestion and keep coding without interruption.

The options for these tools are in the configuration setting for CodeWhisperer. For the free CodeWhisperer Individual Tier users, this setting is available in the IDE. With CodeWhisperer Professional, the AWS administrator can centrally configure this setting on an organization level from the AWS Management Console.

Independent Scan

Many companies with well-established open source policies and procedures routinely use an open source code scanning tool (e.g., Black Duck) to ensure that their code includes no problematic open source and so they can ensure compliance obligations are met. For companies that already do this, this scan can detect any open source code that may have been output as a suggestion from an AI code generator and adopted by the company's developer.

For companies without an established open source policy and/or do not do open source code scans before releasing their software, you should! This is true whether or not you permit your developers to use AI code generators. If you do permit your developers to use AI code generators this is just one more reason to develop an effective open source policy and do code scans.

Mandating Use of AI Code Generator Tools In Company AI Policies

When adopting a corporate AI use policy you should consider various issues regarding use of AI code generators. The following are some issues to consider:

- Do you want to permit employees to use AI code generators?
- If so, further consider:
  - whitelisting an approved set of AI code generators based, at least, on whether they have adequate tools (e.g., filters, referencing and/or other tools) that prevent your employees from seeing known open source code as suggestions and/or reference the license if the suggestion is based on open source
  - requiring the use of a business account where an administrator can configure the settings to mandate use of these tools
  - ensuring that any referenced open source is vetted based on the company's existing open source policy and/or by legal counsel.

And where the cost is justified, consider running your own open source code scan to further ensure there is no tainting or other legal issues and that you can understand and abide by any compliance obligations.

Besides adopting an AI use policy, it is advisable to update your open source policies to address AI code generator issues.

Conclusion

AI has great promise and can bring great efficiency to help you developers. But it can also raise several thorny legal issues. Fortunately, there are solutions to help manage the legal risks. It is prudent for companies to develop and/or update their AI use polices and open source policies to ensure responsible use by employees (and contractors).

Many companies just getting started on these policies often find that a custom in-house presentation on these topics is a great way to start and get their arms around the issues. We routinely do presentations to help companies understand the issues that should be addressed in employee AI use policies and various options for addressing the issues. We also help create these policies. If you have questions on developing AI use or open source policies, reach out to me.

For more details on generative AI, please contact:

James Gatto
Artificial Intelligence Team Co-leader
Open Source Team Leader
+1 (202) 747-1945
jgatto@sheppardmullin.com

James Gatto is a partner in the D.C. office of Sheppard Mullin and is Co-Leader of its Artificial Intelligence Team and leader of the Open Source Team. For over 35 years he has been a thought leader on legal issues with emerging technologies and business models, including over 20 years of experience with AI and open source. He provides strategic advice on all IP, tech transactions and tech regulatory issues. He is an adjunct professor at Ole Miss Law School where he teaches Legal Issues with AI. He has been an invited speaker for the US Copyright Office Listening Session on AI Authorship and the USPTO Listening Session on AI Inventorship Issues. He is an appointed member of the AI/Machine Learning Task Force of the American Bar Association's IP Section. He is also a member of the AI committees of the American Intellectual Property Law Association (AIPLA) and the International Technology Law Association.

This alert is provided for information purposes only and does not constitute legal advice and is not intended to form an attorney client relationship. Please contact your Sheppard Mullin attorney contact for additional information.