tabnine

Generative Models and Open-Source Licenses

Generative models are being increasingly used for software-development tasks. These models can be trained on various types of data sources, including proprietary software code. In this blog post, we will explore the implications for using GPL-protected code in generative models.

First, it is important to understand what the General Public License (GPL) is and how it protects software code. The GPL is a widely used open-source software license that ensures that the source code of a software application is available to the public. It also provides specific permissions and restrictions on how the code can be used, modified, and distributed. If a software application is licensed under the GPL, anyone who uses or modifies the code must also release their modifications under the same license, ensuring that the software remains free and open-source.

Now, let's consider how using GPL-protected code in generative models can be problematic. The main issue is that the GPL requires that any derivative work of GPL-licensed software must also be licensed under the GPL. Therefore, if a generative model is trained on GPL-protected code, it is not clear whether the output generated by the model would also need to be licensed under the GPL. This could limit the ways in which the generated code can be used or distributed, as any downstream users would need to comply with the GPL's requirements.

On the other hand, permissive open-source licenses like the MIT, Apache, or BSD licenses allow for more flexibility in how the code can be used and distributed. These licenses often have fewer restrictions and do not require that derivative works be licensed under the same license. Therefore, using permissive open-source code in generative models can provide more freedom and flexibility in how the generated output is used.

What is Tabnine doing about Open-Source Licenses

Tabnine models are trained on permissive open source licenses. The main licenses that are considered in the Tabnine training set are: Apache-2.0, MIT, BSD-2-Clause, BSD-3-Clause, Unlicense. There are other close variants of these licenses that are included in the dataset (see full list) but they all enjoy the same kind of permissive policy.

Can I get the full list of repositories Tabnine used in Tabnine training?

Yes. You can get the list of all repositories and licenses used to train Tabnine models by check the Tabnine Trust Center (https://trust.tabnine.com/)

I am a Developer, can I remove my Repository from the Tabnine Training set?

Yes. Tabnine respects that not all developers want their data to be used for training purposes. There are two ways to get your repository removed from future training runs of Tabnine:

1. Contact Tabnine Support at support@tabnine.com, please provide your GitHub username and the repositories that you own and would like to be excluded from training.
2. Add a file name `tabnine_robots.txt` to the root directory of your repository. The content of this file is described below.

Conclusion

In conclusion, it is important to train generative models only on permissive open-source code and not on code with restrictive licenses such as GPL. Using GPL-protected code can lead to licensing issues and restrictions on how the generated output can be used and distributed.

Tabnine provides full transparency on the code used for training its models, as well as a mechanism for excluding code from future training runs.

Further Reading

https://thegradient.pub/machine-learning-ethics-and-open-source-licensing/ and

https://thegradient.pub/machine-learning-ethics-and-open-source-licensing-2/

https://githubcopilotlitigation.com/

https://www.theverge.com/2022/11/8/23446821/microsoft-openai-github-copilot-class-action-lawsuit-ai-copyright-violation-training-data

https://www.theverge.com/2023/1/28/23575919/microsoft-openai-github-dismiss-copilot-ai-copyright-lawsuit

Full List of Licenses in the Dataset

Apache-2.0 (https://spdx.org/licenses/Apache-2.0)

Artistic-2.0 (https://spdx.org/licenses/Artistic-2.0)

BSD-2-Clause (https://spdx.org/licenses/BSD-2-Clause)

BSD-3-Clause (https://spdx.org/licenses/BSD-3-Clause)

BSD-3-Clause-Clear (https://spdx.org/licenses/BSD-3-Clause-Clear)

BSD-3-Clause-No-Nuclear-License-2014 (https://spdx.org/licenses/BSD-3-Clause-No-Nuclear-License-2014)

CC0-1.0 (https://spdx.org/licenses/CC0-1.0)

ECL-2.0 (https://spdx.org/licenses/ECL-2.0)

ISC (https://spdx.org/licenses/ISC)

MIT (https://spdx.org/licenses/MIT)

MIT-0 (https://spdx.org/licenses/MIT-0)

MIT-feh (https://spdx.org/licenses/MIT-feh)

SHL-0.51 (https://spdx.org/licenses/SHL-0.51.html)

Unlicense (https://spdx.org/licenses/Unlicense)

Excluding code from being used in Training

You can exclude your repository, or parts of your repository, from being considered for Tabnine training by using the `tabnine_robots.txt` file.

Adding this file to the top directory of your repository allows you to define how Tabnine uses your code for training purposes.

Keeping the `tabnine_robots.txt` file empty means that Tabnine will ignore the repository as a whole.

You can provide more fine-grained instructions to Tabnine similarly to how `robots.txt` works for website crawlers. Each line of the file can start with either `Include` or `Exclude` followed by a path expression in the repository.

For example, consider the following `tabnine_robots.txt` file

```
Include: /src/
Exclude: /test/
```

Means that Tabnine would ignore all code under the `test` directory, but will consider all code under `src`.