Qeios

Peer Review

Review of: "An Empirical Study on Automatically Detecting AI-Generated Source Code: How Far Are We?"

Lior Shamir[1]

1. Michigan Technological University, United States

The paper studies the efficacy of tools that identify AI-generated content and focuses on AI code. Several LLM tools are used to generate code, and different AIGC detectors are tested. The experiments span different kinds of classifiers to test their performance on the various text generators. The paper does not solve the problem but provides useful information that has practical application in the software development world. The analysis is comprehensive, and the authors invested substantial labor in performing a broad range of experiments. The paper is well-organized. The study is limited to the datasets used by the authors, the assumptions about the code, and the specific implementation decisions (e.g., parameter settings). The work is not perfect at this point, and research efforts will continue to fully profile this interesting problem. Still, the results are interesting, and future work can provide further analysis that will allow for a better understanding of this practical problem.

The data are not available, nor is the code. The dataset is rather small because of the manual inspection, but since labor has been invested in it, the dataset can be used by others if it is shared. It will also allow for the comparison of new results to the results shown in the paper. The implementation decisions would have also been clearer if the code could be accessed.

The AIGC detectors are limited to the classifiers trained and tested in the paper. It is a very partial list and does not necessarily reflect the true ability of available AIGC to identify AI-generated code.

Some of the classifiers that were tested provided good performance, much higher than 0.6. That is somewhat not emphasized or discussed in detail in the paper. Some of them are mentioned, but the reader can wonder about the *reasons* for the differences in these specific classifiers.

Shallow learning results such as SVM or RF are not shown.

doi.org/10.32388/3B31WW

The purpose of the cosine similarity analysis is not clear. How is this information useful?

It is interesting that the comments do not allow identification of how the code was generated. An experiment of just classifying the comments as a pure text classification would have made the paper stronger. Combining the comments with the code might make it more difficult for the classifier, and therefore classifying the code and the comments separately would have made sense. Those can also be used as ensemble classifiers.

Some minor comments:

The sentence: “Due to advancements in the field of Natural Language Processing (NLP), LLMs have seen substantial progress in their performance and widespread use” needs to be corrected. LLMs are not driven by advancements in NLP. LLMs mimic NLP but are not driven by NLP advancement.

The following sentence is “More recently, source code has also been included to train the LLMs with the goal of helping with software development activities.” That also needs to be corrected. Chat-GPT 1 was already trained with very large corpora of source code and had good coding abilities.

Figure 1 does not add much to the paper.

There is no need to define accuracy, F1, TPR, TNR, etc. These are textbook terms.

Some LaTeX tags are still in the manuscript and need to be corrected. In its current form, the reader needs to guess which figure or table is being referenced.

Declarations

Potential competing interests: No potential competing interests to declare.