

Research Experience

Burak Yetistiren | burakyetistiren.com/research

For my research so far, I have evaluated the performance of GitHub Copilot with Dr. Eray Tuzun and my friend Isik over the last year. This evaluation is accepted from PROMISE'22 in the form of a conference paper. Our work is an empirical evaluation; the generated code by GitHub Copilot is being evaluated in terms of three metrics, namely code validity, code correctness, and code efficiency. We made our assessment utilizing a benchmark dataset containing 164 problems called the HumanEval Dataset. I attended PROMISE'22 in person to present my paper. Moreover, I also attended the ESEC/FSE'22 conference with support from SIGSOFT.

At the beginning of the work, it was just Dr. Tuzun and myself. I was just introduced to research; hence Dr. Tuzun guided me intensively during the first few weeks, then regularly in our weekly meetings. Therefore, I have written the first draft of our paper; of course, it has changed considerably, but the changes we made were on top of this draft. Moreover, I have coded our experimental setup (Available at: github.com/burakyetistiren/Assessing-the-Quality-of-GitHub-Copilot-s-Code-Generation). During these phases, Dr. Tuzun continued guiding me and reviewing my work, assigning me new work each week. When Isik joined us, she first became familiar with our work, reviewed our experimental setup, and improved it. After we decided to introduce two new sub-experiments to our work, we divided them into two, conducted them and discussed the results separately, and reviewed each other's work. Throughout our work, we read our papers numerous times and gave our reviews. We also shared our paper with other lab members for them to review it a couple of times. In terms of paper reviewing, I regularly review the papers of other lab members before their submission.

As of now, we are working on a journal extension for our paper. This extension will be submitted to the "Empirical Software Engineering" Journal. In this work, we plan to compare the performances of code generation tools such as GitHub Copilot and Amazon CodeWhisperer. Additionally, we explained our work to Microsoft, which in turn might result in a collaboration in the form of a conference paper about automated program repair for the incorrect code generated by GitHub Copilot. To be able to incorporate the increased amount of workload, we now have new members in our group. Furthermore, as I am the member with the most experience in particular to our work about code generators, I am in a position where I lead the group after Dr. Tuzun.

I am also familiar with the process of paper submission and presentation. I was assigned formal (or bureaucratic) work, which was sending the paper to the conference, giving the author information, sending emails to the conference officials or related individuals, etc. As I previously mentioned, I presented our work in person, in Singapore.

Assessing the Quality of GitHub Copilot's Code Generation

Abstract: The introduction of GitHub's new code generation tool, GitHub Copilot, seems to be the first well-established instance of an AI pair-programmer. GitHub Copilot has access to a large number of open-source projects, enabling it to utilize more extensive code in various programming languages than other code generation tools. Although the initial and informal assessments are promising, a systematic evaluation is needed to explore the limits and benefits of GitHub Copilot. The main objective of this study is to assess the quality of generated code provided by GitHub Copilot. We also aim to evaluate the impact of the quality and variety of input parameters fed to GitHub Copilot. To achieve this aim, we created an experimental setup for evaluating the generated code in terms of validity, correctness, and efficiency. Our results suggest that GitHub Copilot was able to generate valid code with a 91.5% success rate. In terms of code correctness, out of 164 problems, 47 (28.7%) were correctly, while 84 (51.2%) were partially correctly, and 33 (20.1%) were incorrectly generated. Our empirical analysis shows that GitHub Copilot is a promising tool based on the results we obtained, however further and more comprehensive assessment is needed in the future.

<https://doi.org/10.1145/3558489.3559072>