# Article information:

(PDF) On the Robustness of Code Generation Techniques: An Empirical Study on GitHub Copilot
<https://www.researchgate.net/publication/367961391_On_the_Robustness_of_Code_Generation_Techniques_An_Empirical_Study_on_GitHub_Copilot>

# Article summary:

1. GitHub Copilot can generate entire functions from natural language descriptions, but it is unclear how robust it is to semantic-preserving changes in the description.

2. An empirical study was conducted on 892 Java methods to understand whether different but semantically equivalent descriptions result in the same recommended function.

3. Results show that modifying the description results in different code recommendations in approximately 46% of cases, and differences in semantically equivalent descriptions may impact the correctness of the generated code by up to 28%.

# Article rating:

Appears moderately imbalanced: The article provides some useful information, but is missing several important points or pieces of evidence that would be required to present the discussed topics in a balanced and reliable way. You are encouraged to seek a more balanced perspective on the presented issues by exploring the provided research topics and looking at different information sources.

# Article analysis:

该文章提出了一个重要的问题，即深度学习（DL）代码生成器的鲁棒性。作者通过对GitHub Copilot进行实证研究，探讨不同但语义等效的自然语言描述是否会导致不同的代码推荐结果。然而，该文章存在一些潜在偏见和局限性。

首先，作者没有考虑到自然语言描述的质量可能会影响生成代码的正确性。他们只关注了不同描述之间是否会产生不同的推荐结果。这种方法忽略了自然语言描述本身可能存在歧义或错误，并且可能导致生成错误的代码。

其次，作者没有考虑到开发人员使用GitHub Copilot时可能面临的风险和挑战。例如，如果开发人员过于依赖Copilot生成代码，则可能会忽略对代码质量和安全性的检查和测试。此外，Copilot还需要访问大量开源项目以进行训练，这可能涉及版权问题和隐私问题。

此外，在文章中并未提供足够的证据来支持作者所得出结论。例如，在46％情况下修改描述会导致不同的代码建议是一个相当高的比例，并且需要更多数据来验证这个结论是否普遍适用于其他编程语言和应用场景。

最后，该文章似乎倾向于支持DL-based code generators，并将GitHub Copilot视为“de facto state-of-the-art tool”。这种偏袒可能会影响作者对其他技术或方法的评估，并使读者误解其研究结果。

因此，在评估DL-based code generators时，我们需要更加谨慎地考虑其鲁棒性、正确性、可靠性以及与其他技术或方法之间的比较优劣。同时，我们也需要注意到使用这些工具时可能面临的风险和挑战，并采取适当措施来减轻它们带来的影响。

# Topics for further research:

* Quality of natural language descriptions
* Risks and challenges of relying on code generators
* Copyright and privacy concerns with training data
* Insufficient evidence to support conclusions
* Biases towards DL-based code generators
* Considerations for evaluating code generators

# Report location:

<https://www.fullpicture.app/item/72a0f60d6b93b237c7b0d7611c1ec9c0>