Evaluating Code Quality Generated in Large Language Models: A Multi-Language Empirical Study

تقييم جودة الشفرة البرمجية المولدة في النماذج اللغوية الكبيرة
(دراسة تجريبية متعددة اللغات)

Dr. Rasha Abdulaziz Bin-Thalab[1], Osamah Abdullah Abduljalil[2]
[1]Associate Professor, Department of Computer Engineering, College of Engineering and Petroleum, Hadhramout University, Yemen
[2]Master's student, Computer Science Department, College of Computer and Information Sciences, Imam Mohammad Ibn Saud Islamic University, Saudi Arabia
[1]r.binthalab@hu.edu.ye, [2]osamah.abduljalil@gmail.com

Received: 10/10/2025 Accepted: 5/11/2025

ABSTRACT

Software engineering has undergone a significant transformation due to the rapid development of Large Language Models (LLMs). Modern LLMs like GitHub Copilot, GPT-3.5, and GPT-4 are becoming more adept in generating useable source code in a range of programming languages with little human intervention. However, there is still debate over their outputs' internal quality, maintainability, and documentation. The study uses three popular programming languages—Python, Java, and JavaScript—to empirically evaluate how successfully LLMs generate code for issues of varying difficulty. SonarQube was used to examine the generated code and quantify cognitive difficulty, cyclomatic complexity, code smells, and comment ratios after the collection of a dataset of algorithmic tasks and independently solving the data via each model. The findings demonstrate that GPT-4 regularly creates code that is easier to maintain and understand than previous models, while Copilot consistently produces more comments but less structurally sound code. GPT-3.5 performs mediocrely and has moderate variability.

Key Words:
- **Large Language Models**
- **Code Generation**
- **Software Quality**
- **Copilot**
- **GPT**

الملخص:

شهدت هندسة البرمجيات تحولًا كبيرًا بسبب التطور السريع لنماذج اللغة الكبيرة (LLMs). أصبحت نماذج اللغة الكبيرة الحديثة مثل GitHub Copilot و GPT-3.5 و GPT-4 أكثر مهارة في إنشاء شفرة مصدر قابل للاستخدام في مجموعة من لغات البرمجة مع تدخل بشري ضئيل. ومع ذلك، لا يزال هناك جدل حول الجودة الداخلية لمخرجاتها وقابليتها للصيانة والتوثيق. تستخدم هذه الدراسة ثلاث لغات برمجة شائعة — Python و Java و JavaScript — لتقييم مدى نجاح نماذج اللغة الكبيرة (LLMs) في إنشاء شفرات برمجية لمشكلات متفاوتة الصعوبة. تم استخدام SonarQube لفحص الشفرات التي تم إنشاؤها وقياس الصعوبة المعرفية والتعقيد

الكلمات المفتاحية:
- نماذج اللغة الكبيرة
- توليد الشفرة البرمجية
- جودة البرمجيات
- Copilot
- GPT

الدوري ومشاكل الشفرات ونسب التعليقات بعد أن تم جمع مجموعة بيانات من المهام الخوارزمية وحلها بشكل مستقل بواسطة كل نموذج. تظهر النتائج أن **GPT-4** ينتج بانتظام كودًا أسهل في الصيانة والفهم من النماذج السابقة، في حين ينتج **Copilot** باستمرار تعليقات أكثر ولكن كودًا أقل صلابة من الناحية الهيكلية. يقدم **GPT-3.5** أداءً متوسطًا ويتميز بتقلب معتدل.	