PuppyCodeReview: An AI-based Code Review System

Chun-Hsiung Tseng [1*], Hao-Chiang Koong Lin[2], Chih-Wei Huang[3], Kai-Chun Hou[4], Jia-Rou Lin[1]

[1] Dept. of Electrical Engineering YuanZe Univ. Taoyuan, R.O.C.

[2] Dept. of Information and Learning Technology National Univ. of Tainan Tainan, R.O.C.

[3] Dept. of Psychology Fo Guang Univ. Yilan, R.O.C.

[4]Dept. of Digital Technology Design, National Taipei University, Taipei, R.O.C.

[*] lendle@saturn.yzu.edu.tw

Abstract: *This study proposes an artificial intelligence (AI)-based code review system, PuppyCodeReview, designed to streamline the code review process. While traditional peer code review is gaining importance in programming education, its effectiveness in university courses remains under question. PuppyCodeReview leverages AI technology to automate code reviews, providing suggestions across multiple aspects, including design, functionality, complexity, testing, naming, and commenting. The system also detects code smells, such as code duplication and long methods.*

Keywords: CodeReview, AI, Programming

1. Background

In emerging programming education methods, peer code review (Code Review / Peer Review[7]) has gained significant attention in recent years. Originally used as an agile development practice in the software industry, peer code review involves two distinct roles in the software development process: the coder (who writes the code) and the reviewer (who reviews the code). These roles can overlap or be interchanged. The primary objective of peer code review is to identify and rectify errors that were not detected during the initial development phase, thereby maintaining code quality. Many well-known international software companies, such as Google, implement internal peer code review practices to reduce coding errors[8]. Kubincová and Csicsolová attempted to introduce peer code review into high school programming courses. Their study found that 42% of students reported discovering more previously unnoticed coding errors, while 28% of students became more willing to spend time writing or modifying code compared to before (Kubincova & Csicsolova, 2018). However, Chong et al. have questioned the effectiveness of applying peer code review in university programming courses. They argue that due to university students' limited experience in software development, their ability to accurately identify coding errors remains highly uncertain (Chong et al., 2021). To remedy this, we proposed PuppyCodeReview, which is a AI-based code review system, to simplify the code review process.

2. Design and Implementation

[7] https://en.wikipedia.org/wiki/Code_review

[8] https://google.github.io/eng-practices/review/

The current implementation uses the Qwen2.5-Coder model to review submitted codes. The prompt below is used (some details are skipped to save space):

Please review the student's ${language} code. The purpose of this program is: ${objective}. First, let me know if the code is correct. Then, based on the provided guidelines, give review suggestions. Please respond in JSON format using the following structure:
... (output format omitted) ...Next, the student's code is: ${studentCode}.
The reference solution is: ${answer}. The review should consider the following aspects: Design, Functionality, Complexity, Tests, Naming, and Comments.Additionally, avoid the following code smells: Code Duplication, Long Methods, God Class, Too Many Parameters, Feature Envy, Inappropriate Intimacy, Refused Bequest, Lazy Class, Artificial Complexity, Overly Long Identifiers, Overly Short Identifiers, and Excessive Use of Literals.

The figure below demonstrates the output after reviewing a student's work:

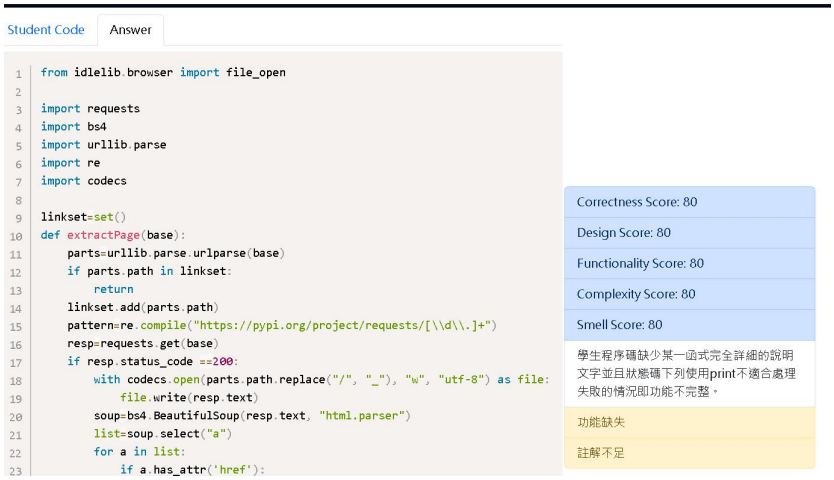

Figure 2 System Output

As shown in the figure, the student's code and the correct answer will be shown on the left hand side. On the right hand side, the score of each measurement will be shown and the system will generate some keywords for the code.

References

Chong, C. Y., Thongtanunam, P., & Tantithamthavorn, C. (2021). Assessing the Students' Understanding and their Mistakes in Code Review Checklists: An Experience Report of 1,791 Code Review Checklist Questions from 394 Students. *2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*, 20 – 29. 2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET). https://doi.org/10.1109/ICSE-SEET52601.2021.00011

Kubincova, Z., & Csicsolova, I. (2018). Code Review in High School Programming. *2018 17th International Conference on Information Technology Based Higher Education and Training*

(ITHET), 1 – 4. 2018 17th International Conference on Information Technology Based Higher Education and Training (ITHET). https://doi.org/10.1109/ITHET.2018.8424617