Hyunjae Suh

Irvine, CA hyunjas@uci.edu hyunjas.github.io hyunjae-suh-607119146

Summary

Ph.D. student specializing in Software Engineering, with a research focus on analyzing and improving Large Language Model (LLM)-generated source code. Experienced in conducting empirical studies, evaluating accessibility, and developing techniques to enhance the quality, detectability of AI-generated code.

Education

University of California, Irvine *Irvine, CA*
Ph.D. in Software Engineering *Sep 2023 – Present*

Kookmin University *Seoul, South Korea*
BS in Computer Science *Mar 2017 – Aug 2023*

Experience

Ph.D. Researcher in Software Engineering and AI *Irvine, CA*
University of California, Irvine *Sep 2023 – Present*

- Conducting empirical research on Large Language Models (LLMs) for software engineering, focusing on code generation, accessibility, and quality assessment.
- Analyzing key characteristics of LLM-generated source code, including its accessibility, and similarity to human-written code.
- Designing prompting strategies (Few-Shot, Self-Criticism, Multi-Agent Debate, etc.) to enhance the accessibility of AI-generated source code.
- Leveraging accessibility evaluation tools (IBM Equal Access, AChecker) to assess LLM-generated code compliance with WCAG 2.1.

Graduate Research Assistant – AI & Software Engineering *Remote*
eBay *Aug 2023 – Dec 2023*

- Developed and fine-tuned open-source LLMs for automated commit message generation, improving software documentation efficiency.
- Investigated the effectiveness of prompt engineering techniques to optimize commit message accuracy, clarity, and contextual relevance.
- Conducted comparative analysis of open-source LLMs to evaluate their performance in real-world software engineering workflows.

Publications

An Empirical Study on Automatically Detecting AI-Generated Source Code: How Far Are We?
Hyunjae Suh, Mahan Tafreshipour, Jiawei Li, Adithya Bhattiprolu, Iftekhar Ahmed
Accepted at the 47th IEEE/ACM International Conference on Software Engineering [ICSE 2025]

Does the Order of Fine-tuning Matter and Why?
Qihong Chen, Jiawei Li, **Hyunjae Suh**, Lianghao Jiang, Zheng Zhou, Jingze Chen, Jiri Gesi, Iftekhar Ahmed
https://arxiv.org/abs/2410.02915

Projects

Detection of LLM-generated Source Code *Research accepted at ICSE 2025*

- Designed and implemented techniques for detecting LLM-generated source code by leveraging fine-tuned LLMs, code embeddings, and feature-based analysis.

- Achieved 82.55 F1-score, demonstrating state-of-the-art performance in distinguishing LLM-generated from human-written code.
- Analyzed key features of LLM-generated code, including syntax patterns, and semantic coherence, to improve detection accuracy.

Accessibility of LLM-generated Source Code

- Investigated the web accessibility of AI-generated code, applying WCAG 2.1/2.2 guidelines to assess compliance.
- Developed and evaluated prompting techniques (Few-Shot, Self-Criticism, Multi-Agent Debate) to generate more accessible LLM-produced source code.
- Integrated accessibility evaluation tools (IBM Equal Access, AChecker) to measure and improve the usability of LLM-generated web applications.

The Impact of Fine-tuning Order on Language Models for Software Engineering

- Built fine-tuning pipelines for transformer-based LLMs, experimenting with different ordering of software engineering tasks (e.g., clone detection, defect detection, code translation, code repair).
- Analyzed the transfer learning effects of fine-tuning order on downstream task performance.
- Conducted quantitative experiments on different LLM architectures, measuring the impact of incremental fine-tuning for software engineering applications.

Teaching Experience

University of California, Irvine *Irvine, CA*
Teaching Assistant *Sep 2023 – Present*

- IN4MATX 115 - Software Testing, Analysis, and Quality Assurance
- ICS 10 - How Computers Work
- ICS 32 - Programming with Software Libraries