An LLM-Based Framework for Simulating, Classifying, and Correcting Students' Programming Knowledge with the SOLO Taxonomy

Shan Zhang
University of Florida
Gainesville, FL, USA
zhangshan@ufl.edu

Pragati Shuddhodhan Meshram
University of Illinois at Urbana-Champign
Urbana, Illinois, USA
psm12@illinois.edu

Priyadharshini Ganapathy Prasad
University of Florida
Gainesville, FL, USA
ganapathyprasadp@ufl.edu

Maya Israel
University of Florida
Gainesville, FL, USA
misrael@coe.ufl.edu

Suma Bhat
University of Illinois at Urbana-Champign
Urbana, Illinois, USA
spbhat2@illinois.edu

Abstract

Novice programmers often face challenges in designing computational artifacts and fixing code errors, which can lead to task abandonment and over-reliance on external support. While research has explored effective meta-cognitive strategies to scaffold novice programmers' learning, it is essential to first understand and assess students' conceptual, procedural, and strategic/conditional programming knowledge at scale. To address this issue, we propose a three-model framework that leverages Large Language Models (LLMs) to simulate, classify, and correct student responses to programming questions based on the SOLO Taxonomy. The SOLO Taxonomy provides a structured approach for categorizing student understanding into four levels: Pre-structural, Uni-structural, Multi-structural, and Relational. Our results showed that GPT-4o achieved high accuracy in generating and classifying responses for the Relational category, with moderate accuracy in the Uni-structural and Pre-structural categories, but struggled with the Multi-structural category. The model successfully corrected responses to the Relational level. Although further refinement is needed, these findings suggest that LLMs hold significant potential for supporting computer science education by assessing programming knowledge and guiding students toward deeper cognitive engagement.

CCS Concepts

• **Applied computing → Education**.

Keywords

Computer Science Education, Large Language Model, Solo Taxonomy

ACM Reference Format:
Shan Zhang, Pragati Shuddhodhan Meshram, Priyadharshini Ganapathy Prasad, Maya Israel, and Suma Bhat. 2025. An LLM-Based Framework for Simulating, Classifying, and Correcting Students' Programming Knowledge with the SOLO Taxonomy. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 2 (SIGCSE TS 2025), February 26-March 1, 2025, Pittsburgh, PA, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3641555.3705125

1 Introduction

Novice programmers often struggle with designing computational artifacts and fixing code errors, leading them to rely on external support or abandon tasks. To enhance their persistence and reduce frustration, researchers have explored scaffolding strategies to improve how novices approach programming [4]. However, in order to provide effective approaches to scaffold students' programming learning process, it is important to first evaluate their conceptual, procedural, and strategic/conditional programming knowledge. In addition, students in introductory programming classes often exhibit varying levels of background knowledge, experience, and understanding. This variation poses challenges for instructors who want to tailor their support to meet the diverse needs of their students.

To address these challenges, one approach is to develop a scalable, and efficient way for ascertaining students' understanding, so instructors can better tailor their support. One such method that has gained significant attention is the Structure of the Observed Learning Outcome (SOLO) taxonomy, a framework widely studied by CS educators for providing a consistent approach to evaluating students' levels of understanding [1]. It's often applied to assess programming skills such as code comprehension, code writing, and algorithmic design [2] through individual assessments.

Moreover, recent advances in large language models (LLMs) have enabled the automation of code generation and error correction, providing new opportunities to support novice learners at scale [3]. In this study, we propose a three-model framework that leverages the capabilities of LLMs to simulate, classify, and correct student responses, effectively reflecting their programming

SIGCSE TS 2025, February 26-March 1, 2025, Pittsburgh, PA, USA

ACM ISBN 979-8-4007-0532-8/25/02
https://doi.org/10.1145/3641555.3705125

knowledge. Specifically, we aim to examine this research question: To what extent is the Large Language Model (GPT-4o) effective in simulating, classifying, and correcting student responses based on the SOLO Taxonomy in computer science education?

2 Methods

In our three-model framework, based on respective prompts, the first model stimulates 100 student responses by referring to real students' responses, the second model classifies these 100 responses using the SOLO Taxonomy, and the third corrects those responses to help students progress from surface-level understanding to deeper cognitive engagement. Details about the SOLO Taxonomy, dataset, prompts, and human evaluation can be found below.

SOLO Taxonomy. We adopted the SOLO Taxonomy by modifying it to focus on the types of knowledge demonstrated during code comprehension and excluding the "extended-abstract" level, as it involves broader conceptual connections beyond the scope of basic programming tasks and we did not expect students at this intro-level stage to relate the program to more abstract or advanced contexts. In this study, the four categories in the Taxonomy are as follows: **Pre-structural** refers to a lack of correct answers and understanding of basic constructs like loops and conditionals. **Uni-structural** describes a correct answer but with an unclear understanding of the program's purpose, as well as the roles of loops and conditionals. **Multi-structural** involves a correct answer and an understanding of loops and conditionals, but only a partial and unclear understanding of the overall program's purpose. Finally, **Relational** denotes a correct answer with a full understanding of the program, including the roles of loops and conditionals.

Dataset. This study collected data from ten undergraduate computer engineering students through a think-aloud session, where they analyzed a program calculating the sum of even Fibonacci numbers. Their responses were categorized into four SOLO Taxonomy levels to assess their understanding of programming constructs.

Prompt. These are the key components of our prompts: 1) the program used in the previous think-aloud study; 2) the program output, detailing its purpose, logic, and key elements to clarify code functionality and expected results; 3) definitions of the four categories of the SOLO Taxonomy; 4) a Q&A section with examples of actual student responses from the previous think-aloud study for each category; and 5) our prompt description, which guides the generation of student responses in line with the SOLO Taxonomy definitions, following a specified format for consistency.

Model One: Simulating Student Responses. In the first model, we used GPT-4o to generate student responses that mirrored the SOLO levels by including all five components in the prompt. For each of the four categories, we first generated five student responses, resulting in 25 in total. These responses were reviewed by one author who conducted the previous think-aloud study independently to ensure that the generated outputs accurately reflected the target SOLO Taxonomy level. After validation, we expanded the simulation to generate 100 responses across all four categories.

Model Two: Classifying Student Responses. The second model classifies the 100 simulated responses according to the SOLO Taxonomy. The prompt includes all components except for Q&A as we do not want to disrupt the model's classification. After getting 100 responses' classifications, the same author reviewed these 100 students' responses no matter which category they were in Model One and Model Two.

Model Three: Correcting Student Responses. Based on the human classification of these 100 student responses, we prompted GPT-4o to modify the students' responses to align with the Relational category definition with human evaluation. This prompt includes all components except for the program output, as we aim to avoid directly providing GPT-4o with the fully correct answers to the program.

3 Results

The results from our three-model framework illustrate varying levels of success across the different SOLO Taxonomy categories. Model 1, which focused on generating student responses, performed exceptionally well in the Relational category, achieving 100% accuracy with all 25 responses. In the Uni-structural category, 88% (22 out of 25) of responses were correct, while performance dropped to 60% (15 out of 25) in the Pre-structural category. However, Model 1 struggled significantly with the Multi-structural category, failing to generate any accurate responses. Similarly, Model 2, which focused on classifying 100 student responses, achieved 100% accuracy in the Relational category. The Uni-structural category showed lower accuracy at 66% (19 out of 29 responses correctly classified), and the Pre-structural category performed better at 92% (11 out of 12). Like Model 1, Model 2 struggled with the Multi-structural category, where no responses were correctly classified.

Model 3, which aimed to correct student responses in the Pre-structural and Uni-structural categories (as there were no human-validated responses in the Multi-structural category from Model 1), successfully transitioned 10 randomly selected responses to the Relational level. Upon human evaluation, all 10 responses were validated as correct. These results demonstrate that while GPT-4o performed well in generating and classifying responses at the Relational level, it faced challenges in the Multi-structural category, where understanding tends to be more fragmented. The model's ability to move from lower levels to the Relational level shows the potential of LLMs in computer science education, assessing programming knowledge and promoting deeper engagement. However, further refinement is needed to address gaps in understanding, especially in the Multi-structural category.

Acknowledgments

This study is supported by the National Science Foundation and the Institute of Education Sciences under Grant DRL-2229612.

References

[1] John B Biggs and Kevin F Collis. 2014. *Evaluating the quality of learning: The SOLO taxonomy (Structure of the Observed Learning Outcome).* Academic Press.

[2] David Ginat and Eti Menashe. 2015. SOLO Taxonomy for assessing novices' algorithmic design. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education.* 452–457.

[3] Majeed Kazemitabaar, Xinying Hou, Austin Henley, Barbara Jane Ericson, David Weintrop, and Tovi Grossman. 2023. How novices use LLM-based code generators to solve CS1 coding tasks in a self-paced learning environment. In *Proceedings of the 23rd Koli Calling International Conference on Computing Education Research.* 1–12.

[4] Yulia Pechorina, Keith Anderson, and Paul Denny. 2023. Metacodenition: Scaffolding the problem-solving process for novice programmers. In *Proceedings of the 25th Australasian Computing Education Conference.* 59–68.