Adaptive Knowledge Assessment in Simulated Coding Interviews

Michael Ion[1] MIKEION@UMICH.EDU
Sumit Asthana[2] ASUMIT@UMICH.EDU
Fengquan Jiao[2] FENGQUAN@UMICH.EDU
Tianyi Wang[2] TYWANGSH@UMICH.EDU
Kevyn Collins-Thompson[1,2] KEVYNCT@UMICH.EDU
[1] *University of Michigan School of Information, Ann Arbor, MI U.S.A.*
[2] *University of Michigan Dept. of Computer Science and Engineering, Ann Arbor, MI U.S.A.*

Abstract

We present a system for simulating student coding interview responses to sequential interview questions, with the goal of accurately inferring student expertise levels. With these simulated students, we explored fixed and adaptive question selection policies, where the adaptive policy exploits a knowledge component dependency graph to maximize information gain. Our results show that adaptive questioning policies show increasing benefits compared to a fixed policy as student expertise levels rise, achieving expert assessment F1-scores of 0.4-0.8 for student expertise prediction compared to 0.25-0.35 for fixed strategies.

Keywords: adaptive assessment, simulated students, question generation, coding evaluation, AI tools

1. Problem Space

Designing effective assessments of student programming knowledge through interviews can help accurately assess their problem-solving and subject-matter mastery (Kannam et al., 2024). As use of generative AI tools becomes more commonplace with students learning how to code, real-time interviews in the tradition of *viva voce* exams offer educators ways assess to assess genuine student understanding. However, research shows there are struggles with both the logistics of scaling such 'live' assessments and evaluating responses consistently and without bias (Novak et al., 2023). Optimizing and evaluating the design and results of an interview-based assessment also requires data on student interactions, which is difficult to obtain at scale because of privacy concerns and implementation time and expense.

To address this problem we developed a pool of *simulated students* as an intermediate testing and optimization tool. Our system models different levels of student knowledge of specific programming knowledge components, which is then elicited via a sequence of interview questions. With these simulated responses and conversation history, we can systematically study in advance in a scalable, low-risk way how different questioning policies are likely to perform across varying levels of student expertise, and then deploy the most effective strategy in the final real-time setting with actual students.

2. Methodological Framework

We use the KLI framework (Koedinger et al., 2012) and its notion of *knowledge components* (KCs), which represent discrete cognitive skills inferred through performance on related

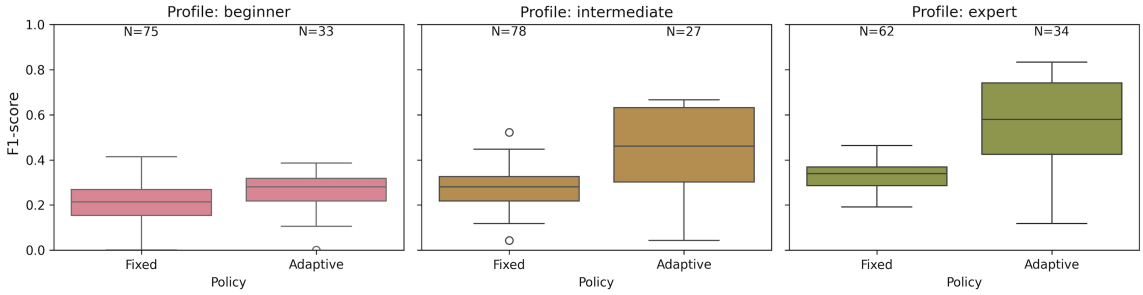

Figure 1: F1-score distributions by policy ($n = 10$ questions) and expertise level, showing how the adaptive questioning effectiveness increases with student expertise level.

tasks. We apply the large language model (LLM) `GPT-4o` to identify and map KCs in individual students' programming work into a prerequisite knowledge graph, where edges show dependencies between concepts. We compare two question selection strategies: an *adaptive policy* that selects the KC that maximizes information about the student's knowledge state, and a *fixed policy* that randomly samples KCs weighted by their uncertainty (modeled using a Beta distribution). At each question step, the selected KC is used to generate a natural language question using the same LLM. We defined three student profiles based on progressive mastery of Python concepts: beginner (e.g., basic syntax and data structures); intermediate (e.g., control structures, functions, basic error handling); and expert (e.g., advanced debugging, optimization, system-level operations). The final inferred student level after questioning was evaluated using F1 score distributions.

3. Preliminary Results

Adaptive questioning policies show increasing benefits compared to a fixed policy as student expertise levels rise (Fig. 1). For expert-level students, the adaptive policy shows an interquartile range (IQR) of F1-scores of 0.4-0.8, substantially higher than the fixed policy's narrower IQR of 0.25-0.35. For intermediate students, the adaptive policy typically performs in the 0.3-0.6 range compared to 0.2-0.3 for the fixed policy. For beginners, the performance distributions largely overlap, with both policies showing F1-scores primarily between 0.15-0.3, indicating less evidence for adaptive questioning at this level.

4. Discussion and Future Work

The hierarchical nature of programming knowledge appears to be particularly well-captured by our adaptive policy, especially when assessing complex knowledge structures characteristic of expert-level students. As part of our validation efforts, we are complementing our simulation-based findings with in-person interviews of data science masters students. Our preliminary results suggest that conversations simulated using LLMs can provide meaningful metrics for educational assessment, opening up new possibilities for conducting educational research at scale. **Acknowledgements.** This research was sponsored in part by the University of Michigan School of Information.

References

Suhas Kannam, Yuri Yang, Aarya Dharm, and Kevin Lin. Code Interviews: Design and Evaluation of a More Authentic Assessment for Introductory Programming Assignments, November 2024. URL http://arxiv.org/abs/2410.01010. arXiv:2410.01010 [cs].

Kenneth R. Koedinger, Albert T. Corbett, and Charles Perfetti. The knowledge-learning-instruction framework: bridging the science-practice chasm to enhance robust student learning. *Cognitive Science*, 36(5):757–798, July 2012. ISSN 0364-0213, 1551-6709. doi: 10.1111/j.1551-6709.2012.01245.x. URL https://onlinelibrary.wiley.com/doi/10.1111/j.1551-6709.2012.01245.x.

Ed Novak, Peter Ohmann, and Marshall College. Oral Exams in CS-Education: Pros and Cons in the Age of AI Assisted Programming. *Journal of Computing Sciences in Colleges*, 39(3):32–33, 2023.