

小学高年级学生在 LLM 辅助编程学习中的行为模式研究

Behavioral Pattern of Upper Elementary Students in Programming Learning Assisted by LLM

杨占山¹, 应晶鑫², 许洁³, 孙丹^{4*}

^{1,2,4} 经亨颐教育学院 杭州师范大学

³ 教育学院 浙江大学

* dansun@hznu.edu.cn

【摘要】 大语言模型 (LLM) 的影响正逐步扩展到低年级学生, 然而鲜有研究探索小学生在使用大语言模型过程中的细粒度行为。本研究采用混合方法, 通过研究团队开发的 CodewithAI 平台, 考察了 64 名小学高年级学生使用 LLM 开展编程学习的情况。研究发现, 高年级小学生主要通过 LLM 寻求对当前 Python 代码的解释并提出问题, 他们经常将代码提交给 LLM 以检查其准确性。研究进一步在所有参与研究的小学高年级学习者中发现了新手探索者、高级编程者、积极调试者和困难编程者聚类。鉴于这些发现, 本研究为未来的教学设计和小学高年级学生人工智能工具的开发提出了教学和开发建议。

【关键词】 大语言模型; 小学高段; 编程学习; 行为分析

Abstract: The influence of Large Language Models (LLMs) is progressively extending to younger students; however, there is a scarcity of research examining the fine-grained behaviors of primary students utilizing LLMs. This study employs a mixed-methods approach, utilizing the CodewithAI platform to investigate the programming learning experiences of 64 upper elementary students with the aid of LLMs. Findings reveal that students primarily seek explanations for their current Python code and pose questions, frequently submitting their code for accuracy checks. Furthermore, the study identifies distinct clusters among the participating upper primary learners, namely novice explorers, advanced coders, active debuggers, and struggling coders. Finally, this study offers pedagogical and developmental recommendations for instructional design and the creation of AI tools tailored for younger students.

Keywords: large language model, upper primary students, programming learning, behavioral analysis

1. 前言

近年来, 生成式人工智能发展迅速, 在各个领域得到了广泛应用。在教育领域, 大语言模型 (LLM) 等生成式人工智能的引入引起了广泛关注 (Wong 等人, 2024)。目前的研究大多集中于大学生的编程学习 (Sun et al., 2024a), 对小学编程学习的关注较少。这一疏忽凸显了在了解低年级学习者如何从 LLM 等高级人工智能工具中获益方面存在的关键差距 (Zhang & Tur, 2023)。

在编程学习过程中, LLM 有可能减轻学生的认知负担 (Elim, 2024)。以过程为导向的观点关注学生在实际教学和学习过程中如何协调其行为和体验的细节 (Sun et al., 2021)。为此, 本研究采用聚类分析、滞后序列分析和统计方法等多模式学习分析方法, 旨在回答, 1: 高年级小学生在 LLM 辅助编程学习中的典型行为模式是什么? 2: 小学高年级学生的典型行为模式有哪些主要特征?

2. 文献综述

编程教育向小学阶段的延伸已形成全球趋势，研究表明小学生可掌握基础编程技能并完成创造性任务 (Sun & Liu, 2024)。这为早期 STEM 能力培养奠定基础 (Grover et al., 2019)。然而，编码对低龄学习者来说可能具有挑战性 (Pituxcoosuvann & Murakami, 2024)。LLM 为解决这些挑战提供了新途径，其核心优势在于分析编程数据、解析学习者能力特征并识别改进点以提供反馈指导 (Sun et al., 2024a)。与传统编程工具不同，由 OpenAI 训练的大型语言模型 LLM 能够用自然语言与用户互动，这意味着没有任何编程经验的学习者也能从 LLM 的使用中受益。Kashefi 和 Mukerji (2023) 的研究清楚地证明了 LLM 在解决数字问题方面的能力。

为此，本研究通过平台日志、感知数据等多模态数据源，结合聚类分析、滞后序列分析等方法，系统探究小学生 LLM 辅助编程的行为模式与认知特征。

3. 研究方法

本研究于 2023 年秋季在华东地区一所小学开设的必修课程“信息科技”课程中进行。研究采用了混合方法来调查学习者的行为模式和编程聚类。64 名五年级学生 (27 名女生，37 名男生) 参加了为期 10 周的课程，他们都是没有 Python 语言编程经验的新手学习者。

在研究小组的指导和支持下，教师将课程分为三个阶段，每节课学习时间为四十分钟。第一阶段和第三阶段分别进行了前测和后测。学习平台采用了作者研究团队开发的在线编程环境 CodewithAI。CodewithAI 是一个创新平台，集成了大语言模型，学生可以通过点击特定功能的按钮或输入问题，交互式地接收反馈。

本研究从三个方面收集和分析数据。在平台数据方面，我们收集了跟踪学习者执行的编程相关操作的日志文件。根据先前的研究 (Sun et al., 2024a) 和当前的平台数据，确定了平台上分为两种类型的九种编程行为 (见表 1)。

表 1 编程行为描述

行为	编码	说明
与编程相关的行为	平均代码行数(AoC)	每个编程项目的平均编码行数。
	平台运行次数 (NoO)	学习者在所有项目中进行平台操作(即登录、保存)的次数)
	运行点击次数 (NoR)	学习者点击“运行”按钮的次数。
	在 Python 中错误运行 (RPI) 在 Python 中正确运行 (RPC)	学生在运行 Python 代码时收到了错误的反馈。 学生运行 Python 代码，并收到了正确的反馈。
与 LLM 相关的行为	解释 LLM 的控制台输出(ECO)	学员点击“解释控制台输出”按钮，接收来自 LLM 的反馈。
	解释 LLM(ECC)中的当前代码	学员点击“解释当前代码”按钮，接收来自 LLM 的反馈。
	在 LLM (RoQ) 中提出问题	学员向 LLM 提出自己的问题。
	向 LLM (SCC) 提交当前代码	学员将当前代码提交给 LLM，以测试其正确性。

4. 研究发现

4.1 高年级小学生在 LLM 辅助编程中的典型行为模式

针对问题 1（高年级小学生在 LLM 辅助编程学习中的典型行为模式是什么？），我们收集了学生的编程行为数据，并应用滞后序列分析来确定观察到的行为模式的变化。分析结果显示，小学生主要开展三种编程行为：主动编码（AoC）（频率=4443）、执行 Python 代码（NoR）（频率=4198）和处理错误反馈（RPI）（频率=2189）。这表明他们非常关注与编程相关的任务，同时还经常在 Python 代码中遇到错误信息。这些见解凸显了 LLM 在编程过程中促进理解和排除故障的重要作用，展示了其增强年轻学习者学习体验的潜力。

4.2 小学高年级学生在 LLM 辅助编程学习中典型行为模式的主要特征

关于问题 2（小学高年级学生的典型行为模式有哪些主要特征？），我们对所有学习者的编程日志数据进行了聚类分析，以确定行为模式的变化。分析结果显示，小学高年级学习者有四个不同的聚类。

聚类 1 被认定为新手探索者，由 47 名学生组成，占样本总数的 42.18%。在四个聚类中，新手探索者聚类在与编程相关的关键行为中表现出最低的频率：运行 Python 代码 (NoR) ($M=31.02$, $SD=27.90$)、错误运行 Python (RPI) ($M=17.23$, $SD=20.04$) 和正确运行 Python (RPC) ($M=13.79$, $SD=1720.04$)。

第 2 组被认定为高级编程者，由两名学习者组成，占样本总数的 3.12%。这些学生的编码活动水平最高，他们编写的代码行数最多 (AoC) ($M=288.06$, $SD=132.67$)，执行的运行次数最多 (NoR) ($M=297.75$, $SD=133.36$)，运行代码时收到的正确反馈最多 (RPC) ($M=241.50$, $SD=108.17$)。

聚类 3 被认定为积极调试者，包括 4 名学习者，占样本的 6.25%。在四个聚类中，聚类 3 在与编程相关的行为中出现代码运行行为 (NoO) 的频率最高 ($M=34.50$, $SD=11.72$)。他们解释控制台输出 (ECO) ($M=6.25$, $SD=2.38$) 和提交任务进行正确性评估 (SCC) ($M=11.00$, $SD=4.30$) 的频率也最高。

聚类 4 被认定为编程困难者，包括 11 名学习者，占样本的 17.18%。在四个聚类中，聚类 4 在与编程相关的行为中表现出最高的运行代码时收到错误反馈的频率 (RPI) ($M=86.18$, 标准差=33.99) 和最低的平台操作量 (NoO) ($M=17.55$, $SD=6.79$)。

5. 讨论与结论

本研究基于从 CodewithAI 平台收集的数据，发现小学高年级学习者在编码过程中经常遇到困难，主要使用 LLM 解释 Python 编码。这些发现与之前的研究一致，表明年轻学习者经常面临语法错误带来的挑战 (Mladenovic et al., 2018) 与大学生的偏好类似 (Sun et al., 2024a; Yilmaz & Yilmaz, 2023)，当初等学习者在编程过程中遇到困难时，他们就不太可能继续完成后续任务，这大大降低了他们对编程的兴趣 (Iskrenovic-Momcilovic, 2019)。因此，将 LLM 整合到他们的学习环境中，可以提供必要的支持和即时反馈，帮助保持他们的兴趣并提高他们的编程技能 (Zhang & Tur., 2023)。

本研究仍存在一定的研究局限。首先，由于使用了特定版本的 LLM，研究结果的普遍性可能有限，未来的研究应比较多个版本或模型的结果。尽管本次调查存在各种限制，但我们有理由相信，LLM 和类似的人工智能工具有可能成为小学高段学习者的有力辅助工具，从而增强编程教学体验。

参考文献

- Elim, E. H. S. Y. (2024). Promoting cognitive skills in AI-supported learning environments: the integration of bloom's taxonomy. *Education*, 1–11.
<https://doi.org/10.1080/03004279.2024.2332469>
- Iskrenovic-Momcilovic, O. (2019). Pair programming with scratch. *Education and Information Technologies*, 24(5), 2943-2952. <https://doi.org/10.1007/s10639-019-09905-3>
- Kashefi, A., & Mukerji, T. (2023). ChatGPT for programming numerical methods. *Journal of Machine Learning for Modeling and Computing*, 4(2), 1-74.
<https://doi.org/10.1615/jmachlearnmodelcomput.2023048492>
- Mladenović, M., Boljat, I., & Žanko, Ž (2018). Comparing loops misconceptions in block-based and text-based programming languages at the K-12 level. *Education Information Technology*. 23, 1483–1500.
- Pituxcoosuvann, M., & Murakami, Y. (2024). Transforming children's Python Turtle graphics learning with LLM technology: A design proposal. In 2024 9th International STEM Education Conference (iSTEM-Ed) (pp. 1-6). IEEE.
<https://doi.org/10.1109/iSTEM-Ed62750.2024.10663193>
- Sun, D., Boudouaia, A., Zhu, C., & Li, Y. (2024a). Would ChatGPT-facilitated programming mode impact college students' programming behaviors, performances, and perceptions? An empirical study. *International Journal of Educational Technology in Higher Education*, 21(1), 14-22. <https://doi.org/10.1186/s41239-024-00446-5>
- Sun, D., Ouyang, F., Li, Y., & Chen, H. (2021). Three contrasting pairs' collaborative programming processes in China's secondary education. *Journal of Educational Computing Research*, 59(4), 740-762. <https://doi.org/10.1177/0735633120973430>
- Sun, D., Ouyang, F., Li, Y., Zhu, C., & Zhou, Y. (2024b). Using multimodal learning analytics to understand effects of block-based and text-based modalities on computer programming. *Journal of Computer Assisted Learning*, 40(3), 1123-1136.
<https://doi.org/10.1111/jcal.12939>
- Sun, L., & Liu, J. (2024). Effects of gamified python programming on primary school students' computational thinking skills: A differential analysis of gender. *Journal of Educational Computing Research*, 62(3), 846-874. <https://doi.org/10.1177/07356331231225269>
- Wong, L., Park, H., & Looi, C. (2024). From hype to insight: Exploring ChatGPT's early footprint in education via altmetrics and bibliometrics. *Journal of Computer Assisted Learning*, 40(4), 1428-1446. <https://doi.org/10.1111/jcal.12962>
- Yilmaz, R., & Yilmaz, F. G. K. (2023). Augmented intelligence in programming learning: Examining student views on the use of ChatGPT for programming learning. *Computers in Human Behavior: Artificial Humans*, 1(2), 100005.
<https://doi.org/10.1016/j.chbah.2023.100005>
- Zhang, P., & Tur, G (2023). A systematic review of ChatGPT use in K-12 education. *European Journal of Education*, 59, e12599.