基于大模型的智能反馈在初中 Python 编程学习中的应用研究

Research on the Application of Intelligent Feedback Based on Large Language Models in Middle School Python Programming Education

刘美杉[1]，钱逸舟[1*]
[1]江南大学教育学院
[*] yqian@jiangnan.edu.cn

【摘要】 随着信息技术的快速发展和教育改革的不断深化，人工智能在教育领域的应用取得了长足进步，尤其在编程教育领域。本研究结合教育部提出的教育目标，探讨智能反馈在初中 Python 编程学习中的作用，分析学生对反馈的感知如何影响其认知负荷和编程表现。利用通义千问大语言模型，在编程自动评测系统桑田中为学生提供智能反馈，识别代码错误并提出改进建议。研究表明，学生的反馈感知与认知负荷呈显著负相关，高反馈感知的学生在代码改进率和编程测试成绩上表现更佳。线性回归建模得出学生的内在认知负荷是影响编程表现的关键因素。虽然大语言模型能够生成个性化的反馈，但是反馈质量仍有提升空间。
【关键词】 Python 编程；大语言模型；智能反馈；认知负荷；反馈感知

Abstract: *Artificial intelligence (AI) has made significant progress in programming education with the rapid development of information technology and educational reforms. This study explores the role of intelligent feedback in junior high school Python programming and examines how students' perceptions of feedback impact their cognitive load and performance. Using the Qwen large language model, our automated assessment tool Mulberry provides AI-generated feedback for learners, identifying errors and offering suggestions for improvement. Results showed a significant negative correlation between feedback perception and cognitive load, with higher feedback perception linked to better code improvement and test scores. Linear regression modeling indicated that intrinsic cognitive load as key factors influencing performance. Although large language models can generate personalized feedback, there is still room for improvement in the quality of feedback generated by the models.*
Keywords: Python programming, large language model, AI feedback, cognitive load, feedback perception

1.引言

随着信息技术的快速发展和教育改革的深入，中小学人工智能教育取得了显著进展。2024 年，教育部发布通知，旨在提升中小学人工智能教育水平（中华人民共和国教育部，2024）。编程教育作为人工智能教育的基础组成部分之一，有助于培养学生的计算思维和创新意识（张进宝，2019）。Python 因其易懂的语法和与高中信息技术课程标准的高度契合，成为初中编程教育的首选语言。然而，初学者在学习编程时常常面临挑战，其中最为常见的难题之一便是调试代码（Qian & Lehman, 2018）。编程环境提供的错误信息对于非英语母语的初学者而言存在较大的理解难度。面对难以理解的错误信息时，初学者会产生较高的认知负荷，并可能失去学习编程的兴趣（Charles & Gwilliam, 2023）。本研究在自研的编程自动评测系统——桑田系统中整合了基于大模型智能反馈模块，对程序进行自动分析，指出错误原因及修改建议。本研究旨在根据学生对智能反馈的不同的感知程度分析学生认知负荷和编程表现，并评估智能反馈在编程教学中的应用效果。本研究将围绕以下三个问题展开：（1）对智能反馈有不同感知的学生在认知负荷水平上有何差异？（2）对智能反馈有不同感知的学生在编程表现上有何差异？（3）学生对智能反馈的感知是否为影响编程表现的因素？

2.文献综述

2.1.反馈在编程教学中应用的国内外研究现状

在编程教育领域，自动评测工具（Automated Assessment Tools，AATs）为学生提供即时、全自动化的反馈（Messer et al., 2024）。学者们已开发并应用了基于多种编程语言的 AATs，如 Java（Denny et al., 2014）、C++（Pettit & Prather, 2017）以及 Python（Zhou et al., 2021）。然而，以上研究均表明 AATs 对学生编程的成绩未产生显著影响。除了效果方面，AATs 还存在许多局限性。AATs 难以提供足够细节帮助学生解决问题，可能导致沮丧（Messer et al., 2024）。

数据驱动技术依赖大量数据集，数据不足可能限制研究的适用性（Zhou et al., 2021）。本研究采用最新大模型技术，旨在减少这些问题，为学生提供更精准有效的反馈。

2.2.大模型在编程教育领域中应用的国内外研究现状

诸多学者已尝试将大语言模型（Large Language Models，LLMs）应用于编程教学领域，主要功能涵盖生成编程练习（Sarsa et al., 2022）和反馈（Leinonen et al., 2023）。对于 Python 语言，Leinonen 等人（2023）基于 OpenAI 开发的大模型 Codex 进行了自动化反馈的设计与改进，但反馈的质量波动较大，且未在实际教学中验证效果。李秀（2024）指出，学习者使用 ChatGPT 时易产生复制粘贴代码的行为，可能影响其问题解决能力。目前，将大模型用于编程教学的实证研究不多，多数研究关注于工具开发，本研究旨在填补这一空白。为防止学生直接复制代码，研究将利用大模型提供智能反馈，以引导方式指出错误并给出修改建议，帮助学生提高独立思考和解决问题的能力。

3.研究设计

3.1.研究对象

本研究选取无锡市某中学编程社团的 34 名初一学生为研究对象，每周一次 70 分钟的授课，共 12 课时，覆盖一个学期的教学内容。课程内容涵盖 Python 基础，包括输入输出、数据运算、条件语句等。研究团队开发了桑田系统，基于通义千问大模型（Qwen-Turbo-0919）为学生提供个性化反馈，帮助他们改进编程错误。Qwen-Turbo 模型是是通义千问系列模型中速度最快的模型，同时具备较好的性能，适用于编程教学场景。图 1 展示了智能反馈案例。

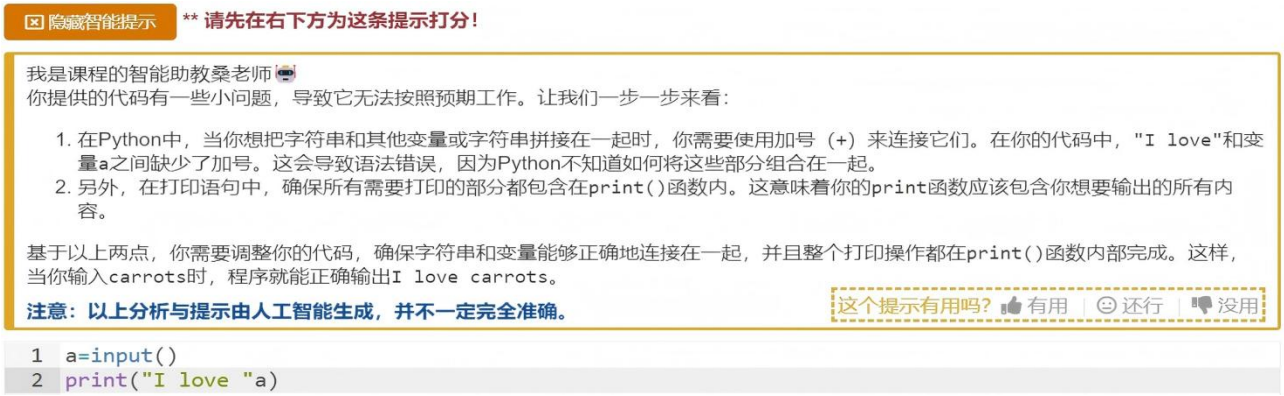

图 1 桑田系统智能反馈案例图

3.2.数据分析方法

本研究通过发放结构化问卷收集学生的认知负荷和对智能反馈的感知。研究依据认知负荷理论，关注内在负荷和外在负荷，采用 Leppink 的认知负荷量表进行数据收集（Leppink et al., 2013）。反馈感知问卷的设计基于技术接受模型，该模型指出个体的实际行为受行为意向的直接影响，而行为意向则由行为态度和感知有用性共同作用决定（高峰，2009）。本研究问卷特别关注反馈态度和反馈有用性两个维度。数据分析使用 SPSS 27.0 软件，进行平均值分析、双变量相关性分析和线性回归建模，旨在为研究的三个核心问题提供实证支持。

4.研究结果

4.1.学生反馈感知的聚类结果

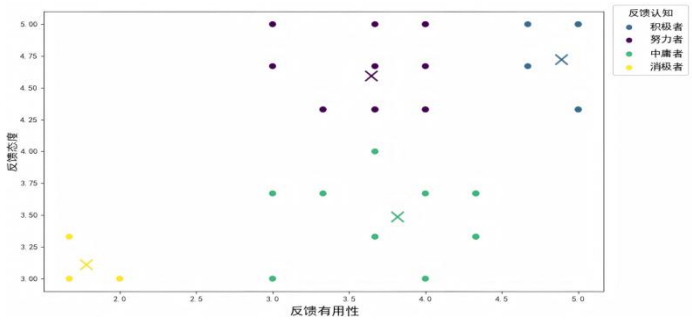

图 2 反馈有用性与态度 K-Means 聚类图

图 2 展示了使用 K-Means 算法聚类结果，平均轮廓系数为 0.486。积极者（6 人）认真且认为反馈有用（有用性=4.89，态度=4.72）；努力者（14 人）认真但反馈有用性一般（有用性=3.64，态度=4.60）；中庸者（11 人）不太认真且反馈有用性一般（有用性=3.82，态度=3.48）；消极者（3 人）不认真且认为反馈无用（有用性=1.78，态度=3.11）。

4.2.学生的反馈感知与认知负荷呈显著负相关关系

问卷数据显示，积极者（内在=2.61，外在=1.5）、努力者（内在=2.74，外在=1.76）、中庸者（内在=2.79，外在=1.91）和消极者（内在=3.78，外在=2.67）的内在负荷和外在负荷依次递增。反馈感知水平高的学生认知负荷较低，反之则高。分析表明，学生对反馈的积极感知与较低的认知负荷显著相关（$p<0.05$）。反馈有用性与内在负荷、外在负荷及总认知负荷显著负相关（$p<0.05$），而反馈态度与外在负荷及总认知负荷显著负相关（$p<0.05$），与内在负荷相关性不显著（$r=-0.311$，$p=0.073$）。

4.3.反馈感知最高的积极者在编程过程中表现出更高的代码改进率和编程成绩

表 1 编程表现平均值分析

	积极者	努力者	中庸者	消极者
代码改进率	38.62%	33.82%	33.47%	32.37%
编程成绩	58.18	53.07	55.22	56.95

表 1 显示，随着反馈感知的提高，代码改进率和认知负荷呈现相似趋势。不过，努力者、中庸者和消极者之间的差异不大。反馈感知与代码改进率的相关性不强（$r=0.187$，$p=0.289$）。反馈感知最强的积极者在代码改进和编程成绩上表现更佳，努力者认真阅读反馈但成绩较低。

4.4.学生的内在负荷是影响编程表现的关键因素

本研究通过线性回归模型分析了学生反馈感知、认知负荷和代码改进率对编程成绩的影响。模型解释了约 24.4%的变异，但实际解释力为 10.8%，较为有限。表 2 显示，内在负荷对成绩有显著负向影响，表明内在负荷是关键因素，但模型解释力不足无法全面解释成绩变化。

表 2 编程成绩线性回归建模

	常量	反馈有用性	反馈态度	内在负荷	外在负荷	代码提升率
β	83.158	-0.170	-2.867	-14.989	8.425	30.332
标准误	40.318	5.696	7.028	5.958	7.961	29.472
Beta	——	-0.006	-0.079	-0.571	-2.516	0.018
t	2.063*	-0.030	-0.408	-2.516*	1.058	1.029

*p<0.05 **p<0.01

5.结语

本研究探讨了大模型智能反馈在初中 Python 编程教学中的应用，旨在评估其对学生认知负荷和编程表现的影响。尽管本研究揭示了智能反馈在编程教育中的潜在优势，但也指出了性能提升的必要性。未来研究应关注智能反馈质量优化，同时探索影响编程表现的其他因素，构建全面模型框架。鉴于研究样本量小、数据稳定性不足，未来研究应改进问卷设计，获取高质量数据支持。最后，学生未充分利用智能反馈可能是因为缺乏有效利用反馈的意识或反馈信息不够清晰，未来教学实践应注重培养学生的反馈意识，提升编程调试能力。

参考文献

高峰. (2009). 教育技术的接受和采纳:几个相关理论的比较. *开放教育研究* (06), 37-41.
李秀. (2024). 生成式人工智能在高中编程教学中的应用探索. *教育传播与技术* (02), 30-37.
张进宝. (2019). 计算思维教育：概念演变与面临的挑战. *现代远程教育研究* (06), 89-101.
中华人民共和国教育部. (2024 年 12 月 3 日). 教育部部署加强中小学人工智能教育. http://www.moe.gov.cn/jyb_xwfb/gzdt_gzdt/s5987/202412/t20241202_1165500.html

Charles, T., & Gwilliam, C. (2023). The Effect of Automated Error Message Feedback on Undergraduate Physics Students Learning Python: Reducing Anxiety and Building Confidence. Journal for STEM Education Research, 6(2), 326–357.

Denny, P., Luxton-Reilly, A., & Carpenter, D. (2014). Enhancing syntax error messages appears ineffectual. Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education - ITiCSE '14, 273–278. https://doi.org/10.1145/2591708.2591748

Leinonen, J., Hellas, A., Sarsa, S., Reeves, B., Denny, P., Prather, J., & Becker, B. A. (2023). Using Large Language Models to Enhance Programming Error Messages. Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1, 563–569.

Leppink, J., Paas, F., Van Der Vleuten, C. P. M., Van Gog, T., & Van Merriënboer, J. J. G. (2013). Development of an instrument for measuring different types of cognitive load. Behavior Research Methods, 45(4), 1058–1072. https://doi.org/10.3758/s13428-013-0334-1

Messer, M., Brown, N. C., Kölling, M., & Shi, M. (2024). Automated grading and feedback tools for programming education: A systematic review. ACM Transactions on Computing Education, 24(1), 1-43. https://doi.org/10.1145/3636515

Pettit, R., & Prather, J. (2017). Automated assessment tools: too many cooks, not enough collaboration. J. Comput. Sci. Coll, 32, 113–121.

Qian, Y., & Lehman, J. (2018). Students' Misconceptions and Other Difficulties in Introductory Programming: A Literature Review. ACM Transactions on Computing Education, 18(1), 1–24.

Sarsa, S., Denny, P., Hellas, A., & Leinonen, J. (2022). Automatic Generation of Programming Exercises and Code Explanations Using Large Language Models. Proceedings of the 2022 ACM Conference on International Computing Education Research - Volume 1, 27–43.

Zhou, Z., Wang, S., & Qian, Y. (2021). Learning From Errors: Exploring the Effectiveness of Enhanced Error Messages in Learning to Program. Frontiers in Psychology, 12, 768962.