

基于生成式人工智能的个性化编程学习系统框架设计

Framework Design for Personalized Programming Learning System Based on
Generative AI徐家乐^{1*}, 田娜¹¹江南大学人文学院

* hiraumi@foxmail.com

【摘要】 生成式人工智能技术的崛起为编程教育带来了全新的可能性。本文提出了一种基于生成式人工智能的个性化编程学习系统框架, 为编程学习提供多层次支持。学生端包括动态学习路径推荐、智能代码分析、自动问题生成以及代码评价等功能, 以满足不同学习阶段学生的个性化需求; 教师端则聚焦于学生学习情况的全面分析与精准指导, 帮助教师高效管理教学活动并提供个性化的支持。

【关键词】 生成式人工智能; 个性化学习; 编程教育; 智能教学系统

Abstract: Generative AI technology has introduced new opportunities for programming education. This paper proposes a personalized programming learning system framework based on generative AI. The student interface includes dynamic learning path recommendations, intelligent code analysis, automatic problem generation, and code evaluation, addressing diverse learning needs. The teacher interface provides comprehensive analysis of student progress and personalized guidance to support efficient teaching management.

Keywords: generative AI, personalized learning, programming education, intelligent tutoring systems

1. 引言

随着生成式人工智能(GenAI)技术的快速发展, 其强大的自然语言理解与文本生成能力使其在教育领域呈现出深度融合的趋势(吴南中等, 2020)。特别在编程教育中, GenAI 可以为学习者提供动态的代码示例、个性化的学习路径以及即时的错误反馈, 从而有效克服传统教学模式中的局限性(Kwak, 2023)。同时, 对于编程教育师资力量薄弱且分布不均的问题, GenAI 作为“智能导师”辅助教师教学、答疑与评价, 是一个前景明朗的解决手段(孙发勤 & 冯锐, 2020)。然而, 如何将 GenAI 高效地应用于编程学习, 构建一个能够满足学习者个性化需求的系统框架, 解决编程课程学习实践缺乏的问题(王晓春等, 2024), 仍值得深入探讨。

本文旨在提出一种基于 GenAI 的个性化编程学习系统框架设计, 重点研究其应用场景, 覆盖编程教育中的学习与评价, 为编程教育迈向智能化和个性化提供理论和实践支持。

2. 个性化编程学习系统的设计目标

本系统框架基于 GenAI 提供的个性化指导, 因而基于易用性、高效性和灵活性设计。

首先, 本系统在传统的线上编程学习系统的基础上, 通过引入 GenAI 满足用户在编程学习中的个性化的需求。本系统借助 GenAI 提供的 API, 可以实现用户与 GenAI 之间的无缝交互, 用户无需进行技术配置, 仅需在对应界面提交问题, 即可快速完成代码分析、解释以及纠正等任务。其次, 本系统对代码分析和纠正这一核心功能进行优化。由于一般的人机交互以诱发式对话为主, 可能会使学生迷失目标(翟雪松等, 2024), 因此通过预先设定高效的提示词模板, 确保 AI 能够快速理解用户需求并生成高质量的答案。用户无需每次手动输入复杂的提

示词，就能直接获得相应反馈。这一设计大幅降低了用户的学习门槛，同时减少了重复操作的繁琐过程。**最后**，本系统在提供通用提示词模板的基础上，设计了灵活的个性化输入机制。对于通用模板无法满足的复杂需求，学习者可以通过自主输入提示词描述问题或提出需求，从而获得高度定制化的支持。这一功能使初学者可以依赖系统默认的提示词模板快速上手，同时使高级学习者通过个人化的提示词获取更深入、精准的反馈，从而提高学习效率。

3. 生成式人工智能应用场景分析

3.1 学生端：学习路径推荐

在传统的学习系统中，大多数学生仅按照系统设定的顺序依次学习，而这种线性的学习方式忽略了学生对知识点掌握程度的差异，无法对学习过程中遇到的重难点内容进行强化。由此，本学习系统引入了**基于 GenAI 的学习路径推荐功能**。该功能能根据学生的实际学习情况，如课程学习进度、学习时长、课后练习的正确率等数据，精准推荐下一步的学习内容。通过这一机制，系统能帮助学生重点攻克学习中的重难点，避免无效重复。

实现上，学生可以在仪表盘界面找到一条**动态的个性化提示语**。这条提示语清晰地指引学生下一步的学习路径。同时系统还在提示语下方提供了多个快捷课程入口，学生可以根据自身需求灵活选择课程并立即开始学习任务。

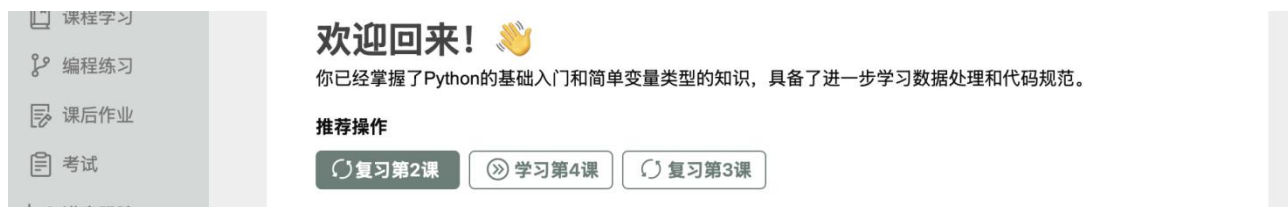


图 1

学习系统学生端仪表盘及推荐语

3.2 学生端：智能代码分析、错误解释和算法优化

学习编程的核心在于实践。在传统的学习过程中，学生通常依赖于 IDLE（集成开发环境）进行编程练习，但这种方式存在三个明显的不足。其一，代码分析通常依赖于学生自行查找问题、修改代码，通过代码运行的结果进行验证。这种方式效率较低，学生很难快速找到潜在问题的根源。其二，代码运行出错时，IDLE 仅提供基础的错误提示信息，这些信息对于初学者而言难以理解。加之有时虽然无运行错误，但程序输出的结果与预期不符，更让学生难以找到问题根源。其三，传统的学习环境并无针对算法效率的优化建议，学生通常只关注代码能否正常运行，而忽视算法的复杂度，导致学生在实际项目中无法有效解决性能问题。

因此，本系统在**代码分析方面**，系统能够快速解析学生的代码结构，标注出潜在的逻辑问题，并简单明了地分析代码的执行逻辑；**错误解释方面**，本系统能对错误信息进行详细解读，结合代码上下文以自然语言提供易于理解的解释，并为学生提供纠正建议或替代实现方案；**算法优化方面**，系统能自动评估学生代码的效率，并提供优化建议。



图 2 学习系统学生端编程界面

在实现层面,如图 2 所示,本学习系统在学生的程序编写界面中设计了一个用于随时调用 AI 的交互控件。该控件允许学生用自然语言提出问题,如“为什么报错”“如何优化算法”等。同时系统会自动收集当前编辑的代码内容等信息,并将这些信息与学生输入的问题一并传递给 AI 进行分析。AI 在接收到问题和代码后,结合上下文快速生成针对性的自然语言回复,包括对问题的简洁而清晰的解答、针对代码的详细分析等,并且必要时附带代码示例。通过这种设计,学生能够在编写代码的过程中实时获得帮助,而无需中断当前任务。

3.3 学生端: 自动问题生成

以往的学生练习问题主要来源于人工编写的纸质习题集或电子题库。学生一旦完成所有习题,就只能反复练习这些已知的问题,难以接触到新的、有针对性的练习内容。重复练习同样的题目容易导致学生从做题转向背题,无法对知识点更进一步地理解和运用。因此,本研究根据学生的学习需求,通过分析学生的知识掌握情况生成多样化且有针对性的练习内容,有效地补充传统习题的不足。这些问题聚焦于学生在实际编程任务的应用,给出问题情境、目标、部分代码以及目标输出的示例,使学生能够清晰地完成这些具体任务。

在实践层面,本系统为学生提供人工编写的题库和由 AI 生成的问题,这样可以保证练习的多样性。人工编写的题库以高质量、规范化的题目为核心;而 AI 生成的问题针对性强,发展了题库的个性化。这样可以拓宽了题目的数量和类型,增强实用性,帮助学生全面提升编程能力。教师通过教师端也可以便捷地管理由 AI 生成的问题,以免产生错误。

3.4 学生端: 代码评价

在编程学习中,代码评价是帮助学生提升编程能力的重要环节。通过系统化的评价机制,学生不仅可以了解自己代码的优缺点,还能够获得针对性的改进建议。但在传统的学习环境中,代码评价通常依赖于教师批改或同学互评,评价过程耗时较长且难以足学生的学习需求。为解决这一问题,本系统引入了基于 GenAI 的代码评价功能。学生可以将自己编写的代码提交给系统进行自动评价,主要评价的维度包括功能正确性、代码质量、算法效率等,同时会为学生生成一段以自然语言生成详细的改进建议等。

3.5 教师端: 学生学习情况分析

传统的教学方法中,教师对学生学习状态的把握依赖于作业反馈及考试成绩,这种方式难以精准反映学生的知识掌握情况。为解决这一问题,本学习系统在教师端引入了学生学习情况分析功能,通过实时收集整理学生的学习数据,如课程完成进度、练习正确率、代码提交次数和学习时长等,为教师提供直观的可视化数据展示,帮助快速了解学生的整体学习状态。同时,系统具备预警功能,可以将学习情况较后的学生报告给教师,提前采取干预措施。

生成式人工智能提示词设计

提示词是用户与 AI 进行交互的桥梁，其质量直接关系到输出效果。良好的提示词设计既能清晰地传递学习者的需求，也能最大限度地提升生成结果的准确性、针对性和实用性。为了减少用户重复输入的操作，高效地使用本系统，本系统针对不同的使用场景设计了多种**预设提示词**，这些提示词根据具体场景的需求精心编写，不仅考虑到了学生和教师在不同任务中的特定需求，还结合了 AI 的语言理解特点，确保提示词能够准确传达上下文信息，减少生成结果的偏差。因此本系统中预设的提示词设计原则有以下四点：（一）**明确性**。提示词的内容必须清晰，确保 AI 能够理解用户需求并生成符合期望的结果。（二）**场景适配性**。针对不同的使用场景，须专门设计不同的提示词模板，使提示词与特定任务需求高度契合。（三）**反馈驱动性**。提示词需要要求 AI 解释问题、分析原因并提出可行的解决方案，而不仅是笼统的答案帮助用户更好地理解 and 解决问题。（四）**引导性**。应考虑避免让 AI 直接给出完整答案导致学生产生依赖（Denny et al., 2024），而是更多地提供分析和建议，帮助用户自主思考问题，鼓励学生深入思考问题本质，锻炼逻辑思维能力和独立解决问题的能力。

针对以上设计原则，下表是在几个应用场景运用的提示词示例。

表 1
提示词模版构建示例表

场景	提示词
路径推荐	我的 Python 语言学习情况：×××。针对以上情况，请给出一句总结性的评价，并在以下内容中选择三个推荐我继续学习的内容。可供选择的学习内容：×××。
代码分析	下面是一段 Python 代码：×××，请用简洁的语言向我描述这段代码的作用，并为代码的关键语句增加注释。
错误解释	下面是一段可能存在错误的 Python 代码：×××。该代码用于实现×××。如果代码没有问题，请直接告诉我。反之，请告诉我代码出错的部分，并告诉我应该从何处修改，但不要直接给出修改后的代码。
算法优化	下面是一段 Python 代码：×××。请为我指出这段代码可以优化之处。
问题生成	请根据知识点×××，生成一道针对 Python 初学者的练习题。该练习题由问题描述、部分不涉及该知识点的代码（可选）、参考输出结果组成。
代码评价	下面是一段用于实现×××的代码：×××，请从功能正确性、代码质量、算法效率这三个维度进行 1-10 的评分，并给出一句话的评语。

总结

本文运用 GenAI 技术设计了一个个性化编程学习系统框架。借助深度学习知识追踪模型和 GenAI 的生成能力，可为学生学习编程知识以及教师教学提供帮助。本文的研究主要集中在框架的理论设计和功能规划层面，尚未开展完整的系统开发和实际应用，因此目前的框架仍需经过实际环境中的检验与优化，以进一步验证其实际效果。GenAI 技术的不断发展为这一框架的落地提供了技术基础，可以预见未来在实际应用中的潜力和影响力。

参考文献

Denny, P., Leinonen, J., Prather, J., Luxton-Reilly, A., Amarouche, T., Becker, B. A., & Reeves, B. N. (2024, March). Prompt Problems: A new programming exercise for the generative AI era. In Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1 (pp. 296-302).

Kwak, M., Jenkins, J., & Kim, J. (2023). Adaptive programming language learning system based on generative AI. Issues in Information Systems, 24(3).

- 厉旭杰,顾雨辰 & 姚持恩.(2024).集成 AI 大语言模型的在线编程实验平台设计与实现.实验技术与管理(08),215-221.
- 孙发勤 & 冯锐.(2020).编程教育中的智能导师系统:架构、设计与应用.远程教育杂志(01),61-68.
- 王晓春,顾小清 & 刘雯.(2024).大学通识编程课程学习障碍及对策.中国大学教学(07),45-51+96.
- 吴南中,夏海鹰 & 黄娥.(2020).课堂形态演进:迈向大数据支持的大规模个性化教学.电化教育研究(09),81-87+114.
- 翟雪松,张丽洁,夏亮亮,徐鑫 & 朱强.(2024).基于 GAI 的逆向工程教学思维在人机协作中的应用研究——以编程教育为例.电化教育研究(09),61-68.