

基于微调 StarCoder2 大模型的“编译原理” 课程实验系统

张友衡, 熊莹, 周放, 劳金赞, 刘茂福

(武汉科技大学 计算机科学与技术学院, 湖北 武汉 430081)

摘要: 针对“编译原理”课程实验教学存在的理论与实践脱节、学生经验不足、编程难度大等问题, 采用渐进式教学方式, 设计并实现了一个基于微调 StarCoder2 大模型的实验教学系统。该系统将课程实验任务进行拆分, 渐进阶梯式分阶段教授, 并利用大模型的语言生成和文本识别能力, 帮助学生逐步掌握编译原理的复杂概念, 提升他们编程和解决复杂问题的能力。该实验系统有利于推动“编译原理”课程的进一步发展完善, 也值得在其他课程进行实践和推广。

关键词: 实验系统; 编译原理课程; 微调; StarCoder2 大模型; 渐进式教学

中图分类号: TP31 **文献标识码:** A **文章编号:** 1002-4956(2025)04-0227-06

计算机专业核心课程“编译原理”, 不仅包含丰富的理论知识, 相关实验也尤为重要, 理论性与实践性极强^[1]。“编译原理”课程对培养学生程序设计能力和问题解决能力有重要作用, 在国内外一流高校教学实践中, 其教学特点都是理论与实践并重^[2-3]。考核方面, 斯坦福大学和卡耐基梅隆大学的实验成绩占比为总成绩的五成及以上, 而且理论教学与实践教学无论在内容上还是进程上都结合得十分紧密。国内高校也十分重视该课程实验, 清华大学根据“编译原理”课程特点和时代背景, 提出使用主流的面向对象设计的 Java 语言进行实验, 并采用案例教学方式, 给学生提供一个教学载体, 在建立初步认识的前提下不断练习, 保持理论联系实际, 并附带图示以建立学生的感性认识^[4]。这些方法能够帮助学生快速跨进实验大门, 但缺乏对学生实验过程的实时帮助, 使学生在课程实验中常常面临理论知识和实际操作上的双重挑战。首先, 编译原理涉及的教学内容十分丰富, 理论较为抽象, 实践难度大^[5], 完成实验任务需要耗费大量时间和精力。北京理工大学调查显示, 八成以上学生在语言学习上存在一定困难, 五成左右学生觉得实验环境很难搭建, 超过四成的学生在实验过程中因心里没底而感到紧张, 进度缓慢^[6]。其次, “编译原理”课程实验通常

采用团队合作形式, 有些学生可能会因为任务复杂而产生畏难心理, 缺乏自信且过度依赖团队核心成员, 而欠缺独立思考, 这不仅会影响个人成长和团队协作, 还会使决策缺乏多样性和创新性, 影响项目进展和学习效果。

随着人工智能技术的进步, 计算机领域正在形成独特的交叉领域^[7], 并逐步影响教育领域变革^[8-9]。大语言模型蓬勃兴起, 基于大语言模型的代码生成辅助工具取得了巨大进展, 代表性的大模型包括 OpenAI 的 GPT-3.5、Google 的 BERT 等^[10]。大语言模型的成功源自自然语言处理领域, 而编译系统本质上是高级程序设计语言信息处理系统, “编译原理”课程实验则是大语言模型自然的应用场景^[11]。这种关联性可以解决学生在学习“编译原理”课程过程中实践经验缺乏、实验案例过少、理论知识与实际操作脱节等问题, 并减轻学生的学习压力, 为学生更好地理解课程内容提供方便。

为了更好地解决学生在“编译原理”课程实验中面临的各种问题和挑战, 有效完成复杂且规模较大的编译系统, 本文设计并实现了一个基于微调 StarCoder2 大模型的“编译原理”课程实验系统。该实验系统旨在通过大语言模型逐步实时引导学生完成编译原理的

收稿日期: 2024-10-31

基金项目: 国家级大学生创新训练项目 (202310488021); 武汉科技大学教研项目 (2023X001)

作者简介: 张友衡 (2003—), 男, 湖北阳新, 硕士研究生, 研究方向为自然语言处理, 1330254543@qq.com。

通信作者: 刘茂福 (1977—), 男, 山东单县, 博士, 教授, 研究方向为自然语言处理, liumaofu@wust.edu.cn。

引文格式: 张友衡, 熊莹, 周放, 等. 基于微调 StarCoder2 大模型的“编译原理”课程实验系统[J]. 实验技术与管理, 2025, 42(4): 227-232.

Cite this article: ZHANG Y H, XIONG Y, ZHOU F, et al. Experimental system for the Principles of Compiler course based on fine-tuned StarCoder2 large language model[J]. Experimental Technology and Management, 2025, 42(4): 227-232. (in Chinese)

各个实验环节,直至编译系统的整体实现。该实验系统为学生提供个性化的学习支持,有利于培养学生软件开发的设计、实现、优化、验证、编程等多方面能力^[12-13]。今后这种基于人工智能的教育技术将会在更广泛的领域得到应用,为提高教育质量和学习体验做出更大贡献。

1 实验系统分析与设计

1.1 系统架构

基于微调 StarCoder2 大模型的“编译原理”课程实验系统,针对复杂度较高的“编译原理”课程实验,旨在为学生提供更高质量的学习和实验体验,通过科学的系统设计,帮助学生更有效地掌握相关知识和技能。

在实验初期,系统会提供详尽的实验资料,帮助学生快速了解实验要求和背景知识。进入基础知识验证阶段,系统会在数据层调用数据库里存储的选择、填空等题目,采用随机抽取方式对学生进行测试,目的是引导学生加深对理论知识的理解。系统会将学生的测试结果进行实时批阅,并及时给出反馈,以便学生了解自己的学习进度与不足,从而针对性地进行自

我提升。

在理论知识掌握之后,学生就可以进入到“编译原理”课程的核心实验验证阶段。在此阶段,StarCoder2 大模型会以悬浮窗形式在学生页面提供辅助,学生在进行编译实验过程中,可以向 StarCoder2 大模型提出问题,并将收到专业且具针对性的解答,以便让学生更好地突破自身瓶颈,提升自身实力。

该系统不仅在理论学习与实践操作之间架起了一座桥梁,并且能够通过智能化、个性化辅助,提升“编译原理”课程实验的教学质量和学习效果。系统服务能够提供实验资料、理论知识、渐进式实验引导、实时反馈和个性化编译指导。

系统架构由业务系统、用户交互和大模型生成三部分组成,涵盖了系统为学生提供学习服务的具体流程,重点体现在大语言模型与学生的交互上。具体系统架构图如图 1 所示。

1.2 渐进式实验教学

渐进式教学是一种持续学习的实现方式^[14],将这种方式应用在“编译原理”课程实验中具有显著好处。该课程的复杂性使学生在一次性接触所有知识时可能会感到压力巨大,通过渐进的逐步引导方式有助于

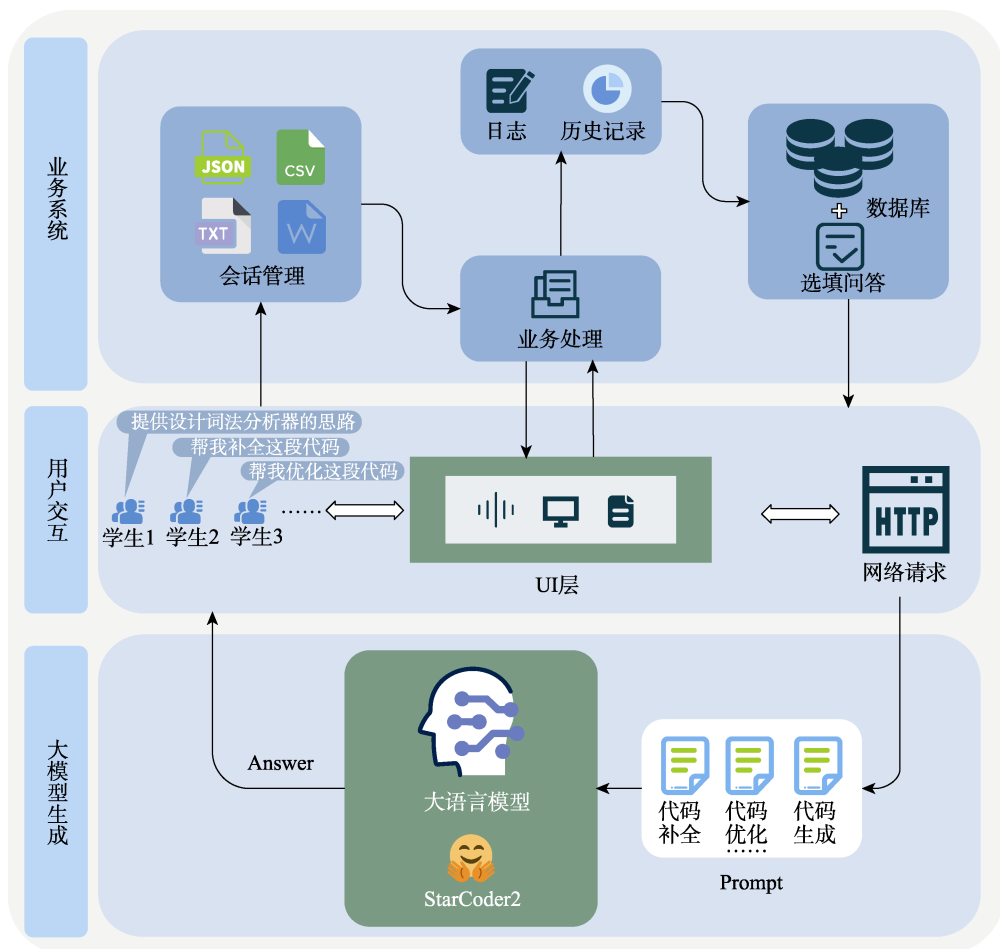


图 1 “编译原理”实验系统架构图

降低这种压力, 让学生更轻松地进行学习。通过渐进阶梯式分段的教学设计, 学生能够在各个阶段建立扎实的知识基础, 并在此基础上不断深化理解, 最终成功实现编译系统的设计与实现。结合 StarCoder2 大模型的个性化辅导, 可以针对学生在每个阶段的表现进

行调试, 帮助他们克服学习中的瓶颈。这种组合不仅可以提高学生的学习效果, 还可以促进学生主动学习和主动探索精神, 在掌握复杂编译原理知识过程中, 积累宝贵的实践经验, 进一步提升综合素质和实际操作能力。具体实验内容如表 1 所示。

表 1 实验内容

实验阶段	内容描述	目标
阶段一: 选填练习	基础知识的选择填空题练习	掌握基本的学科知识和实验要求
阶段二: 函数体练习	先将重要的函数部分逐个实现	掌握函数体的写法和基本原理, 为后续实验内容做铺垫
阶段三: 词法分析	实现词法分析器, 识别基本的语言构造	掌握词法分析的基本原理和实现方法
阶段四: 语法分析	实现语法分析器, 解析语言的语法结构	理解语义分析的原理, 能够编写简单的语法分析程序
阶段五: 语义分析	拓展语义分析器, 语义检查和错误处理	掌握语义分析和错误处理技术, 提高代码的正确性和可行性
阶段六: 中间代码优化	对生成的代码进行优化, 提升性能	掌握基本的代码优化技术, 提高编译器生成代码的执行效率
阶段七: 综合实验	结合所有阶段完成一个简单的编译系统	理解编译系统的整体架构和工作流程

“编译原理”课程的验证性实验环节内容多、程序复杂^[15], 学生做实验时常感到无头绪且缺乏信心。在实验环节采用渐进式教学的具体作法是, 先通过选填练习让学生掌握基本的学科知识和实验要求, 然后在正式进入验证性实验环节前加入一个函数体练习阶段, 该阶段通过将复杂繁琐的综合问题拆解为多个有串联关系的简单函数问题, 逐个进行实现, 从而降低练习难度, 增强学习信心和获得感, 使整个学习过程更加顺畅和有效。

1.3 微调 StarCoder2 大模型

StarCoder2 及其前身 StarCoder 都由 Bigcode 社区推出。StarCoder2 大模型是一种利用预训练技术的语言生成模型, 具有强大的自然语言处理与生成能力, 支持多种编程语言, 在几种低资源语言上, StarCoder2 的数学和代码推理能力均优于高资源语言代码完成方面表现最佳的 DeepSeekCoder 模型^[16], 与“编译原理”课程实验应用有着极高的契合度。

为将 StarCoder2 大模型更好地应用于“编译原理”课程实验, 需要通过微调提升模型的适用性和准确性。模型微调是指在特定任务上对预训练模型进行进一步训练, 以提高其性能与任务适配度的过程。目前的微调方式主要包括数据微调、知识提示学习和基于特定指令的微调。

数据微调是最为常见的一种微调方式, 通过收集与编译原理相关的高质量标注数据, 如词法分析器的标注数据集、语法分析器的语法规则数据、中间代码生成与优化的案例数据, 让模型进一步学习这些领域的特定知识。这些数据可以从已有的编译原理实验数据集中提取, 或者通过人工标注方式生成。还可以通过学生的实验数据、作业数据等进行数据扩充, 使模型更能适应教学环境的多样化需求。

知识提示学习可以增强模型的推理能力。在“编译原理”课程中, 涉及大量的编译器工作原理及其逻辑, 通过将这些知识体系化, 引入知识图谱, 可以让模型在推理过程中更加准确地理解编译步骤之间的关系, 并在生成答案时提供更为精确的解释。

基于特定指令的微调也是一种有效的微调方式。编译原理实验任务通常都具有较高的技术性和复杂性, 可以通过设计特定的指令集, 引导模型在处理这些任务时遵循一定的规则。指令集可以包含词法分析规则、语法树生成步骤、代码优化策略等。通过在微调过程中强化这些指令, 能够使模型更好地按照课程实验要求完成任务。

对于 StarCoder2 大模型的微调过程主要是数据微调, 微调流程图如图 2 所示。

StarCoder2 微调过程从特定任务数据开始, 包括标注好的词法分析、语法分析和中间代码优化数据, 这些数据被输入模型并通过嵌入层转换为嵌入向量 h , 以便模型理解和处理。微调的核心是对模型的预训练权重 W 进行低秩适应调整, 其中基础矩阵保留了模型的通用知识, 而增量矩阵则专注于捕捉编译原理任务的特定特征, 如词法规则、语法树生成步骤和代码优化策略。通过将增量矩阵和基础矩阵相加, 新模型能够更好地适应编译原理课程的下游任务, 如词法分析器和中间代码优化。微调后的 StarCoder2 大模型在课程实验系统中与学生交互, 提供更精确的任务指导。随着学生实验数据和作业反馈的加入, 微调过程可以不断迭代优化, 使模型能够在多样化的教学场景中表现出更高的适应性和准确性。

通过与微调 StarCoder2 大模型的交互和渐进式教学, 能够使学生更深入地理解编译程序的工作机理^[17], 提高课程兴趣, 并逐步提升编程能力和问题解决能力。

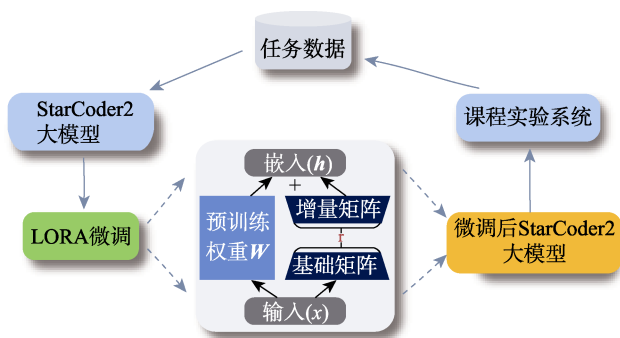


图2 微调流程图

1.4 实验流程

在函数体编程练习之前,教师设立线上答卷,通过选择题和填空题让学生了解编译原理相关基本概念和相关实验工具,帮助学生巩固理论知识,构建扎实的知识基础,为下一步实验做好铺垫。之后进入函数体练习环节。在该环节学生需要逐个实现每个函数体,再将各函数体串联在一起,以便在程序设计模块中获得阶段性成果。为了高质量完成实验内容,系统通过微调后的 StarCoder2 大模型能够提供代码自动生成和代码优化能力,并给出个性化反馈,以便帮助学生更好地实现预定目标,进一步提高学生实验的完成度和学习效果。具体实验流程图如图3所示。

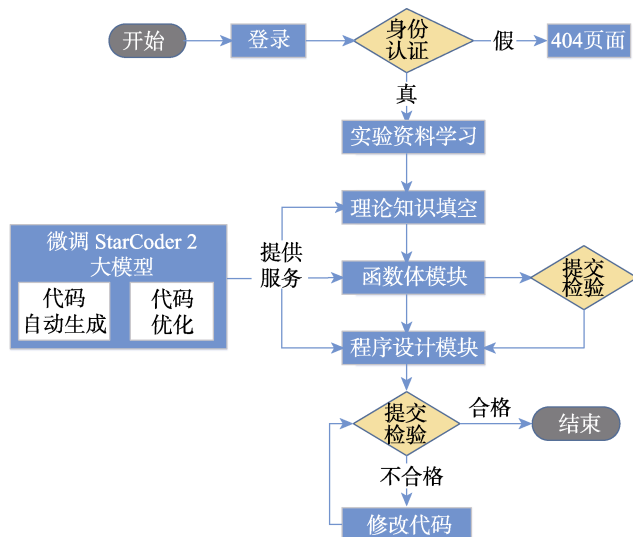


图3 实验流程图

2 实验系统效果

图4展示了编译原理实验场景下 StarCoder2 大模型交互应用实例。

从图4可以看出,“学生1”基础较弱,对实验要求完全没能理解,所以他询问了大模型整体的编程思路;“学生2”较“学生1”知识基础好,但编程基础

学生1	学生2	学生3
问题:请提供一下设计词法分析器的编译思路	问题:请帮我补全语法分析器中解析数字部分的代码。 #include <cctype> double lexer_number(Lexer *lexer) { double value = 0.0; double fraction = 0.0; int fraction_count = 0; // 循环读取字符,直到遇到非数字字符为止 while (std::isdigit(lexer->current_char())) { value = value * 10.0 + (lexer->current_char() - '0'); // 累加整数部分 lexer->next_char(); // 移动到下一个字符 }	问题:请给出下面这段关于字符处理代码的优化结果。 void next_character() { current_character = fgetc(input_file); if (current_character == '\n') {line_number++;} }
大模型应答结果如下: 1、输入字符串;同样地,开始需要将源代码作为字符序列进行读取。 2、标记(Tokenization);与前面的解释相同,创建一组规则去匹配和区分源代码中的不同元素,如标识符、数字、关键词、运算符等。根据这些规则将原始源码转化为一个token列表,其中每一个 token 都代表特定类型的单词或符号。 3、过滤注释;检查源代码是否存在任何注释,并从标记列表中移除它们,因为对于后续的分析来说,注释是无关紧要的。 4、错误检测;在这个过程中,查找那些不遵循编程语言规则的片段,如拼写错误、缺少必需的符号或结构等。当遇到违规情形时,报告相应的错误信息并终止分析流程。 5、输出结果;将经过处理后的tokens集合返回至下一阶段的解析器(parser或syntax analyzer),它将进一步检查该代码是否按照正确的顺序排列并且拥有有效的语义。	大模型应答结果如下: #include <cctype> ... // 如果接下来是小数点,则读取小数部分 if (lexer->current_char() == '.') { lexer->next_char(); // 移动到下一个字符 while (std::isdigit(lexer->current_char())) { fraction = fraction * 10.0 + (lexer->current_char() - '0'); // 累加小数部分 fraction_count++; // 记录小数部分的位数 lexer->next_char(); // 移动到下一个字符 } // 计算小数部分的值 value += fraction / std::pow(10.0, fraction_count); return value;} }	大模型应答结果如下: void next_character() { current_character = fgetc(input_file); line_number += (current_character == '\n') ? 1 : 0; // 使用条件表达式优化 } 优化建议:使用条件表达式简化行号更新的逻辑,减少了if语句的使用,使代码更简洁。可以考虑添加对EOF的处理,以防止在到达文件末尾后无法正确处理字符。

图4 大模型交互应用实例

较差,所以需要大模型帮助其补全代码;“学生3”编程基础最好,可以完整地写出符合要求的函数程序,为了更加精益求精,他将自己写好的代码放入模型中,请求大模型帮助他优化代码,以提高代码效率。由此可见,对于编程基础参差不齐的学生,微调 StarCoder2 大模型可以同时面对,并对他们进行有针对性的个性化指导。在传统的实验教学模式下,当实验教师面对多个学生同时提问时,只能选择一个学生的问题进行解答,其他学生只能默默等待,与之相比,微调 StarCoder2 大模型使教学互动效率有了飞跃式提高。

学生在与 StarCoder2 大模型进行交互过程中,不仅能够获得实时个性化的学习指导,在基础概念和理论知识学习方面能够查漏补缺,并且这种帮助可以贯穿整个实验过程,从理论知识到函数体实现,从词法分析器、语法分析器的设计完成到完整编译器的实现,乃至综合项目的设计与开发。这些项目可以是与编译原理相关的设计与开发,也可以是实际的编译场景。大模型与学生如影随形,不仅能提升学生的实际学习效果和学习兴趣,还能促进“编译原理”实验课程的教学改革和创新。

3 结语

在“编译原理”课程实验教学设计中,渐进式的教学方式,能够使学生在学习过程中逐步掌握复杂的编译原理概念和技能。微调 StarCoder2 大模型的引入不仅提高了学生的学习兴趣 and 积极性,还能够为学生提供个性化的指导和支持,提升了学生的编程能力和问题解决能力。在此实验系统实践过程中,学生通过与微调 StarCoder2 大模型的互动,能够更好地理解编译过程的各个阶段,并从容而高质量地完成实验任务,增强了对编译原理的全面理解。在已取得的积极成果基础上,下一步教学团队将尝试用不同的微调方式进一步优化 StarCoder2 大模型,使其能够给学生提供更具体、详细的指导。同时,推进最新技术与“编译原理”课程教学融合,不断丰富教学资源,开发更多与编译原理相关的实验项目和案例,涵盖不同难度、不同应用场景,满足学生的多样化需求。

参考文献 (References)

- [1] 余月,李凤霞,陈宇峰,等.计算机编译原理课程虚拟实验设计与实践[J].实验技术与管理,2019,36(8):123-126.
YU Y, LI F X, CHEN Y F, et al. Design and practice on virtual experiment of Computer Compilation Principle course[J]. Experimental Technology and Management, 2019, 36(8): 123-126. (in Chinese)
- [2] 张莉,蒋竞.编译课程建设思路及成效[J].计算机教育,2024(5):25-30.

- ZHANG L, JIANG J. Thoughts and outcomes on the Construction of Compiler course[J]. Computer Education, 2024(5): 25-30. (in Chinese)
- [3] 计卫星,王贵珍.基于开源社区的编译原理课程构建[J].中国大学教学,2021(增刊1):46-53.
JI W X, WANG G Z. The Construction of the Principles of Compiler course based on open source communities[J]. China University Teaching, 2021(S1): 46-53. (in Chinese)
- [4] 王朝坤.基于Java的编译原理课程案例教学方法初探[J].计算机教育,2011(11):48-51,60.
WANG C K. An exploration of cases teaching method for the Principles of Compiler course based on Java[J]. Computer Education, 2011(11): 48-51, 60. (in Chinese)
- [5] 于建伟,李旭伟,吴芝明.编译原理演示模块的设计与实现[J].实验室研究与探索,2018,37(9):138-142,180.
YU J W, LI X W, WU Z M. Design and implementation of compiler theory demo module[J]. Research and Exploration in Laboratory, 2018, 37(9): 138-142, 180. (in Chinese)
- [6] 计卫星,王贵珍,李佩.编译原理语言认知实验设计与实践[J].计算机教育,2019(8):94-97.
JI W X, WANG G Z, LI K. Design and practices of the language cognition experiments for the Principles of Compiler course[J]. Computer Education, 2019(8): 94-97. (in Chinese)
- [7] 张科,张铭,陈娟,等.计算机教育研究浅析——从ACM计算机科学教育大会看国内外计算机教育科研[J].中国计算机学会通讯,2019,15(4):16-26.
ZHANG K, ZHANG M, CHEN J, et al. Design and practices of the language cognition experiments for the Principles of Compiler course[J]. Chinese Computer Society Newsletter, 2019, 15(4): 16-26. (in Chinese)
- [8] 李清勇,耿阳李敖,彭文娟,等.“私教”还是“枪手”:基于大模型的计算机实践教学探索[J].实验技术与管理,2024,41(5):1-8.
LI Q Y, GENG Y L A, PENG W J, et al. Tutor or impostor: Exploring computer practice teaching based on large language models[J]. Experimental Technology and Management, 2024, 41(5): 1-8. (in Chinese)
- [9] 张金,宫晓利,高小鹏,等.基于通用大语言模型的计算机系统创新实验设计[J].实验技术与管理,2024,41(10):1-9.
ZHANG J, GONG X L, GAO X P, et al. Innovative experimental design of computer systems based on general large language models[J]. Experimental Technology and Management, 2024, 41(10): 1-9. (in Chinese)
- [10] 胡星,夏鑫,王千祥.代码大语言模型的实践与挑战[J].中国计算机学会通讯,2024,20(8):19-23.
HU X, XIA X, WANG Q X. Practices and challenges of code large language models[J]. Chinese Computer Society Newsletter, 2024, 20(8): 19-23. (in Chinese)
- [11] 刘茂福,毕健旗,熊莹,等.编译原理与自然语言处理的实验对齐研究[J].计算机教育,2020(3):36-42.
LIU M F, BI J Q, XIONG Y, et al. Research on the experimental alignment of compilation principles and natural language

- processing[J]. Computer Education, 2020(3): 36–42. (in Chinese)
- [12] 张洪滨, 周旭林, 邢明杰, 等. AutoConfig:面向深度学习编译优化的自动配置机制[J]. 软件学报, 2024, 35(6): 2668–2686. ZHANG H B, ZHOU X L, XING M J, et al. AutoConfig: Automatic configuration mechanism for deep learning compilation optimization[J]. Journal of Software, 2024, 35(6): 2668–2686. (in Chinese).
- [13] 史涯晴. 突出编程能力培养的编译原理课程教学改革[J]. 计算机教育, 2022(9): 105–108. SHI Y Q. Teaching reform of the Principles of Compiler course to highlight programming ability cultivation[J]. Computer Education, 2022(9): 105–108. (in Chinese)
- [14] 田莉, 郝雯娟, 王志凌. 四旋翼飞行器创新实验平台的渐进式教学实践[J]. 实验室研究与探索, 2023, 42(10): 165–169, 210. TIAN L, HAO W J, WANG Z L. Progressive teaching practice for innovative experimental platform based on quadrotor aircraft[J]. Research and Exploration in Laboratory, 2023, 42(10): 165–169, 210. (in Chinese)
- [15] 孙守卿, 鞠传香, 刘晓红. 计算思维培养为核心的编译原理实验教学实践[J]. 电脑知识与技术, 2018, 14(34): 90, 93. SUN S Q, JU C X, LIU X H. Experimental teaching practice of compilation principle with computational thinking cultivation as the core[J]. Computer Knowledge and Technology, 2018, 14(34): 90–93. (in Chinese)
- [16] LOZHKOV R, LI R, ALLAL L B, et al. StarCoder 2 and the stack v2: The next generation[EB/OL]. arXiv, 2024, <http://dx.doi.org/10.48550/arXiv.2402.19173>.
- [17] 杨奎河, 赵玲玲. 面向专业认证的编译原理课程分析评价及改进[J]. 福建电脑, 2021, 37(7): 158–159. YANG K H, ZHAO L L. Analysis evaluation and improvement of professional certification oriented Compiler Principles course[J]. Journal of Fujian Computer, 2021, 37(7): 158–159. (in Chinese)

Experimental system for the Principles of Compiler course based on fine-tuned StarCoder2 large language model

ZHANG Youheng, XIONG Ying, ZHOU Fang, LAO Jinzan, LIU Maofu

(School of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan 430081, China)

Abstract: [Objective] The rapid development of large language models (LLMs) has significantly advanced code generation tools. To tackle issues such as the gap between theory and practice, lack of experience, and the high complexity of programming in the experimental teaching for the Principles of Compiler course, this study designs and implements an experimental teaching system powered by a fine-tuned StarCoder2 LLM. The system offers personalized learning support during experimental sessions in the course, helping students develop skills in design, implementation, optimization, verification, and programming. [Methods] This paper designs and implements an experimental teaching system based on fine-tuned StarCoder2 LLM with a progressive teaching approach, allowing students to build a solid foundation in each stage of learning. By processing step-by-step through the design and implementation stages, students can deepen their understanding of compiler systems. The StarCoder2 model is fine-tuned specifically for the Principles of Compiler course, with programming tasks and theoretical concepts tailored for this purpose. This fine-tuning ensures the model specialization and effectiveness in code generation and explanation, enabling real-time, personalized support for students at each stage of their learning process. [Results] This progressive teaching approach helps students gradually grasp complex concepts and skills related to the principles of the compiler course. The fine-tuned StarCoder2 LLM not only increases students' engagement and motivation but also offers tailored guidance and support, enhancing their programming and problem-solving abilities. Through interactions with the fine-tuned model, students achieve a better understanding of the compiling process through interaction, completing experimental tasks more efficiently. This elevates their overall comprehension of compiler principles. [Conclusions] By designing and implementing this experimental teaching system based on the fine-tuned StarCoder2 LLM, this paper effectively addresses common challenges in the teaching of compiler principles. The progressive teaching approach deepens students' understanding of the subject while enhancing their programming skills and problem-solving abilities. This teaching system presents opportunities and ideas for the principles of the compiler course and fosters further development and improvement. It is worthy of broader application and promotion in other courses as well.

Key words: experiment system; Principles of Compiler course; fine-tuning; StarCoder2 large language model; progressive teaching approach

(编辑: 张文杰)