

大语言模型技术赋能Spark编程教学实践

张淼, 李志飞, 张冀

湖北大学计算机学院, 湖北武汉

摘要: 随着大数据与人工智能技术的深度融合, 大语言模型的涌现为编程教育模式革新提供了新路径。本文聚焦Spark编程教学场景, 首先阐述了大语言模型技术特点及其在教育教学中的应用, 详细分析了当前Spark大数据编程课程教学中的现状和面临的挑战。接着提出LLMs融入Spark大数据编程教学的创新设计思路, 将LLMs作为辅助工具与传统教学模式相结合。最后, 介绍具体的大模型赋能Spark大数据编程教学实践与效果。

关键词: 大语言模型; Spark编程; 教学应用; 教学创新

Large Language Modeling Technology Empowers Spark Programming Teaching Practices

Miao Zhang, Zhifei Li, Yan Zhang

School of Computer Science, Hubei University, Wuhan, Hubei

Abstract: With the deep integration of big data and artificial intelligence technology, the emergence of large language models (LLMs) offers a new avenue for revolutionizing programming education. This paper focuses on the teaching of Spark programming, beginning with an exploration of the technical features of LLMs and their applications in education. First this paper provides an in-depth analysis of the current state and challenges in Spark big data programming courses. Building on this, this paper proposes innovative approaches to integrating LLMs into Spark programming education, positioning them as auxiliary tools to complement traditional teaching methods. Finally, this paper showcases practical applications and the impact of using LLMs to enhance Spark programming education.

Keywords: Large Language Models; Spark programming; Teaching applications; Teaching innovations

1 引言

随着大数据技术的快速发展, Spark作为分布式计算框架, 凭借其高效的内存计算能力和丰富的生态系统, 已成为大数据处理领域的核心技术之一。掌握Spark编程技能不仅是计算机专业学生的必修内容, 也是企业招聘大数据工程师的重要考量标准。然而, 与传统编程语言相比, Spark编程的复杂性显著提升, 其涉及的并行计算、分布式存储和任务调度等高级概念, 使得教学难度大幅增加。在当

前的Spark编程教学中, 学生普遍面临代码编写难度大、调试过程复杂以及学习曲线陡峭等挑战, 而传统的教学模式难以满足学生的个性化学习需求, 导致教学效率和学习效果亟待提升。

与此同时, 生成式大语言模型 (Large Language Models, LLMs) 技术的突破为教育领域带来了新的可能性[1-3]。以DeepSeek、ChatGPT、文心一言为代表的生成式人工智能工具, 展现出强大的代码生成、逻辑推理和自然语言交互能力。研究

表明, LLMs能够辅助教师快速生成教学案例、提供个性化学习路径, 并通过实时答疑和代码纠错降低学生的学习门槛。在Spark编程教学中, 合理引入大语言模型, 结合传统教学模式, 不仅可以提高学生的代码理解与编写能力, 还能有效减轻教师的教学负担, 优化课程质量, 为编程教学提供新的创新思路[4]。

2 大语言模型及教育教学应用

LLMs作为深度学习技术构建的生成式人工智能系统, 其核心在于通过大规模预训练和微调, 实现对自然语言的理解与生成。凭借其强大的上下文理解能力、代码生成能力和多模态交互能力, 正在深刻变革教育领域的教学模式和学习方式[5]。

2.1 学生个性化学习支持

LLMs能够基于学生的学习数据, 如学习进度、知识盲点和编程风格等, 精准提供个性化的学习路径和资源推荐。例如, 在Spark编程教学中, LLMs可以智能化地为初学者推荐基础的API调用练习, 而对于已经具备一定编程基础的学生, 则可引导其深入探索分布式任务调度与性能调优等更具挑战性的高级任务。通过这种个性化教学模式, Spark编程课程能够更好地满足不同学生的学习需求, 显著提升学习的适配性与效果。

2.2 编程案例学习

LLMs能够实时解答学生在编程过程中遇到的问题, 并提供详尽的错误分析与优化建议[6]。例如, 当学生在运行Spark作业时遇到Task not serializable错误, 可以辅助解释错误的成因, 还能根据学生的需求提供切实可行的代码优化建议。利用这种交互式的智能答疑机制, 可以有效降低学习门槛, 提升学生的学习效率, 为编程学习提供强有力的支持。

2.3 教师教学案例准备

LLMs能够根据具体的教学需求, 快速生成多样化的教学案例, 为教师教学内容的设计提供方便[7, 8]。例如, 在Spark SQL教学中, LLMs可以基于真实数据集自动生成多层次的查询任务, 从

简单的筛选与聚合操作到复杂的多表联结与窗口函数应用。借助LLMs的生成能力, 特别联网搜索增强能力, 可以方便快捷地为教师提供丰富的教学资源作为参考, 帮助极大地提升教学准备效率和多样性。

2.4 教学评估与反馈

LLMs能够帮助教师对学生的交互数据进行记录与分析, 以评估其学习进展与理解程度, 并识别学习过程中存在的困难。基于此, 教师能够快速地了解全面掌握学生的学习情况与需求, 进而优化教学策略, 提供针对性的反馈与支持, 实现更精准的教学干预与指导。

综上所述, LLMs在编程教学中具备广泛的应用优势。其代码生成、案例学习、个性化学习支持以及教学资源生成能力, 使Spark编程课程更加智能化和高效化。然而, 在实际应用中, 还需要结合传统教学方法, 确保学生在获取便利的同时, 仍能保持编程思维训练, 以实现更优的教学效果。

3 Spark编程教学现状及挑战分析

Spark作为大数据计算领域的核心框架, 凭借其强大的分布式计算能力和高效的数据处理性能, 在学术研究和工业实践中得到了广泛应用。然而, 与传统编程语言相比, Spark的学习门槛相对较高, 其涉及的数据处理范式以及计算任务优化等内容较为复杂, 使得教学难度大大增加。在当前的Spark编程教学体系中, 教学方式主要依赖于课堂讲授与实验实践相结合的模式。尽管这种方式在一定程度上能够帮助学生掌握Spark基础概念, 但仍面临诸多挑战, 例如学生对复杂概念的理解不足、实践能力的培养有限等问题, 从而影响了整体教学效果和学习体验[9-11]。

首先, Spark编程的核心在于分布式系统的实践, 但当前教学常陷入“重理论、轻实操”的误区。一方面, 分布式计算原理的抽象性使学生难以直观理解。例如, 在讲解RDD的惰性求值时, 教师通常通过流程图展示DAG的构建与触发机制, 但由于缺乏对任务调度过程的实际操作和感知, 学生往往停留在“知其然不知其所以然”的阶段。另一方面, 实验设计过于简化, 多数高校受限于硬件资

源，无法模拟TB级数据的处理场景，学生缺乏真实环境下的实操经验。

其次，课程通常面向高年级本科生开设，学生背景差异显著。部分专业学生具备扎实的编程能力和分布式系统知识，而另一部分学生仅掌握基础编程技能。传统“一刀切”的教学模式难以兼顾不同层次学生的需求。例如，在讲解Spark SQL优化时，进阶学生希望深入探讨Catalyst优化器的执行计划调优，而初学者在学习基本SQL语法上挣扎。教师若仅按平均进度授课，容易造成“优生吃不饱，差生跟不上”的局面。

此外，当前Spark课程的考核方式主要还是依赖理论考试与固定实验报告，评估维度单一。理论考试侧重概念记忆，难以考察分布式编程能力；实验报告则极易流于形式，其质量良莠不齐，而教师也很难有效评估其代码质量与优化意识。常常出现的一个状况是，尽管大多数学生实验报告得分超过80分，但在后续项目答辩中，仅小部分学生展现了独立设计并优化完整Spark作业的能力。

4 大语言模型赋能Spark编程教学设计

LLMs对于教学实践的赋能体现在“教师、学生和人工智能”的教学互动关系上，利用交互式的大模型生成模式，提升教学质量与学习效率，优化传统教学模式，激发学生的学习兴趣，并培养他们利用LLMs解决实际编程问题的能力[12,13]。

(1) 借助编程案例，实践LLMs解决编程问题的应用流程。例如，Spark课程中最常见的案例“计算一个文本文件中单词的出现频率”，只需输入类似“使用Spark 统计文本文件中的词频”这样的指令，即可通过与大模型的交互逐步拆解任务需求，生成包含map、reduceByKey及collect操作的完整代码，并附带详细的注释解释代码逻辑，帮助学生理解API调用方式及底层执行机制。

(2) 小组协作，LLMs辅助分工与协同。借助LLMs可重构传统小组合作模式，实现智能化任务分配与协同编码。例如，根据小组内所有成员的能力描述（如SQL熟练度、并行编程经验）自动划分角色（如数据预处理、核心算法实现、性能调优），并为每个角色生成定制化任务清单，进而通过协作软件实现合作编程，共享代码片段，确保小组合作

的公平性和高效性。

(3) 反思分享，借助LLMs实现自主纠错分析。构建基于LLMs的反思性学习闭环是提升教学效果的关键。对于每次实验，要求学生使用大模型对编程过程进行复盘：输入原始需求、自编代码及运行结果，生成一份模型提供的代码评审报告，涵盖风格规范、算法效率及异常处理等多维度评估。教师则可以选取学生实验过程出现的典型案例进行对比，组织学生讨论不同编程范式在可维护性、执行效率方面的权衡，进而有效培养学生的元认知能力。

(4) 产学研合作实验项目实践。通过与行业内企业共建实验项目，构建“真实问题-大模型辅助-产业验证”的实践体系。例如，基于某云计算平台“千万级物联网设备数据分析”场景，企业提供包含GPS坐标、传感器读数等字段的脱敏设备日志，学生利用Spark结合LLMs完成从数据清洗、特征工程、异常检测全流程开发。在此过程中，大模型可以辅助生成PySpark代码框架，提供不同技术方案的对比如，帮助学生深入理解实际问题的解决思路与行业标准。

(5) 多维度教学评价，关注学生过程性成长。传统的Spark课程评估主要基于理论考试和固定实验任务，难以全面衡量学生的编程能力。借助LLMs，可以通过实验报告、代码质量及交互数据构建智能化、多维度的教学评价体系，涵盖代码质量分析、风格优化、性能调优能力等多个方面。同时通过分析学生的交互提问序列，可以评估其编程能力的发展情况，追踪其能力提升的趋势，智能化记录学生的过程性成长，为个性化教学提供数据支持。

(6) 教学效果反馈。在教学管理方面，LLMs可以辅助教师监测学生的学习情况，并提供数据分析支持。通过分析学生的代码提交记录、错误分布情况以及学习进度，教师可以及时调整教学策略，针对常见问题设计专项讲解内容。例如，当监测到大部分学生在学习join操作时遇到数据倾斜问题，教师可以有针对性地补充相关优化策略的讲解，并结合实际案例演示不同partition策略的效果。这种基于数据驱动的教学反馈机制，不仅能够精准匹配学生的学习需求，还能显著提升课堂教学的针

对性和有效性。

5 大语言模型赋能Spark编程教学实践

基于某高校《Spark大数据分析》课程展开教学实践。该课程面向计算机科学与技术专业本科生，开设于第6学期，采用“理论+实践”双轨制（各16课时），依托超星学习平台进行资源分发与过程管理，同时结合VSCode集成大模型插件，提供智能化的交互式编程体验。课程重点考察学生对Spark编程核心概念的理解程度、编程能力的提升以及实验任务的完成效率。。

学生通过交互的方式利用大模型，其介入学习的过程可以拆解为“需求拆解-代码生成-性能优化”三阶段。例如常见案例“如何使用Spark计算文本数据中的单词频率”，大模型首先拆解子任务，并生成带有中文注释的PySpark代码框架。在学生补充核心逻辑时，模型的“链式思考”功能能够提供实时支持，帮助学生快速完成代码编写。实验数据显示，学生在初次实验中的代码编写时间平均减少28.4%，表明大语言模型能够有效降低初学者的学习门槛，使学生更快进入实践操作阶段。

此外，大语言模型通过智能任务分配与协同编码支持，推动小组合作从“机械分工”向“能力互补”转型。借助大模型根据学生的技能标签划分的分工，学生更深入掌握了Spark编程的前沿技术，并对协作编程有更加深入的思考。大模型的引入，使得学生对Spark编程基础知识掌握得更加牢固，编程效率也大大提升。

同时，在产学合作的实际项目中利用大模型辅助学习，学生更好地提前了解企业中如何运用所学知识解决实际问题的过程，为以后的就业奠定了一定的基础。编程过程中，实时的交互也帮助学生及时发现代码中的错误，借助大模型实现快速准确的编程。除此之外，从结果上来看大模型的智能化评价也能够对学习过程进行细致分析，及时追踪学生的学习进展并提供个性化反馈，激发学生的学习兴趣，帮助显著提升学习效率。

整体来说，引入大语言模型技术后，学生对于得到了极大的提升，降低了Spark编程教学的学习门槛，有助于培养学生的创新思维和解决复杂工程问

题的能力。

6 结论

当前，Spark编程作为大数据处理的核心技术，其教学面临着分布式系统复杂性高、个性化学习支持不足等多重挑战，传统教学模式已难以满足新工科背景下对实践型与创新型人才培养的需求。大语言模型凭借其在代码生成、智能答疑和多模态交互等方面的优势，与Spark编程教学在任务拆解、性能优化及案例实践等环节高度契合。将大语言模型技术融入Spark课程体系，需要引导学生深入理解分布式计算原理与大模型技术机理之间的融合逻辑，从而培养其系统设计能力和解决复杂问题的思维方式。本文基于某高校《Spark大数据分析》课程的教学实践也表明，该模式可显著提升代码规范性、任务执行效率及创新方案产出率，为大数据编程教育提供新范式。未来需进一步探索大模型与传统教学资源的动态协同机制，持续优化人才培养体系。

参考文献

- [1] 祝智庭, 赵晓伟, 沈书生. 融智课堂: 融入AI大模型的创新课堂形态[J]. 电化教育研究, 2024, 45(12): 5-12.
- [2] 郭丰, 杨清香, 郑春辉, 等. AIGC赋能的新形态工科实验教学初探[J]. 电化教育研究, 2025, 46(01): 72-78+85.
- [3] 陶炜, 沈阳. 从ChatGPT到Sora: 面向AIGC的四能教育和范式革新[J]. 现代教育技术, 2024, 34(04): 16-27.
- [4] 徐慧, 鞠小林, 王皓晨. 大模型下编程教学面临的挑战与应对[J]. 计算机教育, 2023, (11): 60-64.
- [5] 申丽萍, 何朝帆, 曹东旭, 等. 大语言模型在中学历史学科中的应用测评分析[J]. 现代教育技术, 2024, 34(02): 62-71.
- [6] 鞠小林, 张艳梅, 王皓晨, 等. 基于大语言模型辅助教学的Python编程课程教学探索[J]. 计算机教育, 2024, (09): 33-37. DOI:10.16512/j.cnki.jsjyy.2024.09.024.
- [7] 汪芳, 赵左, 王毅航, 等. 人工智能在程序设计教学应用中的探索与实践[J]. 计算机教育, 2023(11): 45-50.
- [8] 孙沛铨, 詹涵舒, 张爱, 等. 大语言模型驱动的学生评教智能分析法[J]. 计算机教育, 2025, (01): 215-219.
- [9] 张彦芳, 高璐, 李艳. AI技术项目驱动下的程序设计类课程教学模式创新[J]. 计算机教育, 2025, (02): 159-163.

- [10] 尹良泽, 徐建军, 李姗姗, 等. 人工智能时代下的计算机程序设计课程教学探索[J]. 计算机教育, 2025, (02): 123-127.
- [11] 徐慧, 鞠小林, 王皓晨. 大模型下编程教学面临的挑战与应对[J]. 计算机教育, 2023(11):60-64.
- [12] Cooper G. Examining science education in ChatGPT: An exploratory study of generative artificial intelligence[J]. J Sci Educ Technol, 2023, 32(3): 444 - 452.
- [13] Kohnke L, Moorhouse L, Zou D. Exploring generative artificial intelligence preparedness among university language instructors: A case study[J]. Computers and Education: Artificial Intelligence, 2023(5): 100156.

