

面向程序设计类课程的知识图谱构建和应用设想

伍 星^{1*}, 邓秋菊², 熊心志¹

¹ 重庆大学计算机学院 重庆

² 重庆移通学院公共大数据安全技术重庆市重点实验室 重庆

【摘要】随着信息技术快速发展和在各行业中广泛应用,尤其是人工智能时代到来,引发了社会对编程专业人才需求快速增长,因此对程序设计类课程的教育提出了更高要求。各高校目前已开设众多程序设计类课程,但大部分学生中学阶段并未接触过程序设计,因此教学过程中部分学生存在较大学习困难,因此本文提出构建面向程序类设计课程的知识图谱对课程内容进行有效梳理和可视化展示以降低学习难度,同时针对课程学习前期知识点掌握不牢导致中间阶段学习困难,以及程序编写过程中对编译错误与已学习知识点难以关联导致程序调试困难,通过建立课程知识点与知识点、程序编译错误提示与知识点的非线性链接降低学生对相关知识点查找和学习。

【关键词】知识图谱;程序设计;课程教学

【基金项目】重庆大学教学改革研究项目(2014Y23):基于搜索的课程知识共享平台研究;重庆市研究生教育教学改革研究项目(yig133019):计算机类全日制专业学位实践教学研究;重庆移通学院课堂教学改革项目(23JG2168):Java 语言程序设计

【收稿日期】2025 年 10 月 9 日

【出刊日期】2025 年 11 月 2 日

【DOI】10.12208/j.ije.20250395

Construction and application of knowledge graphs for programming courses

Xing Wu^{1*}, Qiuju Deng², Xinzhi Xiong¹

¹School of Computer Science, Chongqing University, Chongqing

²Chongqing Key Laboratory of Public Big Data Security Technology, Chongqing Yitong University, Chongqing

【Abstract】With the rapid development and widespread application of information technology in various industries, especially the arrival of the artificial intelligence era, there has been a rapid increase in demand for programming professionals in society. Therefore, higher requirements have been put forward for the education of programming courses. At present, many programming courses have been designed in various universities, but most students have not been exposed to programming during their secondary school years. Therefore, some students have significant learning difficulties in the teaching process. Therefore, this article proposes to construct a knowledge graph for programming courses to effectively organize and visualize the course content to reduce learning difficulty. At the same time, in order to address the difficulties in learning in the middle stage caused by weak grasp of knowledge points in the early stage of course learning, as well as the difficulty in linking compilation errors with learned knowledge points during program writing, non-linear links between course knowledge points and knowledge points, program compilation error prompts and knowledge points can be established to reduce students' search and learning of related knowledge points.

【Keywords】Knowledge graph; Programming; Course teaching

1 研究背景

程序设计类课程作为高校计算机专业必修课程和理工类非计算机专业必修公共基础课程,具有授课面积大,学生能力差异大的显著问题。尤其是信息技术迅速发展及其在各领域的广泛应用,尤其是近十年来人

工智能技术突飞猛进的发展,具有程序设计思维,尤其是良好编程能力对选修该课程的同学至关重要,只有能够针对实际问题设计和实现算法的能力才能为后续课程中工程实现能力奠定良好基础。

程序设计类课程教学过程中教学方式持续研究和

*通讯作者:伍星(1978-)男,四川遂宁人,汉族,博士,副教授,研究方向:人工智能。

分析对降低学习难度、激发学习兴趣、提高强编程实践能力、促进学生计算思维养成具有重要意义。所以, 本文首先对目前各高校程序设计类课程的教学状况进行了解和认识, 找出导致教学效果不佳, 尤其是学生实际编程能力较弱的本质原因, 并通过对被这些问题所困扰的学生提供自动化的实时在线帮助让程序设计学习变得更为友好。

2 程序类课程的问题

高校中的程序设计课程一般在大一进行授课, 各个学校根据培养方案开设了不同的程序设计类课程, 比如: C、C++、C#、Java 和 Python。这些课程均采用传统教学大纲和教学模式。教学大纲通过对程序设计中的数据表示、执行流程、模块化设计、复杂数据结构、外部数据读写等知识点的学习, 使学生能够掌握一门帮助各专业后续教学且具有广泛应用价值的编程语言。教学模式通常采用理论课和实验课分离, 学时各占一半或或者理论课占比较大的方式。笔者结合多门程序设计类课程的教学经验分析, 发现现有模式存在如下问题:

2.1 授课进度与前期知识点掌握存在冲突

首先, 学生从高中学习刚转入大学学习会对大学课程授课进度存在极大不适应。大学课程由于理论学时存在严格要求, 授课进度会明显快于高中。其次, 程序设计课程的内容与数学、英语这些课程存在明显差异。数学、英语这类课程是中学课程的延续, 知识的跨度和难度不大。虽然教育快速发展让大部分学生在中学了解计算机相关知识, 但中学开设程序设计课程的并不多, 因此程序设计类课程的内容对大多数学生而言属于进入了一个完全陌生的领域, 从课程一开始就会被众多新名词和新概念包围, 而这些新概念在原有知识体系中并没有可参照对象。最后, 课程教材中知识点较多, 并且为了教材内容完整性, 存在许多小知识点, 而这些知识点在课程前期大部分需要学生记忆。上述三点原因造成了在授课进度要求前提下, 学生在课程前期感觉较大学习压力, 顶住压力的同学会逐渐熟悉和习惯, 但一部分同学会存在理解困难, 学习进度较慢和掌握程度较差, 而当更多新的教学内容不断涌来时, 难度进一步加剧, 导致一步赶不上, 步步赶不上。

2.2 后期教学内容与前期知识的关联存在缺失

目前程序设计类课程教材大多按照章节知识点安排, 属于线性编排, 教材内容中知识点之间缺乏非线性关联, 尤其是到了课程教学的中后期教材中例题程序由于问题复杂程度加大, 程序会越来越长, 而前期知识

点理解存在缺失的同学阅读这些程序就会存在较大困难, 即使发现了困难也不知道该学习什么知识点可以解决所面临的困难, 从而导致学习难度增大, 甚至放弃对课程的学习, 因为已经无法知道可以通过补什么内容可以赶上学习进度了。

2.3 实验教学方式与解决编译错误的关联存在缺失

程序设计类课程的实验教学一般包括验证性和设计型两类, 二者均需要在开发环境中编写和调试实际的程序代码, 而课程实验课的早期大部分同学不适应程序设计语言复杂的规则要求, 因此编写的程序会存在大量编译错误。虽然近年来学生英语水平快速提高, 但是当出现编译错误后, 大部分同学面对编译环境给出的错误提示仍然束手无策, 主要原因既可能是错误提示的单词超过了中学词汇量, 也可能是还未意识到英语不仅仅是一门课程知识, 更是一门用来解决问题的语言工具, 但更大原因是现有错误提示信息与教材中知识点间联系没有有效体现。

针对程序设计类课程教学中存在的上述问题, 本文提出利用知识图谱对课程知识点进行有效梳理, 然后以该知识图谱为基础, 投建课程中的知识点之间、例题程序中程序语句与知识点之间, 构建学生编译错误提示与知识点之间这几类信息的直接关系, 为学生的学习过程提供自动化的实时在线帮助, 从而降低学生的知识查找难度。

3 解决方案

3.1 课程知识图谱构建

知识图谱是用可视化方式对真实世界中存在的各种实体和概念、以及他们之间的相互关系进行描述。知识图谱是自然语言处理基础上发展出来的, 其构建过程依赖于自然语言处理中的命名实体识别和实体关系抽取技术, 并利用非关系数据库等进行存储以提供便利的检索。随着自然语言大模型技术快速发展, 命名实体识别和实体关系的精度越来越高, 引发了各个领域的知识图谱构建。构建过程中利用命名实体识别和实体关系抽取技术从海量领域文本中抽取大量实体, 并对实体间关系进行分类。构建之前需要领域专家定义实体的类别和关系的类别。

相对于利用海量数据构建知识图谱的方法, 本研究根据实际需要提出了自动构建与人工构建相结合的方法, 主要原因是课程教材的电子版数量无法达到海量数据级别, 自动抽取会容易存在较多错误, 而作为课程的知识图谱又必须向学生传递准确的知识。具体方

法是由教师首先定义所要抽取实体的类别和实体关系的类别, 然后选择较好的命名实体抽取和实体关系抽取算法, 利用能找到的程序设计课程电子版作为数据源进行直接抽取。自动抽取完成后由教师完成对抽取结果组成知识图谱的审核、修改后形成每门程序课程的知识图谱。教师在对自动抽取结果确认过程中, 还应进一步确定根据知识图谱中结点和关系与其对应知识点的重要程度。

3.2 课程内容与知识图谱的关联

课程内容包括知识描述文本和例题程序代码两部分, 要实现与知识图谱中知识点的关联必须先实现对教材内容和例题程序的电子化。教材内容如果有电子版可使用最好, 如无电子版也可以使用扫描设备进行扫描和识别, 例题程序代码教材一般会提供因此可直接使用。

3.2.1 知识描述文本与知识图谱的关联

知识图谱中包括表示知识点的命名实体, 因此可以采用自然语言技术自动定位知识描述文本中出现的与知识点有关的描述, 并生成该描述与知识图谱中的知识点的超级链接, 针对这些链接教师进行人工审核, 从而形成知识描述文本与知识图谱的关联图, 避免现有教材中对知识以章节为单位的线性描述。

3.2.2 例题程序代码与知识图谱的关联

例题程序代码包含多条语句, 每条语句对应至少一个知识点, 因此可以将每句代码与知识点的关联可以看做多个二分类的文本分类问题, 分类器个数就是知识点的数量, 每个二分类器的作用就是判断一条语句是否属于某个知识点。例如: 知识图谱中的知识点可以是: 变量定义、输入输出、条件表示、选择结构、循环结构、函数定义等。

程序设计类课程的语言较多, 且同一种语言也有很多种教材, 如果每位教师对自己所使用教材全部人工标注需要耗费大量人力和时间, 会降低授课教师使用积极性。同时不同学校学生的学习能力不同, 教师可能针对自己的学生需要指定不同粒度的知识辅助。针对此问题提出了例题程序代码与知识图谱的关联采用封闭和开放相结合的原则, 封闭是指构建基本的构建基本的例题程序代码与知识图谱的关联, 开放是指教师可以根据自己需求进行进一步的定制化管理(新增、删除或修改)。封闭的例题程序代码与知识图谱的关联过程采用机器学习算法实现可有效降低使用难度, 而机器学习依赖于大量标注数据。虽然目前程序设计类教材众多, 每本教材都基本提供了例题代码, 可

比较容易收集大量程序代码, 但如果对数据进行全面标注仍然存在较大成本开销, 因此提出分为人工、半自动和全自动的数据标注过程。在收集大量例题程序代码基础上首先进行人工标注: 选择其中小部分程序代码对每条语句对应知识点进行人工标注。下一阶段进行半自动标注: 当标注量达到一定程度后开始训练针对每个知识点二分类的分类器, 利用训练得到分类器对所搜集的所有数据进行自动分类, 人工对分类结果审核, 审核合格的数据加入到标注数据中。最后阶段为全自动标注: 当分类器达到较高分类精度后, 利用分类器对所有数据或新采集数据自动分类, 从分类结果中选择高置信度的数据加入标注数据集。

3.3 程序错误与知识点的关联

实验课中学生需要进行大量验证性和设计性程序编写, 该过程中学生容易受到读不懂编译器的错误提示信息困扰, 因此针对学生出错程序的错误提示给出之相关的知识点, 可以让学生知道正确的语法该如何写, 从而有效降低学生学习难度。该功能实现需要建立一个常用的提示信息与知识图谱中知识点的映射关系, 因此提出以人工方式建立常见错误提示与知识点的映射关系。首先在课程中采集大量学生编写的发生编译错误的源程序, 然后人工标注这些错误相关的知识点, 而对程序错误的搜集不可能短时间完成, 因此采用增量方式进行, 并在使用中实时增补。学生编译程序出现错误后首先应立足于根据编译器给出的提示信息自己解决, 无法解决时将自己的程序和错误提示进行提交, 提交后反馈针对错误的知识点和与之相关联的正确实例代码。上述过程还可以及时搜集课程中学生目前实际编写程序的能力和知识掌握程度, 教师可根据总体存在的问题进行针对性的知识弥补。

4 结束语

本文所设计方案以程序设计类课程的知识图谱构建为核心将知识点进行组织, 实现课本知识、程序编译错误与知识点之间的非线性链接, 让学生可以在阅读教材和编程实践中遇到问题时, 可以更好的获取实时帮助, 以避免错误无法及时解答导致的学习兴趣降低。

虽然该方案可以有效降低学生在程序设计课程中知识学习和程序编译错误的解决, 缓解课程学习中中期期的难题, 但随着课程知识的学习和题目难度增加, 程序编译错误数量会随着对知识熟悉而降低, 但逻辑错误会明显增加。此时虽然向学生推送知识点的方式无法解决这类问题, 但仍然可以通过学生将错误程序提交给教师, 教师分析通过调试分析问题, 并用这些学

生实际发生逻辑错误的程序讲解调试过程,会更加引发学生的兴趣,从而摆脱程序设计课程中教师都会讲程序调试方法,而大部分同学在程序出现错误后却很少使用调试工具来解决问题。

参考文献

- [1] 易任娇,朱晨阳,周竞文,国内外一流高校计算机入门类线上课程调研[J],计算机教育,2022(6):79-83.
- [2] 刘凤娟,赵蔚,姜强. 基于知识图谱的个性化学习模型与支持机制研究[J],中国电化教育,2022(5): 75-81,90.
- [3] 魏晔,崔贯勋,基于知识图谱的 Python 程序设计课程教学设想[J],计算机教育, 2024(2): 51-54.
- [4] 孙丽郡,孟繁军,徐行健.课程知识图谱构建技术研究综述[J].计算机工程,2025,51(11):1-21.
- [5] 冷泳林,新工科背景下基于知识图谱的程序设计类课程群建设研究[J],计算机应用文摘,2024(8):4-6.
- [6] 耿冰蕊.课程知识图谱的构建与教学应用初探——以“计算机网络程序设计”课程为例[C]//北京高校电子信息类专业群暨教育部电子信息类专业虚拟教研室全国院校教育教学研究成果论文集.中国传媒大学信息与通信工程学院,;2025:402-407.
- [7] 郑飞军.基于 Python 程序设计课程的知识图谱构建研究[J].电脑知识与技术,2024,20(33):171-174.
- [8] 肖明胜,王敏,郭英清,等.程序设计语言课程知识图谱构建研究[J].赣南师范大学学报,2022,43(06):95-100.

版权声明: ©2025 作者与开放获取期刊研究中心 (OAJRC) 所有。本文章按照知识共享署名许可条款发表。
<http://creativecommons.org/licenses/by/4.0/>



OPEN ACCESS