

文章编号: 2095-2163(2021)07-0144-08

中图分类号: TP183

文献标志码: A

# 基于 Neo4j 的 C++课程知识图谱的构建和推理

郎亚坤, 苏超, 王国中, 李国平

(上海工程技术大学 电子电气工程学院, 上海 201620)

**摘要:** 当前教育领域对知识图谱的研究大多停留在对学科间知识图谱的研究上, 而这类知识图谱普遍存在节点、关系和属性单一, 以及知识点划分颗粒度较粗等问题。然而, 在实际教学过程中, 学科内的知识图谱对学生的作用更大, 比如: 当学生对某一知识点掌握程度不够的情况下, 可以根据知识图谱找到与当前知识点相似的知识点或者基础知识点进行补充学习; 或者对某一知识点已经掌握的情况下, 可以根据知识图谱找出下一个要学的知识点等等。因此, 本文基于 Neo4j 图数据库构建了 C++课程知识图谱并给出相应的知识点推理。实验结果表明: 构造后的课程知识图谱包含 C++课程领域的所有知识点及知识点间的关系, 可以很好地进行知识点和学习路径的相关推理。

**关键词:** Neo4j; C++; 知识图谱; 知识推理

## The construction and reasoning of C++ course knowledge graph based on Neo4j

LANG Yakun, SU Chao, WANG Guozhong, LI Guoping

(School of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai 201620, China)

**[Abstract]** The current research on knowledge graphs in the education field mostly stays on the research of interdisciplinary knowledge graphs. However, this kind of knowledge graph generally has problems such as single node, relationship and attribute, and coarse granularity of knowledge points. And in the actual teaching process, the knowledge graph within the subject has a greater effect on students. For example, when students do not have enough mastery of a certain knowledge point, they can find knowledge points similar to the current knowledge points or basic knowledge points according to the knowledge graph for supplementary learning. Or when they have mastered a certain knowledge point, they can find the next knowledge point to learn based on the knowledge graph, etc. Therefore, this paper constructs a C++ curriculum knowledge graph based on the Neo4j graph database and gives the reasoning of the corresponding knowledge points. The experimental results show that the constructed curriculum knowledge graph contains all the knowledge points in the C++ course field and the relationships between the knowledge points, and can be used to perform relevant reasoning about the knowledge points and learning paths.

**[Key words]** Neo4j; C++; knowledge graph; knowledge reasoning

## 0 引言

知识图谱(knowledge graph, KG)自2012年由Google公司提出以来,目前已受到工业界和学术界的广泛关注。知识图谱最初提出的目的在于优化搜索引擎,而随着近些年的快速发展,目前已广泛应用于电商、金融、医学等领域,在智能搜索、智能问答、智能决策和推荐系统等方面发挥着重要的作用<sup>[1]</sup>。

近年来,随着互联网技术的迅猛发展以及“互联网+”理念在各领域的应用,同时随着大数据、云计算、人工智能等新一代信息技术的发展,将新一代信息技术应用在教育领域,打造教育信息化已是必

然趋势<sup>[2]</sup>。而无论是传统教育、还是以“互联网+”为背景的新型教育,教育本身都是一个数据密集型行业,而知识图谱的本质也是将事物和概念通过相互之间的关系连接起来,因此,将知识图谱应用在教育领域是一种必然的趋势<sup>[3]</sup>。

当前,知识图谱在教育领域中应用的研究越来越多。如,孙涵等人<sup>[3]</sup>根据老年教育的特征,探索分析了知识图谱在老年教育领域的可能性。吉婷婷<sup>[4]</sup>基于CNKI期刊文献关键词搜索,运用关键词共现的知识图谱可视化分析方法分析了我国地理学科核心素养研究的格局与演进特征,以期为地理学科核心素养研究的演进方向与发展趋势提供参考。

**基金项目:** 国家重点研发计划资助(2019YFB1802700);上海工程技术大学研究生创新计划资助项目(19KY0232)。

**作者简介:** 郎亚坤(1994-),男,硕士研究生,主要研究方向:推荐系统、智能教育;苏超(1993-),男,硕士研究生,主要研究方向:图像处理、智能教育;王国中(1962-),男,博士,教授,博士生导师,主要研究方向:数字音视频信息处理、智能信息处理、智慧教育等;李国平(1974-),男,博士,高级工程师,主要研究方向:音视频编码、智能媒体处理、机器学习与识别等。

**通讯作者:** 王国中 Email: wanggz@sues.edu.cn

收稿日期: 2020-12-20

王兰成<sup>[5]</sup>认为多学科协同开展网络舆情知识图谱研究,将形成相互补充和相互验证的有机整体成果,具有重要的理论价值和现实意义。张春霞等人<sup>[6]</sup>设计了基于数学课程本体的数学课程知识图谱构建方法,提出了一种基于数学课程知识图谱的知识推理方法等等。

综上所述,知识图谱应用在教育领域的研究已越来越多,但是现有的教育知识图谱更多的是针对学科间的研究,缺乏对单一课程知识点的分析,然而在学生的真实学习环境中,学生更想获得课程中知识点的分布情况、学习顺序以及难易程度等。本文以学生在课程中的学习为背景,设计构建出一种基于 Neo4j 的 C++课程知识图谱,并根据构建的知识图谱给出详细的 C++知识点推理过程,旨在将构建后的知识图谱应用于学生的课程设计、学习路径推荐等方面,实现个性化教学。

## 1 知识图谱相关研究

### 1.1 知识图谱的定义

知识图谱是以大量的数据为基础,将其整理成机器能处理的知识库能够以可视化的形式展示的网络组织结构<sup>[7]</sup>。具体是以科学知识为对象,显示学科的发展进程与结构关系的一种图形,并同时具有“图”和“谱”的双重性质和特征<sup>[8]</sup>,本质上是一个存储知识实体与实体之间关系的结构化网络,可以帮助形式化描述、理解现实世界的事物及其相互关系。知识图谱由“头实体—关系—尾实体”三元组构成,以三元组的方式来表示以及存储实例数据,构建数据间的语义链接<sup>[9]</sup>。知识图谱的三元组通常可以表示为  $G = \{E, R, S\}$ , 其中  $E = \{e_1, e_2, \dots, e_{|n|}\}$  表示知识图谱中的实体集合;  $R = \{r_1, r_2, \dots, r_{|n|}\}$  表示知识图谱中的关系集合; 而  $S \subseteq E \times R \times E$  表示知识图谱中的三元组集合<sup>[10]</sup>。

### 1.2 知识图谱的分类

根据知识图谱的应用领域,一般将知识图谱分为以下 2 种,即:通用知识图谱 (General-purpose Knowledge Graph, GKG) 和领域知识图谱 (Domain-Specific Knowledge Graph, DKG)<sup>[11]</sup>。其中,通用知识图谱是指不面向特定领域的知识图谱,具有一定的普适性,而领域知识图谱是指面向特定领域的知识图谱,具有特殊性。对于通用知识图谱的构建,当前已有如:谷歌公司构建的 Freebase、莱比锡大学和曼海姆大学共同开发的 DBpedia、百度知心和搜狗知立方等代表性的通用知识图谱。而对于领域知识

图谱的构建,目前依然处于一种探索阶段,一方面在于领域知识图谱的构建过程中需要有专业的领域知识为前提,另一方面也在于专业领域内的数据资源体量比较小,满足不了数据量的需求,当前,领域内的知识图谱主要包含有:音乐领域图谱 MusicBrainz、地理领域图谱 GeoNames 等等<sup>[12]</sup>。

### 1.3 知识图谱的构建方式

知识图谱的构建方式主要有自上而下和自下而上两种方式。其中,自上而下的构建方式是先定义所需要的模式,再将各种实体知识加入到知识库中;而自下而上的方式是先从各种数据中抽取出知识实体,再筛选出置信度较高的知识实体去构建顶层的模式<sup>[13]</sup>。知识图谱的构建主要包括数据获取、知识抽取、知识融合、知识推理和知识应用等步骤<sup>[10]</sup>,如图 1 所示。数据的获取是知识图谱构建的前提,常见的数据格式有结构化的数据、非结构化和半结构化数据,结构化数据可以直接对数据提取,而非结构化数据和半结构化数据在提取前需要先转换为结构化数据;当获取到数据后,便可以进行数据的抽取以获取知识实体及实体之间的关系,然后对知识实体进行链接并进行本体推理、路径推理等,最终构建完成的知识图谱可以应用在如推荐系统、语义搜索等领域。

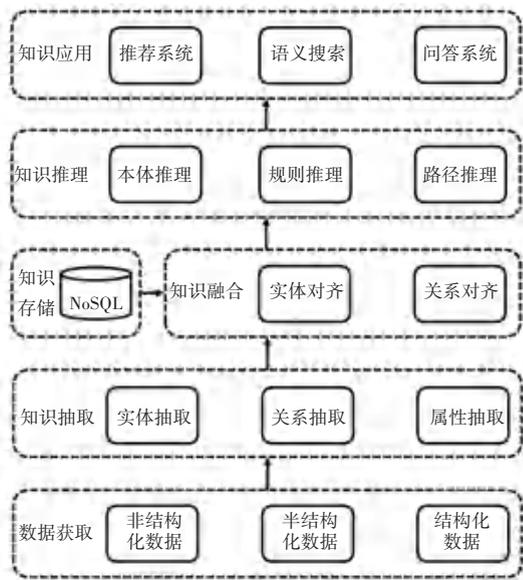


图 1 知识图谱的构建过程

Fig. 1 The construction process of the knowledge graph

## 2 C++课程知识图谱的构造

### 2.1 方法概述

当前,针对专业知识领域知识图谱的构建依然处于一种探索阶段,大部分专业知识领域的图谱构

造仅仅是一种概念间的关系建模,即针对概念中的显性知识进行构建。李光明<sup>[14]</sup>认为课程知识图谱的构造不仅要构建基本知识中的显性知识点,而且还要在此基础上根据知识概念间的关系进行推理演绎,从而获取知识图谱中的隐性知识点。因此,在构建 C++课程知识图谱的过程中,不仅要针对领域概念这类基础性知识进行划分,同时也要考虑隐性知识的获取。此外,专业知识领域的知识图谱的构造过程中还需要专业教师和专家学者指导审核,并根据其建议进行相应的调整与改进。综上所述,本文基于 Neo4j 图数据库构建一种 C++课程的知识图谱,整体流程图如图 2 所示。

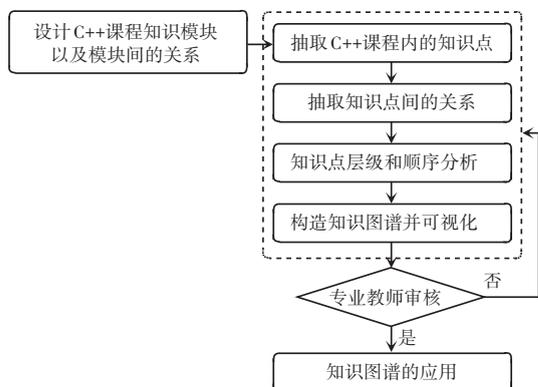


图 2 基于 C++课程知识图谱构建方法流程图

Fig. 2 Flow chart of the method for constructing knowledge graph based on C++ curriculum

从图 2 可知,构建 C++课程的知识图谱,首先需要设计 C++课程的知识模型,并获取各知识模块之间的关系,然后抽取课程中的知识点和知识点间的关系,并进行知识点层级和顺序的分析,最后可视化出 C++课程的知识图谱。

### 2.2 知识模块的设计以及模块之间的关系

针对具体的课程,由于每个学生的学习能力和先验知识的不同,因此,对于课程知识点的学习所需要的时间和精力也是不同的。但是不管学生如何学习,学生对课程知识点的学习必须符合教学大纲的要求。因此,对于 C++课程而言,必须先充分了解课程大纲的需求和掌握各知识模块以及模块之间的关系。本文以《C++Primer(第五版)》课程大纲为主,以《Visual C++面向对象程序设计教程与实验》和《C++面向对象程序设计》为辅,同时参考网络上相关的 C++课程资料作为补充,以此整体把握 C++课程的大纲需求。

经过分析,本文将 C++课程知识模块分成以下 5 种,分别为:基本语法模块、容器和算法模块、类和

数据抽象模块、面向对象编程与泛型编程模块以及高级主题模块。见表 1。

表 1 C++课程知识模块构建表

Tab. 1 C++ course knowledge module construction table

模块类别	部分模块属性
基本语法	变量和基本类型、标准库类型、数组和指针、表达式、语句、函数
容器和算法	顺序容器、关联容器、泛型算法
类和数据抽象	类、赋值控制、重载操作符与转换
面向对象编程与泛型编程	面向对象编程、模板与泛型编程
高级主题	异常处理、命名空间、多重继承与虚继承、类成员的指针、嵌套类

由表 1 可知,这 5 种知识模块涵盖了整个 C++课程的所有知识,同时,大纲要求课程中的各个知识模块的学习是按照一定的顺序进行的,如图 3 所示。除此之外,不同知识模块又有各自的子模块,以“类和数据抽象”为例进行说明,如图 4 所示。

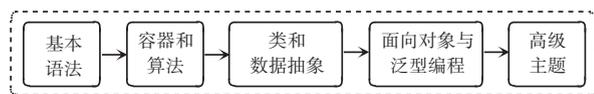


图 3 C++课程知识模块的学习顺序图

Fig. 3 The learning sequence diagram of the C++ course knowledge module

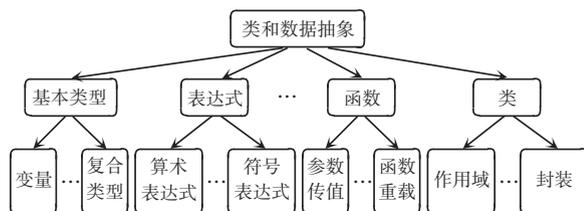


图 4 子模块的树形结构图

Fig. 4 Tree structure diagram of sub-module

### 2.3 课程知识点的抽取

课程知识点的抽取需要参考相关的课程大纲和课程目录,一方面在于课程大纲通常是由本专业的授课教师编写,能够以系统性、连贯性的形式叙述该课程的主要学习内容,另一方面在于课程目录不仅可以反映课程的知识结构,同时也考虑了知识点间的学习顺序。因此,根据课程大纲和课程目录构造知识图谱可以更好地反映学生完成相应的学习目标需要掌握的知识,帮助学生更好地梳理整个课程的知识脉络。

根据课程大纲进行知识点的抽取主要是针对课程大纲中的文本关键词进行提取,通常采用 jeba 中文分词工具进行中文的分词,并去除停用词以及不相关的词。本文以《C++》的课程大纲为例,对课程

大纲中的学习目标及每一章节做分词处理。课程大纲中的部分章节做分词处理前、后对比见表 2。

表 2 分词前、后的知识点对比表

Tab. 2 Comparison table of knowledge points before and after word segmentation

章节标题	章节部分知识点
类与对象	第一节:类与对象
	第二节:构造函数与析构函数,构造函数、缺省参数的构造函数、析构函数、重载构造函数和拷贝构造函数
	第三节:构建有效和可靠的类与对象
	第四节:对象数组与对象指针;对象数组、对象指针和 this 指针
分词前	类与对象 构造函数 析构函数 构造函数 缺省参数 构造函数 析构函数 重载构造函数 拷贝构造函数 构建有效可靠类 对象 对象数组 对象指针 对象数组 对象指针 this 指针
分词后	类 对象 构造函数 析构函数 构造函数 缺省参数 构造函数 析构函数 重载构造函数 拷贝构造函数 构建有效可靠类 对象 对象数组 对象指针 对象数组 对象指针 this 指针

从表 2 可以看出,切分后的知识点与原本知识点产生了很大的误差,比如知识点“构造函数”,分词后变成了“构造”、“函数”;知识点“对象数组”,分词后变成了“对象”、“数组”。造成这种现象的原因在于分词工具只是对通用词进行处理,在传统的

分词字典中并不包含这类专业词语。所以对专业领域的数据处理,要有专业的语料。因此,对于C++课程知识点的提取,本文构建了自定义分词典,并在分词工具和课程目录的辅助下,对分词后的知识点重新定义和补充。得到新的知识点集合见表 3。

表 3 根据自定义分词词典过滤后的知识点表

Tab. 3 Table of knowledge points filtered according to the custom word segmentation dictionary

章节标题	章节部分知识点
类 对象	构造函数 析构函数 缺省参数 析构函数 重载构造函数 拷贝构造函数 对象数组 对象指针 this 指针

通过对比表 2 和表 3 可以看出,重新定义后的知识点词典能够准确地切分出课程中的专业术语,进一步提高提取知识点的准确性。本文对《C++》课程中的所有章节进行相同的知识点提取及补充,共得到有效知识点的数量为 316 个。

### 2.4 关系抽取

#### 2.4.1 知识点关系抽取

关系抽取的目的是为了获取知识点间的关系,而采用分词工具获得知识点间的关系误差较大,为了确保准确性,需要手动定义知识点间的关系,主要包括有:包含关系、属于关系、进阶关系、基础关系、并列关系以及同义关系。其中“包含”和“属于”,“进阶”和“基础”是一组相对关系,只要知道其中一种关系便可以推导出另外一种关系。对此拟做详述如下。

(1)包含和属于关系。包含和属于这两种关系用于描述知识点的从属关系,表现为知识点间的层级性。课程中的粗粒度知识点可以拆分成更细粒度的知识点,如知识点 A 包含知识点 B,也就是说知识点 B 是知识点 A 的进一步细化,B 表示更具体的知识。在《C++》课程中,以“参数传递”知识点为例:参数传递可以拆分成“传值参数”和“传引用参数”,用三元组的形式可以描述为<参数传递,包含,引用参数>,<引用参数,属于,参数传递>,如图 5 所示。



图 5 包含和属于关系

Fig. 5 Containment and belonging relationship

(2)进阶和基础关系。进阶和基础这两种关系用于描述知识点间的依赖关系,表现为知识点间的顺序性。如知识点 A 是在知识点 B 已经掌握的基础上进行学习的,则知识点 A 是知识点 B 的进阶知识,基础关系则相反,即知识点 B 是知识点 A 的基础知识。这里以学习 C++“语句”知识点为例:在掌握 C++“语句”基础上,才能学习“函数”这个知识点,用三元组的形式可以描述为<语句,进阶,函数>,<函数,基础,语句>,如图 6 所示。



图 6 进阶和基础关系

Fig. 6 Advanced and basic relationship

(3)并列关系。并列关系表示为课程中的知识点的学习不存在顺序依赖性。如《C++》中的“控制逻辑运算符”和“算术运算符”知识点,用三元组的形式可以描述为<逻辑运算符,并列,算术运算符>,

如图 7 所示。



图 7 并列关系

Fig. 7 Juxtaposition relationship

(4) 同义关系。同义关系表现为多个知识点都可以对同一个概念进行描述。如知识点 A 可以对概念 X 描述, 知识点 B 也可对概念 X 描述, 则 A 和 B 具有同义关系。如对某个语句块进行“异常处理”, 可以使用 throw 表达式, 也可以使用 try 语句块进行处理。用三元组的形式可以描述为 < throw 表达式, 同义, try 语句块 >, 如图 8 所示。



图 8 同义关系

Fig. 8 Synonymous relationship

经过知识点关系抽取后, 4 种关系数量对应见表 4。

表 4 4 种知识点关系抽取结果表

Tab. 4 The results of four kinds of knowledge point relationship extraction results

序号	知识点关系名称	知识点数量
1	包含和属于	316
2	基础和进阶	97
3	并列	65
4	同义	24

### 2.4.2 知识点层级和顺序关系建模

为了统一描述, 本研究从课程大纲中提取的实体统称为知识点, 然而在实际的学习中, 知识点的学习应该由点到面, 最终形成一个知识网络。本文采用袁满等人<sup>[15]</sup>提出的细粒度课程知识组织模型定义知识点的顺序和层级关系, 模型的定义如下。

**定义 1 知识点模型** 该知识点不可分割成其它知识点, 即语义层级最低, 用  $K_p$  (Knowledge point) 表示。

**定义 2 知识链模型** 用  $K_c$  (Knowledge Chain) 表示, 是知识点的上一层级, 知识链是知识点之间按照不同的逻辑关系组成的链状结构。如式 (1) 所示:

$$K_c = \langle (K_{pi}, K_{pj}), R \rangle \quad (1)$$

其中,  $K_{pi}$  和  $K_{pj}$  表示不同的知识点,  $R$  表示知识

点间的关系, 分别为顺序关系和层级关系。

**定义 3 知识单元  $K_u$  (Knowledge Unit)** 表示知识链的上一层级, 知识单元是知识链组成。如式 (2) 所示:

$$K_u = (K_{ci} \mid i = 1, 2, 3, \dots, N) \quad (2)$$

**定义 4 课程实体模型  $K_{ckem}$  (Course Knowledge Entity Model, CKEM)**, 表示知识单元的上一层级, 由不同的知识单元构成。如式 (3) 所示:

$$K_{ckem} = (K_{ui} \mid i = 1, 2, 3, \dots, N) \quad (3)$$

本文将课程中的学习内容按照“知识点—知识链—知识单元”的形式组成知识实体模型, 如图 9 所示。

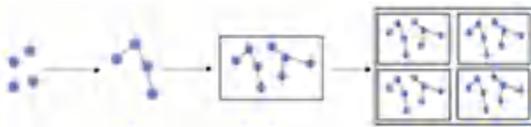


图 9 知识点的连接流程

Fig. 9 Connection process of knowledge points

### 2.5 知识图谱的可视化

本文采用 Neo4j 图数据库进行知识点的存储, 不同于传统的关系型数据库将数据存在库表字段中, 图数据库将数据和数据之间的关系存在节点和边中, 在图数据库中这被称作“节点”和“关系”。任何一个关系包含一个开始节点、一个结束节点和一个由开始节点指向结束节点的边组成, 数据库中所有节点通过各种关系联系在一起。传统数据库的一些特性如增删改查 (CRUD)、处理事务的能力在图数据库中也都支持。Neo4j 的图模型如图 10 所示。

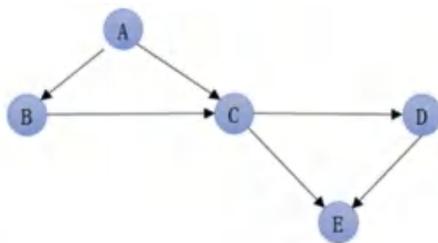


图 10 Neo4j 的图模型

Fig. 10 The graph model of Neo4j

图 10 中的圆圈表示节点, 节点间的有向连接线表示节点间的关系。根据提取到的知识点及知识点间的关系, 结合节点和关系的固有属性便可以构造知识图谱, 以三元组的形式 <实体 1, 关系, 实体 2> 存储在图数据库 Neo4j 中。本文依据《C++》提取的知识点实体及其关系分别构造实体文件和关系文件, 见表 5 和表 6。将实体文件和实体关系文件导

入 Neo4j 数据库,最终 C++课程知识图谱可视化如图 11 所示。

表 5 实体文件  
Tab. 5 Entity files

文件名	描述
CourseKnowledgeEntityModel.csv	课程名
KnowledgeUnit.csv	知识点单元
KnowledgeChain.csv	知识链
KnowledgePoint.csv	知识点

表 6 实体间关系文件

Tab. 6 Inter-entity relationship files

文件名	描述
include.csv	包含关系和属于关系
advance.csv	进阶关系和基础关系
constellation.csv	并列关系
synonym.csv	同义关系

图 11 展示了 C++课程知识图谱的局部知识点关系。图 11 中的绿色节点表示 C++课程名,红色节点表示知识点单元,蓝色节点表示知识链,黄色部分表示具体的知识点。

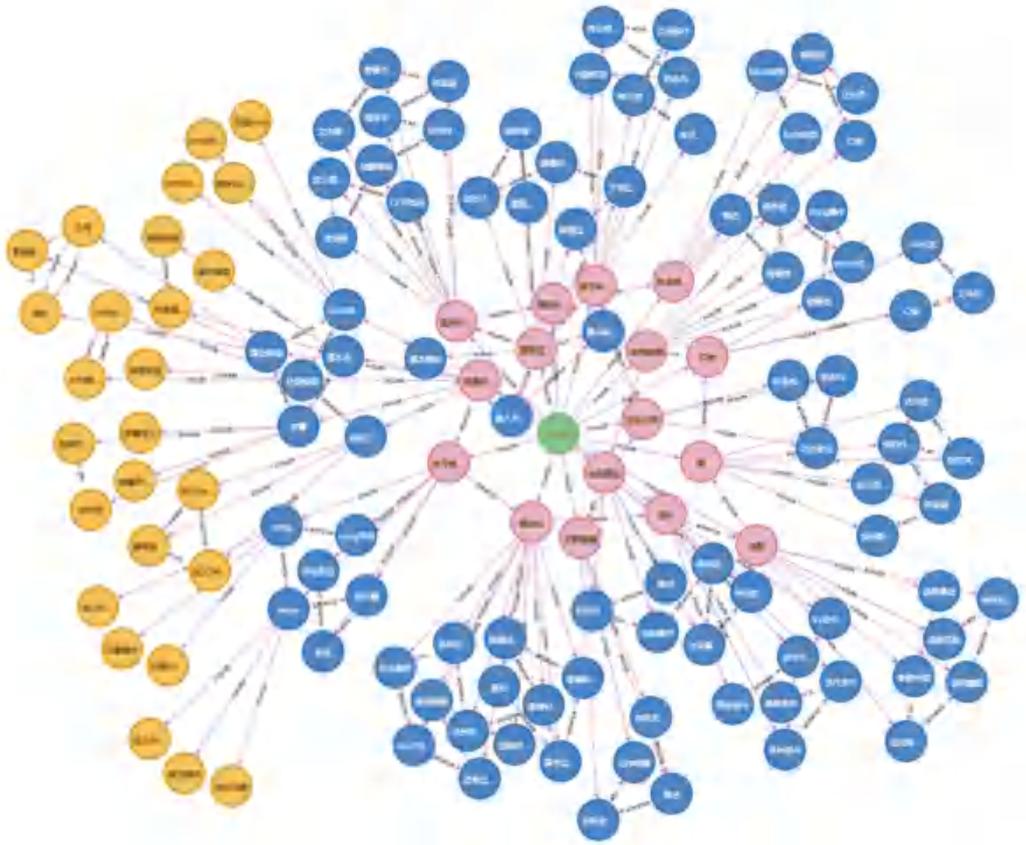


图 11 C++课程知识图谱局部截图

Fig. 11 A partial screenshot of the C++ course knowledge map

### 3 C++课程知识点的推理实验

知识点在知识图谱中的快速检索依赖于知识点间的错综复杂的关系,本文在构造《C++》课程知识图谱时建模了 4 类关系,分别是:包含(include)和属于、进阶(advance)和基础、并列(parallel)和同义(synonym)。针对 4 种关系的实验结果分述如下。

(1)包含和属于关系。包含和属于是一组相对关系,知道其中一个关系就可以推导出另外一个关系,在构造知识图谱时只需构造其中一个关系即可。

这类关系描述的是知识点的层级性。以知识点“表达式”为例,该知识点的包含和属于关系实验结果如图 12 所示。

图 12 表明了知识点“表达式”既属于“C++11”,同时又包含知识点集合{“位运算符”,“算术运算符”,……}等 12 个知识点。

(2)进阶和基础关系。进阶和基础也是一组相对关系,同理,在构造知识图谱时只需要构造其中一个关系即可。这类关系描述的是知识点的顺序性。以知识点“条件语句”为例,该知识点的进阶和基础

关系实验结果如图 13 所示。

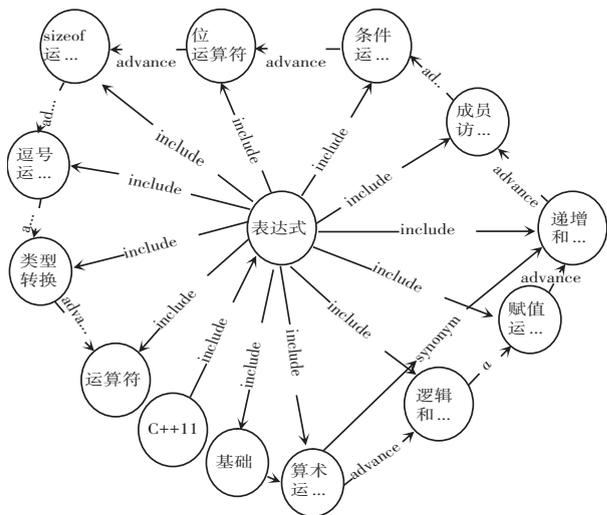


图 12 知识点“表达式”的包含和属于关系图

Fig. 12 The relationship diagram of inclusion and belonging of knowledge points "expression"



图 13 知识点“条件语句”的进阶和基础关系图

Fig. 13 The advanced and basic relationship diagram of knowledge points "conditional sentences"

图 13 表明了知识点“条件语句”的进阶知识点集合为{“迭代语句”,“跳转语句”},基础知识点集合为{“语句作用域”,“简单语句”}。需要注意的是,本文默认的进阶和基础知识点的输出深度为 2。

(3)并列关系。并列关系描述的是知识点的学习既不存在顺序关系、也不存在层级关系,如知识点“break 语句”、“continue 语句”和“goto 语句”,这 3 类知识点都表示语句的跳转,而“break 语句”强调的是跳出剩余的循环、“continue 语句”强调跳出当前循环;“goto 语句”则强调跳入指定的语句内执行。并列关系实验结果如图 14 所示。

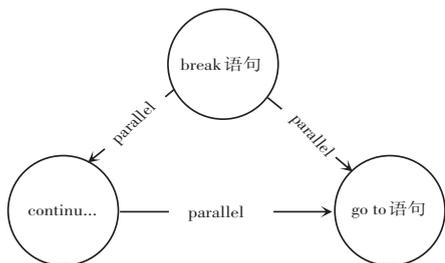


图 14 知识点“break 语句”的并列关系图

Fig. 14 The parallel relationship diagram of the knowledge point "break statement"

点可以对同一个概念进行解释,如知识点“if 语句”和“switch 语句”,这 2 个知识点都可以对语句的执行进行条件判断。同义关系实验结果如图 15 所示。

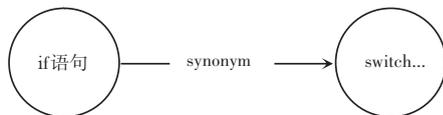


图 15 同义知识点“break 语句”和“switch 语句”

Fig. 15 Synonymous knowledge points "break statement" and "switch statement"

### 4 结束语

本文详细介绍了 C++课程领域知识图谱的构建过程,主要包括本体数据模型搭建、知识点获取、关系抽取和知识图谱可视化四个部分。首先分析了课程大纲中对知识点内容的概括,使用 jieba 分词工具和人工辅助的方式从数据集中获取知识实体,然后根据人工构建的关系词典抽取了实体间的关系,最后用 Neo4j 图数据库将获取的实体信息以及关系信息进行知识存储并可视化。虽然构造好的知识图谱可以根据具体学习场景完成知识点的推理,但是构造知识图谱的过程中需要专业教师的干预,降低了构建过程的智能化。下一步将重点研究在不影响推理准确率的基础上,提高知识图谱的构造流程智能化。

### 参考文献

- [1] 阮彤,孙程琳,王昊奋,等. 中医药知识图谱构建与应用[J]. 医学信息学杂志,2016,37(4):8-13.
- [2] 董小倩,张小花. 传统课堂与智慧课堂教学对比研究[J]. 现代交际,2019(2):12-13.
- [3] 孙涵,齐悦. 知识图谱在老年教育中的应用[J]. 电子技术与软件工程,2020(19):186-188.
- [4] 吉婷婷. 我国地理学科核心素养研究的格局演进—基于关键词的知识图谱分析[J]. 地理教学,2020(17):4-9,61.
- [5] 王兰成. 多学科视域网络舆情知识图谱研究的现状和展望[J]. 情报学报,2020,39(10):1104-1113.
- [6] 张春霞,彭成,罗妹秋,等. 数学课程知识图谱构建及其推理[J]. 计算机科学,2020,47(S2):573-578.
- [7] 覃晓,廖兆琪,施宇,等. 知识图谱技术进展及展望[J]. 广西科学院学报,2020,36(3):242-251.
- [8] HE X, LIAO L, ZHANG H, et al. Neural collaborative filtering [C]// Proceedings of the 26<sup>th</sup> International Conference on World Wide Web. Perty, Australia: ACM, 2017:173-182.
- [9] 姚萍,李坤伟,张一帆. 知识图谱构建技术综述[J]. 信息系统工程,2020(5):121,123.
- [10] 曹皓伟,徐建良,窦方坤. 基于 Neo4j 生物医药知识图谱的构建[J]. 计算机时代,2020(6):35-38.
- [11] 王毅,沈喆,姚毅凡,等. 领域事件图谱构建方法综述[J]. 数据分析与知识发现,2020,4(10):1-13.