文章编号:1671-4229(2023)05-0063-09

计算机课程相似度计算方法及其改进

仲启玉[1],张少宏[1*],丁 汉[1],张芷芊[2]
(1. 广州大学 计算机科学与网络工程学院,广东 广州 510006;
2. 广州华南商贸职业学院 信息工程学院,广东 广州 510650)

摘 要:在传统描述方法上,计算机科学课程一般使用教学计划和教学大纲来描述专业知识和课程知识点的整体结构,由于计算机科学课程结构复杂,各个课程间联系紧密,依靠传统方法不足以把握计算机科学课程的总体结构和专业课之间的普遍联系。文章针对传统方法的不足,提出构建基于计算课程知识图谱的方法,具体包括:①课程知识图谱的构建,量化课程知识点关系,构建知识点关系矩阵,量化各知识点间的关系;②课程关系的构建,量化所有课程间的关系;③国外计算机名校关系的构建分析。研究发现,计算课程知识图谱的构建和分析将对目前计算机科学专业课程的改革研究具有一定的创新意义,这种计算机科学专业课程知识图谱的构建方法,可弥补传统课程描述方法上的不足,为计算机科学专业课程改革分析提供一定的数据支持,同时,对相似度的衡量提供了一种新的方法,提高了相似度衡量速度。该方法也可移植到其他学科。

关键词:知识图谱;文本量化;相似度计算;聚类分析
中图分类号:TP391.1 **文献标志码**:A

Calculation methods and improvement of similarity in computer courses

ZHONG Qi-yu[1], *ZHANG Shao-hong*[1*], *Ding Han*[1], *ZHANG Zhi-qian*[2]
(1. School of Computer Science and Cyber Engineering, Guangzhou University, Guangzhou 510006, China;
2. School of Information Engineering, Guangzhou South China Business Trade College, Guangzhou 510650, China)

Abstract: In traditional methods of describing computer science courses, instructional plans and course outlines are commonly used to represent the overall structure of computer science's specialized knowledge and course content. However, due to the intricate nature of computer science courses and their interconnectedness, relying on traditional methods makes it challenging to grasp the overall structure of computer science courses and the connections between specialized courses. This article addresses the limitations of traditional methods in describing computer science courses and proposes a method for constructing a computer course knowledge graph to address these shortcomings. Specifically, this includes the construction of a course knowledge graph, quantifying the relationships between courses and knowledge points, building a knowledge point relationship matrix to quantify relationships between various knowledge points, constructing course relationships to quantify relationships between all courses, and analyzing relationships between top computer science schools internationally. The construction and analysis methods of the computer course knowledge graph are expected to have innovative significance for current reforms in computer science education. They provide a new approach to building a knowledge graph for computer science education and can be easily adapted to other disciplines. This new computer course knowledge graph fills in the gaps of traditional course description

收稿日期:2021-08-02;**修回日期**:2022-03-16
基金项目:广东省基础与应用基础研究基金资助项目(2022A1515011697)
作者简介:仲启玉(1996—),女,硕士. E-mail:1103930728@qq.com
* 通信作者. E-mail:zimzsh@qq.com
引文格式:仲启玉,张少宏,丁汉,等. 计算机课程相似度计算方法及其改进[J]. 广州大学学报(自然科学版),2023,22(5):63-71.

methods, and offers data support for the reform and analysis of computer science education. Additionally, it introduces a novel method for measuring similarity, thereby enhancing similarity calculation speed.

Key words: knowledge graph; text quantification; similarity calculation; cluster analysis

随着互联网技术的不断发展,计算机科学成为时下热门专业。对教育数据进行挖掘可用于设计更好、更智能的学习技术,并更好地为学习者和教育者提供有用信息[1]。如根据学生在教室中的座位选择来评估数学能力,对学生课堂笔记的记录情况研究学生学习信念和学习能力关系等[2]。通过对学生建模可以获得一个相对清晰的学生学习风格模式。文献[3]从学习系统中获得学生、课程、评价等级的数据,构建课程推荐系统,以及可用于预测学生成绩的方法,所提出的课程推荐系统可以在实践中应用。简单的课程标签化难以有效利用相关辅助信息,计算机领域知识点往往蕴含在高校教学大纲、课程大纲、网页以及计算机领域的文献等资源中,若将这些知识融合在一起,把相关知识点之间的层次关系和联系用知识网络的形式表现出来,形成简洁清晰的知识网络图[4],通过这样的图结构可以快速了解各知识点之间的联系和区别,对于掌握计算机知识整体结构帮助非常大。现有的课程仅提供本身的孤立信息,而对于某一领域知识的掌握不能仅靠孤立的课程学习,而是要掌握课程间的关系以及某课程和这一领域间的关系,比如计算机学科间的相似或递进关系对于计算机领域知识的学习就很重要,学习计算机不仅需要很好的数学基础,同时还需要其他学科背景知识的积累。现有的关于计算机课程关系的研究多为先修关系(prerequisite)的研究。文献[5]研究了大学课程中概念与课程之间的联系,将大学课程表示为一个以课程和概念为节点,以它们之间的连接为边缘的图,利用图论的方法来表示二者之间的关系。文献[6]提出了先决条件网络来可视化学术课程中的隐藏结构,将学术课程视为一个复杂的系统,其中的节点代表课程,节点之间的连接则可以从课程列表中轻松获得课程的先修关系。文献[6]提供了一种评估课程的方法,并且为课程修订提供了框架,使整个课程的重要程度可见,并提供了定量的分析。文献[7]研究如何从课程依赖中提取概念先决条件关系,提出一个优化的框架来解决该问题,并创建了一个用于研究此问题的真实数据集,包含来自11所美国大学的计算机科学课程清单以及课程的概念对和先修课程标签。文献[8]研究了一种基于课程和概念对大学课程中的连通性和知识流进行了定量检查的方法,为课程编制和表示课程依赖概念提供了有效方法。文献[9]提出一个可视化大学课程结构的工具,包括课程和知识之间的联系,以及使用图论概念检测、分析和可视化课程结构,便于课程学习和课程修订。本文以国外计算机名校的计算机科学课程为分析对象,以课程和知识点为双关系图,量化知识点之间、课程之间、高校之间的联系,进行计算机课程知识图谱的构建分析,直观反映计算机专业课程体系,为高校开展计算机课程和学生选择课程学习提供一定的数据参考。本文讨论了如何获取相关数据,如何提取课程关键词、构建语料库、如何计算相似度等。利用多种算法进行计算,并根据计算的结果进行对比分析,选择出最适合本文的算法。

1 构建语料库

1.1 课程数据的爬取

网络爬虫是一种能自动下载网页的程序,本文使用Python语言来设计适合本文的网络爬虫以获取数据。为获取具有代表性的计算机科学课程文本内容,本文选取的文本来自排名靠前的国外计算机名校。通过各个学校官网上的计算机科学课程信息,选取需要的字段。为了研究计算科学课程的知识图谱,选取字段:课程id\课程名\课程描述\课程关系,用课程描述来代表每门课程。爬取下来的课程信息保存为Excel文件,保存格式如下:课程ID(ID)、课程名(Name)、课程描述(Course Description)、排选课\预选课(Prerequisites),将每一项存在一列中,一共有4列数据,以便后续读取。数据保存格式如图1所示。最终爬取了约100所学校,近6 000门课程,每所学校存为一个Excel文件单独存放,每一行是一个课程信

息,文件名为学校名简写。

ID	Name	Course Description	Prerequisites
CS 101	Intro Computing: Engrg & Sci	Fundamental principles, concepts, and methods of computing, with emphasis on applications in the physical sciences and engineering. Basic problem solving and programming techniques; fundamental algorithms and data structures; use of computers in solving engineering and scientific problems. Intended for engineering and science majors.	Prerequisite: MATH 220 or MATH 221
CS 105	Intro Computing: Non-Tech	Computing as an essential tool of academic and professional activities. Functions and interrelationships of computer system components: hardware, systems and applications software, and networks. Widely used application packages such as spreadsheets and databases. Concepts and practice of programming for the solution of simple problems in different application areas. Intended for non-science and non-engineering majors.	Prerequisite: MATH 112
CS 125	Intro to Computer Science	Basic concepts in computing and fundamental techniques for solving computational problems. Intended as a first course for computer science majors and others with a deep interest in computing.	Prerequisite: Three years of high school mathematics or MATH 112
CS 126	Software Design Studio	Fundamental principles and techniques of software development. Design, documentation, testing, and debugging software, with a significant emphasis on code review.	Credit is not given for both CS 242 and CS 126. Prerequisite: CS 125. For majors only

图 1　数据保存格式

Fig. 1　Data saving format

1.2　数据预处理

使用 Python 的 NLTK 工具来进行英文文本的数据预处理,构建文本词库。为了对课程内容信息进行分析,需要对课程内容数据进行单词小写化、去标点符号、分词处理及去停用词等处理,以消除脏数据对结果的影响。

将所有单词统一小写化,标点符号在文本中无实际意义,因此,将文本去除标点符号。英文语料的分词方法有空格分词、re 匹配符号进行分词和 NLTK 分词器等,本文选用 NLTK 分词器进行分词。原理很简单,依据空格和标点符号来进行分词处理。停用词指的是在文本中没有什么实际意义的词,如“the”“all”“so”等词,在语料库中还会影响最终分析结果,因此,在对课程文本进行分析前,将这些词剔除。

由于英语单词词形多变,一个单词可能有名词、动词、形容词、副词等各种形式,其表达的词根意思是一样的,为了有效地提高关键词密度,需要对课程文本单词进行词形还原处理。词形还原就是返回词的原形,根据单词的词性来提取单词的词干,对单词进行词性的识别,根据词性标注来处理单词词缀,从而还原单词。可以看出,单词经过词形还原处理后是具有一定意义且完整的词,能准确地表达文本,符合对数据预处理结果的预期。根据单词词性进行词形还原,结果存储在课程表格第四列,数据预处理结果如图 2 所示。

id	name	description	words
CSCI 1011	Introduction to Programming with Java	ed programclassa	class java computer widget introductory language gui program method inheritance container override use applet structure oriented object control exception course
CSCI 1311	Discrete Structures I	ismrelationparti	mathematics calculus math mathematical graduate student science graph computer matrix prerequisite isomorphism fall induction proof formal tree semigroups 1231 credit 1220 propositional function equivalence field predicate group partition sequence relation set
CSCI 3907	Big Data	n the schedule o	spring class fall schedule topic announce
CSCI 3908	Research	projecta arrange	research prerequisite junior senior project experimentation arrange status apply
CSCI 6212	Design and Analysis of Algorithms	t be expect to l	calculus mathematical math algorithmic turing spring algorithm java graph optimization summer year posse student conquer greedy fall search application design procedural skill backtrack general technique theory machine np complete registration bound analysis traversal dynamic language program data expect good branch prior background divide oop structure discrete include know sort

图 2　数据预处理结果

Fig. 2　Results of data preprocessing

1.3 课程文本量化

计算机是无法直接识别文本的,为了对文本进行有效地数学分析,需要将其进行量化,转化为向量表示。词袋模型和词向量模型是两种最常用的模型。向量空间模型就是词袋模型,其中,最简单的是基于词的独热表示(one-hot representation)。工程上比较常用的是用词的 TF-IDF[10] 值作为权重,它是文本处理中最常见的一种权重计算方式;另一种就是文本的分布式表示方法(distributed representation),其中,LSA、LSI、pLSI,以及 Word2Vec[11]、Doc2Vec 方法都属于分布式表示。

词袋模型的方法将文本变成一串数字(索引)的集合,在词袋模型中,将文档单词映射,并统计这种单词的出现次数。这种向量表示法不保存原始句子中词的顺序。

词向量也叫词嵌入,就是将单词映射到向量空间里,将词用向量来表示。One-hot 表示方法是把每个词表示为一个很长的向量,向量的维度就是词汇表的长度,其中,绝大多数元素为0,只有一个维度的值为1,这个维度就代表了单词,但是这种表示方法中,任何两个单词都是孤立的,无法在语义层面上表达单词之间的关系。Distributed representation 表示方法通过训练,使所有单词向量构成向量空间,每个单词被映射到较短的单词向量。本文采用 distributed representation 来表示词向量。

1.4 特征词权重量化

对于文本数据的处理,首先就是要获取文本的特征信息,提取文本关键词,包括候选关键词提取、用词频或 TF-IDF 等过滤,以及使用阈值从候选词中选择关键词。

本文使用课程描述的特征词来代表这一门课程,特征词提取的算法有很多种,不同算法提取的效果也会不同。常见的有 TF 算法和 TF-IDF 算法等。TF 算法中一些常用词的权重很大,因此,不适合使用 TF 算法来衡量权重。本文选取了 TF-IDF 算法来计算语料库的词的权重,进行特征词的权重量化。TF-IDF 算法目前在文本分类中被广泛使用,它借鉴了信息检索。某词在某文档中是高词频,而在整个文档集中该词又是低文档频数,可以通过 TF-IDF 公式获取较高的权值。TF-IDF 倾向于过滤掉常用词并保留重要词,对比 TF 算法,TF-IDF 算法更符合本文对于特征词权重的衡量,因此,本文使用 TF-IDF 算法来量化表示特征词权重。

2 相似度计算

2.1 课程相似度计算

根据现有相似度度量方法进行特征词(知识点)与特征词(知识点)之间的相似度计算、课程与课程的相似度计算以及学校与学校的相似度计算。本文计算课程之间的相似度主要是单词语义相似度的度量,选用 TF-IDF 模型、LDA 主题模型及词向量模型等来建模并计算相似度,并通过这3种方法的相似度计算效果进行分析比较,从而选择适合本文的计算方法。

使用 TF-IDF 方法计算的课程相似度过于分散,从0.1~0.9,而且如果两门课程之间想要相同的词,则相似度为0,效果不好,不符合本文对于相似度计算的要求,因此,后续对于相似度的计算不使用这种方法。

利用 LDA 主题模型计算出来的相似度非常密集,都在0.5之间,效果不是很好,而且最致命的缺点是 LDA 模型相当不稳定,对于不同的语料库计算相似度需要训练不同数量的主题,由于本文语料库比较大,用这种方法计算相似度不切实际,因此,LDA 主题模型计算相似度的方法也不适用。

本文采用词向量来表示课程文本单词的表示方法,通过训练可以将每个词映射为一个固定长度的向量,这些向量构成了一个词向量空间,其中,每一个向量可以当作空间上的一个点,通过计算向量之间的距离来表示词之间的相似性[12]。图3为词向量文件,其中,每一个单词都是由一个300维的向量来表示的,一共有 13 119 个单词。从第二行开始每一行代表一个单词,用空格隔开,图中第二行代表单词“the”,后边是它的向量表示。这就是整个语料的整体结构,是一个标准的词向量表示文本,可以使用 gensim 等第三方库直接读取。

基于词向量计算课程相似度算法设计如下:

(1)计算词之间的相似度,即通过计算两个词向量的夹角余弦值来计算两个词的相似度;

(2)计算课程之间的相似度,即基于词之间的相似度矩阵,通过计算得出两门课程的词的相似度矩阵,然后计算词相似度矩阵行方向与列方向词的最大值/中位数相似度总和,再求平均值,即为两门课程之间的相似度。

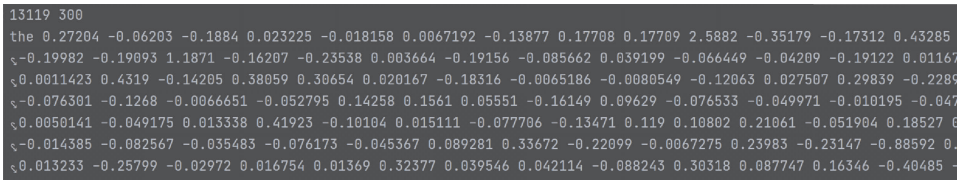

图 3　词向量文件

Fig. 3　Word vector file

词向量方法计算的相似度效果比前两种方法效果要好,但是相似度也相对比较集中。中位数方法与最大值方法整体趋势是一样的,本文选用最大值法,通过对比课程得出相似度的计算还是存在一定的偏差,因此,需要改进算法。

2.2　设计词组相似度计算的新算法

基于词向量计算课程相似度的算法效果虽然很好,但是对于本文存在着很大的弊端:计算相似度时间过长,计算 60 门课程之间的相似度要 0.5 h,而语料库的课程信息较多,因此,需要对算法进行改进。运行时间缓慢的原因主要有:①多次读取遍历向量模型:每一次计算词的相似度,就得遍历两遍词向量模型找到其对应的词向量,然后再计算它们之间的相似度,多次读取消耗时间;②多次重复计算词对相似度:课程的词有大量重复,也就意味着多次重复计算同一词对的相似度。

针对上述两个原因,对向量模型的算法进行优化设计:①减少多次读取遍历向量模型的时间,即使用字典将向量模型一次读取存储起来,字典{key:value}采用{词:向量}的存储方式,每当需要查找一个词的向量时,直接使用关键字索引的方式即可;②利用标注矩阵减少词对计算次数。

具体实现方法及步骤如下:

(1)遍历语料库,得到词汇表(即语料库所有单词词汇);

(2)生成标注矩阵。根据词汇表,生成词汇 * 词汇的二维矩阵,初始化全部为 0,遍历课程对,如果要计算相似度的词,则加 1;

(3)生成词相似度矩阵。根据词汇表,生成词汇 * 词汇的二维矩阵,初始化全部为 0,根据标记矩阵,如果两个词的标注矩阵值 >0,则计算这两个词的相似度,从而完成知识点与知识点关系的量化;

(4)计算课程之间的相似度。先构建两门课程的词矩阵,根据词相似度矩阵,得到两门课程的词矩阵的相似度矩阵,然后计算词相似度矩阵行方向与列方向词的最大值/中位数相似度总和,再求平均值,即为两门课程之间的相似度,从而完成所有课程关系的量化,描述所有课程之间的关系。

算法描述:

算法 1:　词组相似度计算算法

输入:计算机科学课程语料库,课程数据;
输出:课程知识点之间的相似度矩阵;
输出:课程之间的相似度矩阵;
输出:高校之间的相似度矩阵;

1. 读取课程数据,分别提取以下指标:
课程编号 ID、课程名字 Name、课程描述 Description、高校名称 School。
2. 构建课程知识点之间的相似度矩阵:
for each Word Pairs (w,v) in Description do
　根据所有词生成词矩阵(知识点矩阵),初始化为 0;
　if 词对(w,v)需计算相似度
　　then 词对(w,v)标注值 +1;
　else 词对(w,v)标注为 0;
　if 词对在标注矩阵值 >0
　　then 计算词对相似度;
　else 词对相似度值为 0;
end;
3. 构建课程相似度矩阵:
for each Course Pairs (i, j) in Description do
　for each Word w in Course Pairs (i, j) do
　　计算包含 w 的词对的相似度最大值 max;
　end;
　课程对(i, j)的相似度值:所有 max 加起来的平均值;
end;
4. 根据课程相似度矩阵与其构建方法构建学校相似度矩阵。

标注矩阵生成结果如图 4 所示。标注矩阵中,很多词对之间的标注矩阵远远大于 1,说明这

些词对不止计算一次。用标注矩阵的方法避免了这些词对多次计算相似度,从而加快了计算速度。图 4 中标注矩阵为 0 的单词对说明两个单词之间不用计算相似度的。

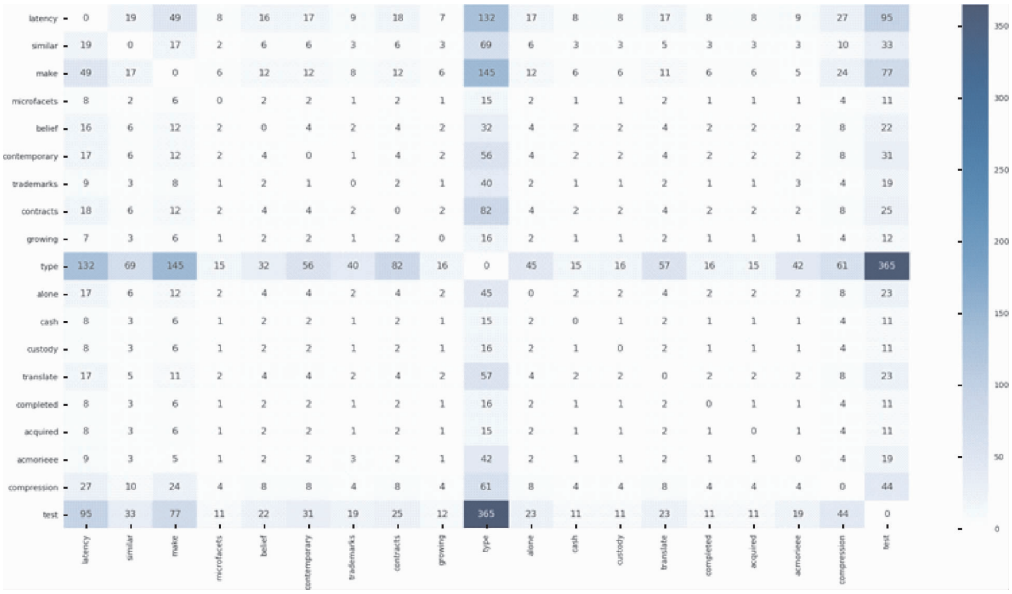

图 4 标注矩阵生成结果

Fig. 4 Results of labeling matrix

相似度矩阵构建结果见图 5。由图 5 可以看出词之间的相似度,如词对(sensor, surveillance)=0.53,说明这两个单词之间的相似度为 0.53,根据这些词(知识点)的相似矩阵继而可以进行下一步的计算,即课程相似度的计算。

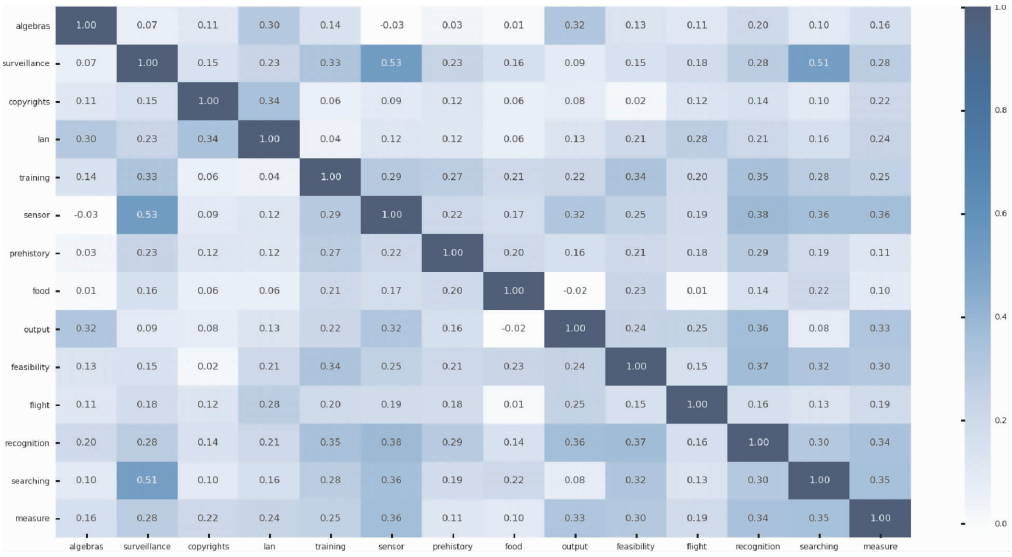

图 5 相似度矩阵构建结果

Fig. 5 Results of similarity matrix construction

课程热力图见图 6,可以看出相似度主要集中在0.3 ~0.7 之间,没有什么异常的数据,可通过课程实际情况大概衡量相似度的准确性。选择一门课程 Cloud Computing,它和 Developing Enterprise Web Applications 相似度最大,相似度为 0.79,和 Research Topics in Computer Science 相似度最小,相似度为 0.41。根据课程描述,可以看出这种方法计算出来的相似度相对比较准确,运算速度较快,符合本文的预期要求。

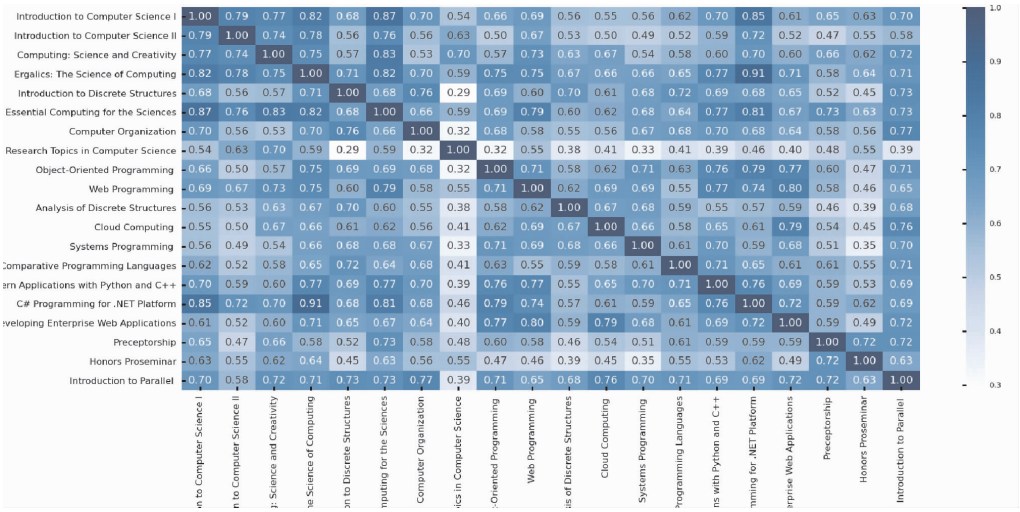

图 6　课程相似度热力图

Fig. 6　Heatmap of similarity between courses

2.3　学校相似度计算

计算学校之间相似度的方法与计算课程相似度矩阵方法类似,即读入所有课程相似度矩阵,然后计算行方向与列方向词的最大值/中位数相似度总和,再求平均值,即为两所学校之间的相似度。

学校相似度计算为计算机专业课程改革提供一定的数据支持,也为学生选择合适的学校提供了一定的参考数据。

2.4　相似度算法运行时间比较

两种算法相似度计算时间对比如图 7 所示。由图 7 可知,新算法比旧算法要快大约 2 ~ 3 倍,且计算相似度课程数目越多时,节约的时间越多。比如当计算 10 所学校的课程相似度时,旧方法需要 22 h,而新方法只需 6 h,节约了 16 h;当计算 30 所学校时,可节约 32 h。这只是部分结果,新算法还需进一步优化。

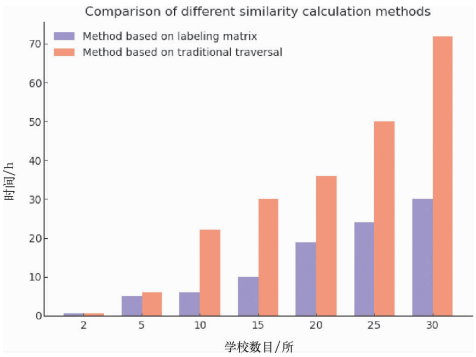

图 7　不同方法相似度计算时间对比

Fig. 7　Comparison of similarity calculation time of different methods

3　聚类分析

3.1　层次聚类

(1)课程层次聚类

采用层次聚类法[13]计算语料库课程之间的相似度,使用前面计算的相似度文件直接聚类,而不需要重新计算距离,并按课程相似度由高到低的顺序来进行排序,排序后重新划分节点。根据选用方法的原理及相似度计算的目标,选用 ward 的方法来进行层次聚类。

由课程层次聚类结果(图 8)可以看出,课程经过层次聚类后被分为几个类,同一个类别的课程分布的距离比较靠近,由此可以看出哪些课程更为相似。

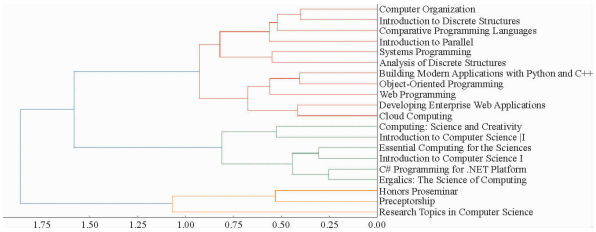

图 8　课程 - 层次聚类树形图

Fig. 8　Courses hierarchical clustering dendrogram

(2)学校层次聚类

学校层次聚类结果(图 9)显示看出各个高校聚类后的分类结果及空间分布,可以判断哪些学校更为相似,从而为学生选择高校提供一定的参考。

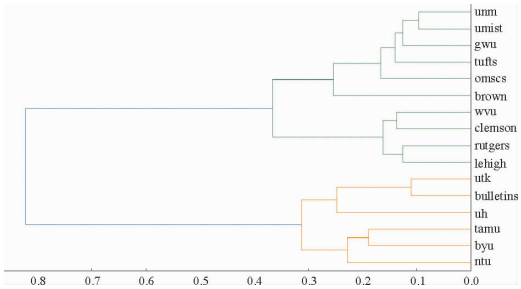

图 9 学校 - 层次聚类树形图

Fig. 9 Schools hierarchical clustering dendrogram

但从高校分布的散点图(图 10)来看,部分同一分类的高校距离较远,比如 wvu(西弗吉尼亚大学)与 brown(布朗大学)在空间上的距离比较远,但是经过聚类却属于同一类别,所以,高校关系的层次聚类效果,并不符合预期结果。

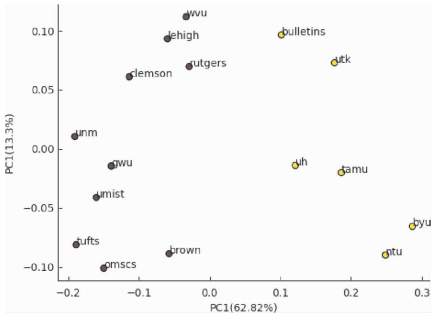

图 10 学校 - 层次聚类散点图

Fig. 10 Schools hierarchical clustering scatter diagram

3.2 K-means 聚类

K-means 算法[14]是依据数据在空间中的分布位置关系将数据聚类为 K 个簇。从原始集群中选择 K 个点作为簇的质心,重新计算质心,样本点选择归入离自己在空间分布最近的簇,计算每个簇的质心,并更新簇的质心。

(1)课程 - K-means 聚类

从课程 - K-means 聚类结果(图 11)可以看出,经过 K-means 聚类的课程关系分布与课程层次聚类效果一样,同一类别的课程在空间上的分布接近,符合预期结果。对于课程的 K-means 聚类来说,K 值的选择是比较重要的,如何选择一个合适的 K 值,需要经过多次合理试验,并根据聚类效果进行选择。

(2)学校 - K-means 聚类

从学校 - K-means 聚类结果(图 12)可以看出,经过 K-means 聚类后各个高校之间的关系与层次聚类效果不同,接下来对这两种方法的结果进行对比分析。

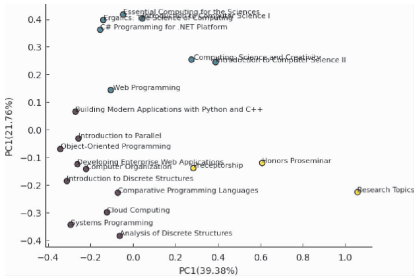

图 11 课程 - K-means 聚类散点图

Fig. 11 Courses K-means clustering scatter diagram

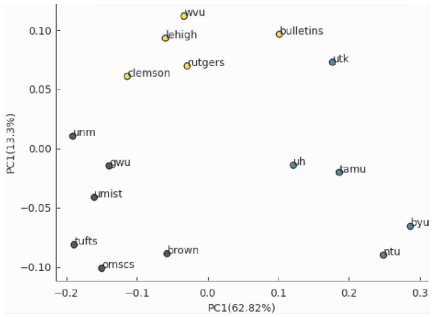

图 12 学校 - K-means 聚类散点图

Fig. 12 Schools K-means clustering scatter diagram

3.3 高校聚类结果分析

由图 10 和图 12 可以看出,两种聚类结果空间的分布大体相同,但由于参数不同以及算法原理不同,最后的聚类分类结果不同,比如在层次聚类中,学校 wvu(西弗吉尼亚大学)与 bulletins(宾夕法尼亚州立大学)是不属于一个类别的,但是在 K-means 聚类中,它们属于同一个类别,而且在 K-means 聚类中,同一类别的高校比层次聚类的高校在距离上更加接近,因此,与层次聚类相比,K-means 聚类的结果更符合预期的结果。

4 结 论

在传统的计算机科学课程描述方法中,存在着很多不足:①缺乏对专业课程间关系的整体结构的量化,只是提及了部分课程的关系,而没有具体量化各课程间的关系;②没有对课程知识点之间关系的量化,无法获得各知识点之间的具体联系;③基于文档描述各个课程,不适合在整体上把握计算机科学专业课程的结构。

针对计算机科学传统描述方法的不足,本文以国外排名靠前的计算机名校的计算机科学课程数据为语料库,通过对课程知识点的特征提取、权重计算和量化表示,介绍了相似度计算的多种方法,通过不同的算法计算课程间的相似度,并对计

算结果进行分析比较,最终选择最符合预期目标的算法。同时,在课程相似度的基础上完成了对学校相似度的计算,并分别利用层次聚类和 K-means 聚类对课程和学校进行聚类分析比较。本文提供了从课程信息数据中发现其中所蕴含的知识、课程体系结构、课程关系计算的方法,为课程学习者提供课程的整体框架和关键知识内容,为改善计算机教育质量提供了科学依据和数据支持。

本文设计了词组相似度计算的新算法,原来的相似度算法直接遍历词向量文件,找出所有课程对中单词对的向量进行相似度计算,这样会导致词对相似度多次重新计算。本文对相似度计算算法进行改进,利用标注矩阵,使每个词对仅需计算一次相似度,从而避免了词对相似度的重复计算,有效提高了相似度计算的速度。新的相似度计算方法具有显著的借鉴价值,有大量的应用场合,比如无向图挖掘时的点对相似度计算,社交网络社区发现,聚类融合时点对相似度比较等。研究过程中遇到了很多问题,比如包括如何快速计算相似度、如何处理高维度向量等,通过对各种算法进行对比,最终确定算法的选择,解决了这些问题,为计算课程知识图谱的构建提供了牢固的数据基础和算法基础。

参考文献:

[1] Baker R S. Educational data mining: An advance for intelligent systems in education[J]. IEEE Intelligent Systems, 2014, 29(3): 78-82.

[2] Manjula M. A systematic review on educational data mining[J]. International Journal of Scientific Research in Science, Engineering and Technology, 2018, 4(4): 164-170.

[3] Thanh-Nhan H L, Nguyen H H, Thai-Nghe N. 2016 Eighth International Conference on Knowledge and Systems Engineering, October 6-8, 2016[C]. Piscataway: IEEE, 2016.

[4] Zhu P, Zhong W, Yao X M. Auto-construction of course knowledge graph based on course knowledge[J]. International Journal of Performability Engineering, 2019, 15(8):2228-2236.

[5] Varagnolo D, Knorn S, Staffas K, et al. Graph-theoretic approaches and tools for quantitatively assessing curricula coherence[J]. European Journal of Engineering Education, 2021, 46(3): 344-363.

[6] Aldrich P R. The curriculum prerequisite network: Modeling the curriculum as a complex system[J]. Biochemistry and Molecular Biology Education, 2015, 43(3): 168-180.

[7] Liang C, Ye J, Wu Z, et al. Proceedings of the 31st AAAI Conference on Artificial Intelligence, Feburary 4-9, 2017[C]. Palo Alto: AAAI, 2017.

[8] Knorn S, Varagnolo D, Staffas K, et al. Quantitative analysis of curricula coherence using directed graphs[J]. IFAC-PapersOnLine, 2019, 52(9): 318-323.

[9] Wengle E, Knorn S, Varagnolo D. COnCUR-COherence in CURricula: A tool to assess, analyze and visualize coherence in higher education curricula[J]. IFAC-PapersOnLine, 2020, 53(2): 17598-17603.

[10] 罗燕, 赵书良, 李晓超,等. 基于词频统计的文本关键词提取方法[J]. 计算机应用, 2016, 36(3):718-725.

[11] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[EB/OL]. (2013-01-16)[2021-06-07]. https://arXiv.org/abs/1301.3781.pdf.

[12] Islam A, Inkpen D Z. Semantic similarity of short texts[M]//Nicolov N, Angelova G, Mitkov R (eds.). Recent Advances in Natural Language Processing V, Amsterdam: John Benjamins Publishing Company,2007.

[13] Ah-Pine J, Wang X. Proceedings of the 15th International Symposium on Intelligent Data Analysis, October 13-15, 2016[C]. Berlin:Springer, 2016.

[14] Alhawarat M, Hegazi M. Revisiting K-means and topic modeling, a comparison study to cluster Arabic documents[J]. IEEE Access, 2018,6:42740-42749.

【责任编辑: 卓祯雨】