计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2022,31(6):10−18 [doi: 10.15888/j.cnki.csa.008568]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于异质数据源的计算机学科知识图谱构建①

李华昱, 刘烨宸, 李家瑞, 闫　阳

(中国石油大学 (华东) 计算机科学与技术学院, 青岛 266580)
通信作者: 刘烨宸, E-mail: 1507010311@s.upc.edu.cn

摘　要: 计算机学科评估需要对学科整体信息进行汇总, 过于依赖专家经验且历届学科评估信息复用程度低. 针对此问题, 该文提出了一种计算机学科评估知识图谱构建方法. 该方法基于 CIR 模型建模知识图谱, 设计了针对文本数据的基于依存句法分析的无监督命名实体关系抽取方法和针对表格的数据流组合模型抽取知识, 并借助 Neo4j 图数据库实现知识图谱可视化, 为更多学科知识图谱的构建提供思路和借鉴.

关键词: 学科评估; 知识抽取; 知识图谱; 可视化分析

引用格式: 李华昱,刘烨宸,李家瑞,闫阳.基于异质数据源的计算机学科知识图谱构建.计算机系统应用,2022,31(6):10–18. http://www.c-s-a.org.cn/1003-3254/8568.html

Construction of Computer Subject Knowledge Graph Based on Heterogeneous Data Sources

LI Hua-Yu, LIU Ye-Chen, LI Jia-Rui, YAN Yang

(College of Computer Science and Technology, China University of Petroleum (East China), Qingdao 266580, China)

Abstract: Computer subject evaluation needs to summarize the overall information of the subject. It relies too much on expert experience, and the reuse degree of previous subject evaluation information is low. In response to this problem, this study proposes a method for constructing a knowledge graph of computer subject evaluation. The method constructs a knowledge graph model based on the CIR model, designs an unsupervised named-entity and relation extraction method based on dependency parsing for text data and a combination model of data flow for table data to extract knowledge, and visualizes the knowledge graph using Neo4j. This model can provide ideas and references for the construction of knowledge graphs of other subjects.

Key words: subject evaluation; knowledge extraction; knowledge graph; visual analysis

知识图谱因其表达知识的方式与人类认知世界的形式具有相似性, 得到越来越广泛的应用. 知识图谱产生初期主要是用作计量分析工具, 现已逐渐与各领域结合作为数据库使用. 得益于其知识结构, 知识图谱可以方便地处理大量数据, 分析数据进而发现研究的趋向. 在教育研究领域, 知识图谱也发挥了重要作用. 教育领域早在 2007 年就引入知识图谱进行计量分析, 到现在已经形成了一大批运用知识图谱方法和工具的教育科研成果. 计算机学科评估知识图谱属于教育知识图谱的一个分支, 主要服务于大学学科评估, 通过将学科知识转化为三元组数据存储在图数据库中, 再经过知识推理等方法实现智能问答系统可以方便地检索分析数据, 因此计算机学科评估知识图谱的研究构建将有助于学科大数据自动化和智能化处理.

知识图谱技术是知识图谱建立和应用的技术. 现阶段中文领域知识图谱的构建多面向半结构化数据, 在缺少标注数据, 数据源结构不规范的情况下, 基于规则或字典的实现命名实体识别, 采用半监督方法实现关系抽取, 如文献 [1] 中采用本体约束和专家分类的方式构建知识图谱. 为了更自动化地构建知识图谱, 本文

① 基金项目: 山东省自然科学基金面上项目 (ZR2020MF140)
收稿时间: 2021-09-14; 修改时间: 2021-10-14; 采用时间: 2021-11-17; csa 在线出版时间: 2022-05-26

基于 CIR 模型, 提出了面向异质多数据源的计算机学科评估知识图谱自动构建方法, 从文本数据的知识抽取和表格数据的三元组合成两个流程进行介绍, 将知识存储在图数据库 Neo4j 中并实现可视化, 该方法可为学科评估知识服务提供依据.

1 相关研究

本文对于 1 121 篇包含“知识图谱”主题词的教育领域文章的统计分析绘制表 1, 可以发现, 早在 2007 年, 知识图谱就出现在教育领域中, 直至 2013 年仍仅得到少量应用. 于 2014 年教育领域知识图谱才开始得到发展, 教育领域知识图谱呈现引进、认可、迅速普及的趋势.

表 1 年度发文量情况表

年份	数量
2007	2
2008	2
2009	4
2010	5
2011	18
2012	19
2013	26
2014	60
2015	62
2016	122
2017	170
2018	272
2019	376

该阶段教育领域知识图谱的研究主要以文献计量分析、聚类分析、图谱分析为主, 旨在发现教育研究热点和趋势、为科研指明方向, 因此并没有将知识图谱作为学科知识点的存储工具, 系统化地构建一个应用为主的教育知识图谱. 目前, 像 Wolframalpha、Freebase 等英文知识图谱规模越来越庞大, XLORE 双语百科知识图谱[2]、百度知心等国内的知识图谱也迅速发展, 但是缺少中文的教育知识图谱. 而在中文知识图谱构建技术方面, 知识的获取越来越趋向于采用机器学习和深度学习结合的方法. 比如基于深度学习的中文生物医学实体知识图谱[3]、基于深度学习的网络信息资源知识图谱[4]、基于深度学习的中文林业知识图谱等. 这些知识图谱都依赖于高质量的学科标注数据, 对于没有数据基础的学科来说借鉴较为困难.

国外教育知识图谱以美国个性化教育平台 Knewton[5] 为代表. Knewton 平台依托知识图谱技术, 覆盖了多学科、多学段的知识. 借助其庞大的知识网络, Knewton 可以诊断学生的认知水平和学习进度, 智能化地为学生推送学习资源和设计学习路线. 这不仅仅是为学生提供知识, 更是对知识图谱技术的深度挖掘, 值得我们学习借鉴. 这是知识图谱技术与教育深度结合的结果. 相比之下, 中国教育知识图谱的发展还位于起步阶段, 大多数教育知识图谱还仅是统计分析文献用于发现研究热点, 国内将知识图谱作为知识存储工具探索智能化教育的有清华大学的 eduKB、互联网教育智能技术及应用国家工程实验室的“唐诗别苑”等. 本文构建计算机学科的教育知识图谱也是对中文教育领域知识图谱的一次探索

2 异质数据分析

2.1 数据来源

本文以第 4 轮学科评估所需的领域相关信息为准, 以中国石油大学 (华东) 计算机专业简况表作为数据支撑, 以部分网络知识作为辅助. 第 4 轮学科评估计算机学科所涉及到的数据主要有以下 4 类:

(1) 师资队伍与资源. 包括教师的基本个人信息、职务、研究领域、所属团队和支撑平台等.

(2) 人才培养质量. 包括教学成果、精品课程、优秀学生信息以及毕业生的相关信息等.

(3) 科研成果. 包括发表的论文、申请的专利专著和科研项目.

(4) 权威网站更新的信息.

从结构上看, 计算机学科评估知识图谱的数据源主要有两种: 表格形式的半结构化数据和文本形式的非结构化数据. 下面将针对两种数据进行数据结构分析.

2.2 计算机学科知识特征分析

2.2.1 半结构化数据结构分析

知识图谱中的半结构化数据是指不符合图数据库的形式关联的数据模型结构, 但包含相关标记的数据, 经过一定的处理可以转换为结构化数据, 如图 1 和图 2. 图 1 为学科评估简表中的支撑平台一表截图, 图 2 为自动抓取自专利网站的专利信息.

图 2 中 4 列分别是专利名称、申请日、申请公布号、公布时间. 要注意的是, 表格的第 0 行为列名, 可以直接作为属性或者关系的名称使用, 而 CSV 文件不含表头, 需要人工补充, 且其是以逗号 (这里是以字符) 分隔, 没有明确的列.

I-3 支撑平台			
序号	平台类别	平台名称	批准部门 (与批文公章一致)
1	其他省部级重点实验室北京市	石油数据挖掘北京市重点实验室	北京市科学技术委员会
2	其他省部级工程研究中心山东省	山东省云应用工程研究中心	山东省发展和改革委员会
3	其他省部级工程研究中心山东省	青岛市随钻仪器及信息处理工程技术研究中心	青岛市科技局

图 1 学科评估简表支撑平台表截图

图 2 抓取自专利网站的专利信息

2.2.2 非结构化数据结构分析

“梁鸿, 博士、教授、北京大学博士后、硕士生导师. 国家教育科研网格二期建设项目专家组成员, 中国计算机学会高级会员”为本知识图谱所要处理的非结构化数据, 全部为文本数据, 来源于中国石油大学 (华东) 官网教师简介, 内容真实可靠. 可以看到这类文本数据十分复杂, 没有固定的结构, 且是很长的句子. 每条数据的描述实体只在数据头部, 之后为其属性或关系, 不再重复点明实体的名称, 如果单纯地采用中文分句方法容易丢失最重要的语义信息. 针对这种数据, 本文先识别并记录一条数据的实体, 用实体与后文语义拼接的方式获取结构化数据, 具体实现将在本文第 4 节中说明.

3 CIR 模型

计算机学科评估知识图谱基于 CIR 模型建模, CIR 模型指计算机学科中的概念 (C), 实例 (I) 和约束 (R).

3.1 概念 (concept)

C 表示概念, 是知识图谱中一组同类单元的抽象表达, 如计算机学科评估知识图谱中的“课程”概念、“团队”概念、“单位”概念等, 它能够唯一标识一个有效单元, 定义了知识图谱的框架, 保障了数据一致性.

(1) 概念 (concept)

$$c : T_c = \{t_1, t_2, \cdots, t_n\}$$

集合中, $t_1, t_2, \cdots, t_n$代表 n 个不同的概念名称, 而这些概念名称都可以统一用概念 c 来表示. 例如“希尔排序”是插入排序的一种, 又称“缩小增量排序”, 其概念就可以表示为:

$$c : T = \{\text{希尔排序}, \text{缩小增量排序}\}$$

(2) 概念集合 (C)

$C = \{c_1, c_2, \cdots, c_m\}$概念集合是由不同的概念组成的, $c_1, c_2, \cdots, c_m$表示 m 个不同的概念. 例如: 课程、团队、单位, 可以表示为: $c = \{\text{课程}, \text{团队}, \text{单位}\}$

3.2 实例 (instance)

I 表示实例的集合, 在计算机学科评估知识图谱中每个概念类都具有多个具体的实例, 比如“学科方向”概念类就包含“数据挖掘”“图像处理”等实例. 实例包括实体、关系实例和属性实例 3 类. 知识图谱以<实体-关系-实体>和<实体-属性-属性值>两种形式的三元组存储知识, 表示为$D = (E, R, S)$, 其中, D 表示知识库, 如 YAGO, YAGO 是由德国马普研究所研制的链接数据库. E 表示知识库中的实体集合, R 表示知识库中的关系集合.

$$T_R = \{t_{R_1}, t_{R_2}, \cdots, t_{R_j}\}$$

关系是指概念与概念之间的联系, 关系集合中 $t_{R_1}, t_{R_2}, \cdots, t_{R_j}$代表 j 个不同的关系实例. 常见的关系有包含、属于、同义等, 则其关系集合表示为:

$$T_R = \{\text{包含}, \text{属于}, \text{同义}\}$$

概念 S 表示知识库中的属性集合, 描述概念所具有的特征.

$$T_S = \{t_{S_1}, t_{S_2}, \cdots, t_{S_n}\}$$

表示属性 S 的 n 个属性值$t_{S_1}, t_{S_2}, \cdots, t_{S_n}$在计算机学科评估知识图谱中为确保数据的严谨性, 属性均为数据属性, 即概念类自身拥有的属性, 比如“教学成果”概念类的“获奖等级”属性中, 数据属性值包括“一等”“二等”“三等”. 需注意的是数据属性并不仅限制于等级或者数字, 像“性别”属性的属性值就为“男”和“女”, 表示为:

$$T_{S=\text{性别}} = \{\text{男}, \text{女}\}$$

3.3 约束 (rule)

在计算机学科评估知识图谱中, 约束可以分为检验型约束和规则型约束. 两者区别见表 2.

以上为计算机学科评估知识图谱 CIR 模型. 在知识图谱实际构建中, 基于概念设计本体, 基于实例构建知识点, 基于约束设计推理规则, 使其更具有层次性和模块性, 数据准确性得到保证, 可扩展性增强, 充分发挥图数据库可推理的优势.

表 2 检验型约束和规则型约束

	检验型约束	规则型约束
区别	规定了已知知识的存储和应用形式	规定了未知知识的挖掘和构造方法
作用	有效地防止数据冗余, 确保数据的规范性和统一性.	可以实现知识图谱的推理, 智能问答和隐含关系挖掘都是靠规则型约束实现的.
举例	检验型约束“‘获奖年度’应与获奖证书名称或内容的年度表述一致”, 避免了因表达不同而产生的奖项重复问题.	根据 (Person, 研究方向, 领域*A*)和 (领域*A*, 属于, 领域*B*)产生推理 (Person, 研究方向, 领域*B*)

4 计算机学科评估知识图谱的构建

领域知识图谱需要融合领域专业知识和高质量的数据, 因此多会采用自顶向下的方式构建, 计算机学科评估知识图谱也采用自顶向下的构建方式, 框架如图 3.

计算机学科评估知识图谱的整体构建流程分为领域本体模型构建、信息抽取、知识融合、知识存储和应用 4 部分.

4.1 领域本体模型构建

根据 CIR 模型, 概念设计规定了本体建模规范. 计算机学科知识图谱包含的一级概念为: $c:T_c=\{$人物, 课程, 团队, 专利, 论文$\}$, 根据一级概念搭建本体模型, 并填充关系和属性. 在领域本体建模过程中, 领域专家的参与可以保证全域 Schema 的权威性. 通过构建领域本体, 可以约束实体关系抽取, 即规定了信息抽取步骤中抽取的实体和关系的类别, 确保知识的质量. 计算机学科部分本体建模如图 4 所示.

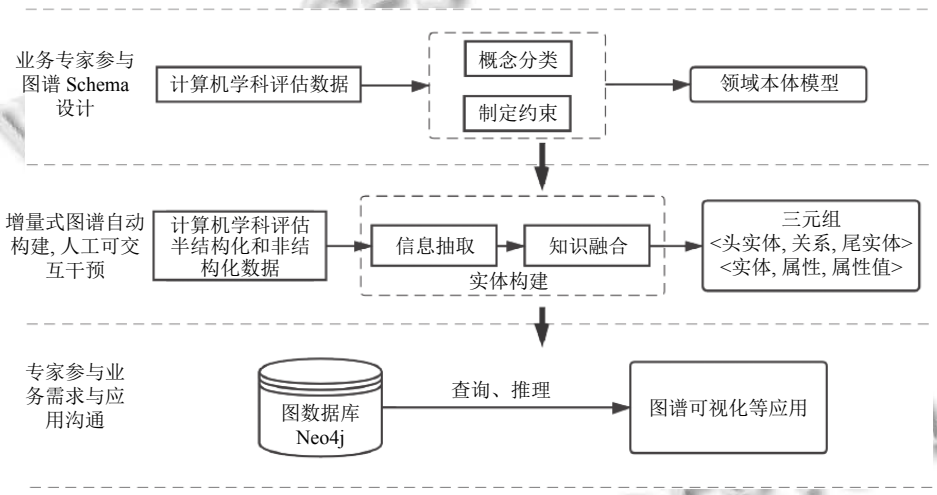


图 3 计算机学科评估知识图谱框架

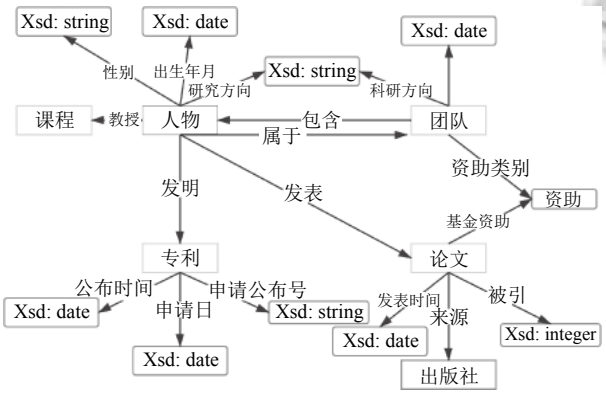


图 4 计算机学科本体模型

4.2 信息抽取

CIR 模型的检验型约束规定本文的源数据有半结构化和非结构化两种, 因此本文设计两种不同的封装器, 对应两种信息抽取方法, 可以把源数据转化为三元组数据. 源数据的存储形式也已经说明, 在此不再赘述, 下面开始介绍本文的两种信息抽取方法.

4.2.1 面向半结构化数据的数据流组合模型

数据流组合模型设计用来实现表格和 CSV 数据的信息抽取, 该方法充分运用数据源中标题和数据的位置特征, 通过制定相应的规则, 从表格中抽取出实体、关系及属性值组合成三元组.

(1) 表格数据三元组转换方法

1) 首先进行数据预处理, 去除无关列, 将表格拆解为单元格, 并为每个单元格分配二维坐标, 设第 1 个标题的坐标为 (x,y), 则其对应的第 1 条表身数据的坐标为 $(x,y+1)$, 第 2 条数据的坐标为 $(x,y+2)$; 第 2 个标题的坐标为 $(x+1,y)$, 其对应的第 1 条表身数据的坐标为

(x+1,y+1), 第 2 条数据的坐标为 (x+1,y+1), 以此类推[6].

2) 根据 CIR 模型的概念对单元格中的信息进行标注, 标注规则为: 对标题进行分类, 若标题属于实体概念, 则其对应的数据标注为实体; 否则被标记为属性值.

3) 根据 CIR 模型规则型约束, 将标注后的数据组合为三元组, 规则如下: 对于表格中同一条数据标注产生的实体 A、实体 B 和属性值 M, 其对应的标题分别为 X、Y、N, 则生成三元组 (A, Y, B) 和 (A, N, M). 下面以图 1 的表格为例进行说明. 转换流程如图 5 所示.

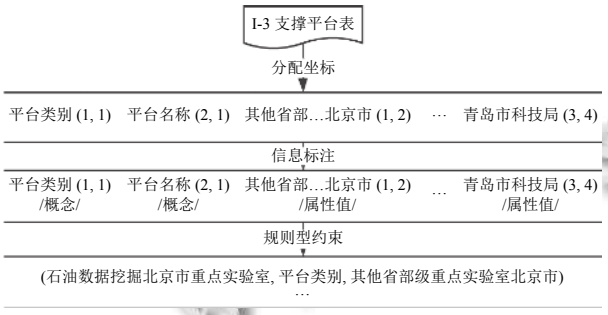


图 5 I-3 支撑平台表的转换说明

(2) CSV 数据三元组转换方法

CSV 数据信息抽取基于 CSV 三元组转换算法实现, 算法实现伪代码如算法 1 所示.

该算法有两个输入, 一是 CSV 数据文件, 另一个是为 CSV 文件预设的每个字段所对应的名称. 如对于一条数据“基于接收概率滑动窗口的船联网分块数据传输方法 CN201910661544.5 2019-07-22 CN110381469A 2019-10-25”预设信息应为“专利名称 (实体) 申请日 申请公布号 公布时间”. 注意, 每一个 CSV 文件应对应一段预设信息, 可以为 CSV 文件排序命名, 将预设信息按顺序统一存储在一个文件中. 拆分数据和预设信息后按照算法进行组合并添加实体字段即可获得三元组数据. 通过此算法可以实现 CSV 数据文件的信息抽取.

算法 1. CSVtoTriple

输入: Information.csv, List Preset
输出: triple

```
function TRAOFCSV(Information.csv, Preset)
    initialization
    file_path = your_path, triple = dict()
        s←split(Preset,"\ t")
        f←with open(Information.csv)
            row←rstrip(row, "\ n")
            list←split(row"\ t")
            entity_str←list[0]
        while entity_str!=" " and list[i] !=" " do
            r←s[i]
            c←list[i]
            triple←TOTRIPLE(triple,entity_str,r,c)
            i++
        end while
    end for
    return triple
end function
function TOTRIPLE(triple,entity_str,r,c)
    for row in f do
    if r !=" " then
        triple['']= [triple,entity_str,r,c]
        triple←AppendToJson().append(file_path,triple)
    else
        catch exception
        throw exception
    end if
    return triple
end function
```

至此, 半结构化数据的信息抽取已经转换完成, 通过此方法获得了所有表格和 CSV 文件中数据的三元组表现形式.

4.2.2 文本信息抽取

(1) 中文句子类型主要类别

中文句子主要类型有陈述句、特殊句、疑问句. 计算机学科评估知识图谱的非结构化数据都是对目标的介绍, 没有不确定性内容也就没有疑问句, 所以这里不予考虑.

通过观察陈述句就可以发现虽然中文语法十分复杂, 但主谓宾等这些句型格式就很适合在实体识别和关系识别后通过调整顺序得到有效的三元组, 据此, 这里参考文献 [7] 中提出的 3 种现象, 结合陈述句的基本句型结构, 可以覆盖经过处理后的计算机学科文本数据的联合抽取. 但是存在一些问题, 上述陈述句句型结构只是对于短句的分析, 对于本文需要处理的源数据“张三主讲过《计算机辅助几何设计》《高级计算机图形学》等研究生课程, 以及《数据结构》《C 语言程序设计》等十多门本科课程, 教学效果优良. 主要研究领域为计算机图形图像、大数据智能处理与云计算等”这类文本信息没有任何句型结构可以利用, 而且前后语义关联性强, 盲目分句容易丢失信息, 所以对于该方法的第一步是选择合适的分句方法, 确保不丢失信息. 下面进行具体说明.

(2) 基于依存句法分析的无监督抽取模型

1) 对于文本长句子, 首先要进行分句. 目前普遍使

用的中文分句方法是找到一个“, . ！ ”这类的典型断句符断开, 而这类方法的发展方向只是考虑更多的符号是否存在断句可能, 这并不适用于本文所要处理的数据. 因为本文所要处理的数据是首先点名实体, 之后全部都是对于该实体的介绍, 如果根据断句符号进行断句, 就会造成分割出来的句子缺失主语的问题. 本文结合中文分句方法, 针对所需处理数据设计了补全主语的分句方法.

首先清洗数据, 去除掉无用的文字或符号. 然后设计字典, 将所有实体 (此处为人名) 登录在字典中, 为其标注类型和链接实体的其他表达方式. 分词时参照字典优先切分实体, 并记录其类型为其词性. 也可以单独标注语料中的所有实体及其位置. 在个人简介数据中, 第一个实体即为整个句子的描述对象, 将其确认为补全后面短句所需的主语, 然后使用正则表达式进行句子分割, 会得到一些缺失主语的短句, 此时将主语补充在缺失主语的谓语之前, 就可以得到一系列分割成功的短句.

2) 优先提取隐藏信息. 中文中 3 种常见的语言现象[7]: ① nominal modification-center (NMC) phenomenon; ② Chinese light verb construction (CLVC) phenomenon; ③ intransitive verb (IV) phenomenon. 由于这 3 种关系在提取过程中容易丢失信息, 所以先对其进行处理. 对于 NMC 现象, 需要将实体与主词链接起来, 提取时直接将主词定义为实体. 而依存分析对于处理 CLVC 现象和 IV 现象有天然的优势.

3) 中文分词[8] 和词性标注. 本文中文分词和词性标注都基于 HMM 模型[9], HMM 模型如图 6, 包含 3 个参数 λ, 状态序列 I 和观测序列 O.

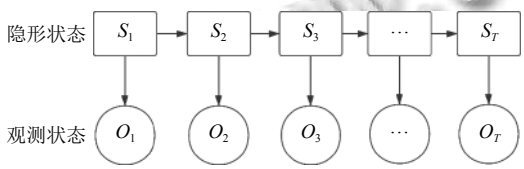


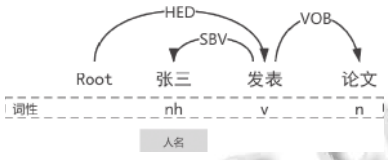
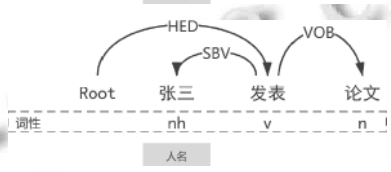
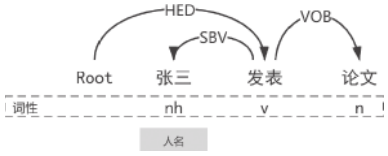
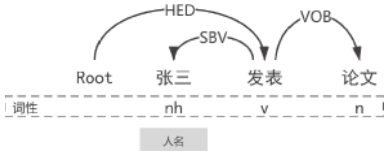
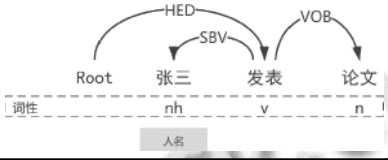
图 6 HMM 模型

基于 HMM 的中文分词过程为: 将待分词语句作为观测序列 O, 训练语料和状态值集合计算得到 λ 的 3 个参数, 使用 Viterbi 算法求解使条件概率 $P(I/O)$ 最优的状态序列 I.

基于 HMM 的词性标注过程为: 将待标注语句作为观测序列 O, 将已知词性集作为状态值集合和训练预料计算得到 λ 的 3 个参数, 使用 Viterbi 算法求解使条件概率 $P(I/O)$ 最优的状态序列 I.

4) 根据词性分析结果, 匹配依存关系生成三元组. 本文规则型约束用到的依存关系与其对应生成三元组规则如表 3.

表 3 依存关系与生成三元组对应表

依存图结构	三元组模式	生成三元组
	(nh, v, n)	(张三, 发表, 论文)
	(nh, v-p, ns)	(张三, 出生于, 青岛)
	(nh, v-p, ns)	(张三, 出生在, 青岛)
	(nh1, v, n) (nh2, v, n)	(张三, 发表, 论文) (李四, 发表, 论文)
	(nh, v, n1) (nh, v, n2)	(张三, 发表, 论文) (张三, 发表, 专著)

以例句“张三发表论文”为例, 存在“发表”与“张三”的 SBV 依存关系和“发表”和“论文”的 VOB 依存关系. 按照实体出现的先后顺序构成的实体对<张三, 论文>的依存句法关系组合为 SBV-VOB, 因此提取得到的三元组为<张三, 发表, 论文>. 基于依存句法分析的无监督命名实体关系抽取具体算法如算法 2, 算法 3 所示.

算法 2. 依存分析

输入: words, postags
输出: triple

```
1: function PARSE(words, postags)
2:     initialization
3:     par_model_path = os.path.join(os.path.dirname(_file_), 'yourpath/parser.model')
4:     parser = Parse()
5:     triple = dict()
6:     parser.load(par_model_path)}, arcs←parser.parse(words, postags)
7:     rely_id←[arc.head for arc in arcs]
```

```
8:    relation←gets[arc.relation for arc in arcs]
9:    heads←['Root' if id == 0 else words[id-1] for id in rely_id]
10:   triple←Totriple(triple, words, heads)
11:   return triple
12: end function
```

算法 3. 基于依存分析的联合抽取模型

输入: words, postags
输出: triple

```
1: function TOTRIPLE(triple, words, heads)
2:    import networkx as nx
3:    G = nx.Graph()
4:    for word in words do
5:       G.add_node(word)
6:    end for
7:    G.add_node('Root')
8:    for i in range(len(words)) do
9:       G.add_edge(words[i], heads[i])
10:   end for
11:   for word in words do
12:      source←word
13:      for word in words do
14:         target←word
15:         if source!= target and distance(source, target)==nx.shortest_
path(G, source, target) then
16:            triple = [source,G.edge,target]
17:            triple←AppendToJson().append(triple)
18:         else
19:            continue
20:         end if
21:      end for
22:   end for
23:   return triple
24: end function
```

至此, 计算机学科评估知识图谱的信息抽取工作全部完成, 获得了通过非结构化和半结构化数据提取出来的三元组.

4.2.3 信息抽取方法评估

为了评估该系统信息抽取的性能, 我们需要确定实验抽取结果的准确率和召回率. 尽管本系统的信息抽取方法是无监督的, 但是仍需标注数据来计算实验抽取结果的准确率和召回率. 为此, 我们人工标注了 2 000 个实体关系对 (半结构化 1 500 个, 非结构化 500 个). 准确率 P 和召回率 R 的计算公式如下:

$$P = C_1/C_2 \tag{1}$$

$$R = C_1/C_3 \tag{2}$$

其中, C_1 表示实验中抽取出来的正确的实体关系对, C_2 表示实验中抽取出来的实体关系对总数, C_3 表示实验数据源中包含的实体关系对数, 我们对 CSV 文件数据、表格文件数据、文本数据分别计算两个参数. 实验结果如表 4 所示.

表 4 实验结果

评估指标	CSV数据	表格数据	文本数据
准确率P	0.976 7	0.965 4	0.826 6
召回率R	0.919 3	0.943 5	0.567 3

从实验结果可以看出, 本系统的面向半结构化数据的数据流组合模型准确率和召回率都可以达到 0.9 以上, 对于非结构化数据的信息抽取, 该方法可以达到较高的准确率, 但召回率只有 0.567 3, 存在信息遗失现象, 经过两次迭代后可以将召回率提升到 0.6 左右, 但继续迭代会导致准确率下降, 因为文本类型的数据中存在少量的抽取模型, 而这些模型却能覆盖大部分语料, 因此继续迭代的意义不大.

4.3 知识融合

上一步已经获得了构建知识图谱所需要的三元组, 但是由于是异质数据源转换而来, 虽然数据的真实性有保障但是仍会存在同一实体存在不同名称的情况. 这时候就需要进行知识融合. 知识融合的总体任务是计算实体间的相似度, 把相似度在一定阈值内的实体划定为同一实体, 本文的处理方法是在同一实体的不同表述间连接 sameAs 关系, 在用户对一个实体进行查询时, 也对其有 sameAs 关系的实体做相同的查询. 下面介绍本文使用的实体相似度计算方法.

对于知识图谱中的节点对, 可以根据其与附近实体的映射关系 (父子关系/兄弟关系) 来计算相似度, 这种相似度被称为结构级相似度[10] . 相似度的计算方法有以下 3 类:

$$sim_S(C_1,C_2) = \mu_P sim_{\text{string}}(SC_1, SC_2) \tag{3}$$

$$sim_B(C_1,C_2) = \mu_B sim_B(BC_1, BC_2) \tag{4}$$

$$sim_R(C_1,C_2) = \mu_R sim_R(RC_1, RC_2) \tag{5}$$

式 (3)、式 (4)、式 (5) 分别是父类、子类和兄弟相似度的计算方法. 对于节点C_1, C_2, 其父类节点分别是SC_1, SC_2, 子类节点分别是BC_1, BC_2, 兄弟节点分别是RC_1, RC_2. μ_P, μ_B, μ_R分别是父类规则、子类规则、兄弟规则中相似度衰减系数. 计算求得以上 3 个相似度后, 通过加权平均的方式进行融合, 最终得到节点间相似度为:

$$sim_{structure}(C_1,C_2)=\frac{\alpha sim_S(C_1,C_2)+\beta sim_B(C_1,C_2)+\gamma sim_R(C_1,C_2)}{\alpha+\beta+\gamma} \quad (6)$$

其中, α, β, γ是加权平均系数, 由于各类规则对于节点影响不同, 通常有$\alpha>\beta>\gamma$.

为了测试知识融合的效果, 本文随机抽取 1 000 个融合后的实体对, 根据其共有属性和关系计算其语义相似度[11] 进行评估, 语义相似度的计算公式是:

$$sim_W(C_1,C_2)=\sum_{\rho(\omega C_1,\omega C_2)\in S(C_1,C_2)}\frac{Wordsim(\omega C_1,\omega C_2)\times \max(idf(\omega C_1),idf(\omega C_2))}{} \quad (7)$$

其中, $Wordsim(\omega C_1,\omega C_2)$表示节点$C_1,C_2$共有属性的词相似度, $idf(\omega C_1)$是ωC_1的逆文本频率指数[12], 经测试发现, 相较于取$idf(\omega C_1)$, $idf(\omega C_2)$中的平均值或最小值, 取最大值时任务效果更好. 实验测得的语义相似度分布图如图 7 所示. 实验显示相似度达到 0.8 以上的实体数量占总数量的 71.3%. 实验证明通过计算结构级相似度进行知识融合的方法效果良好.

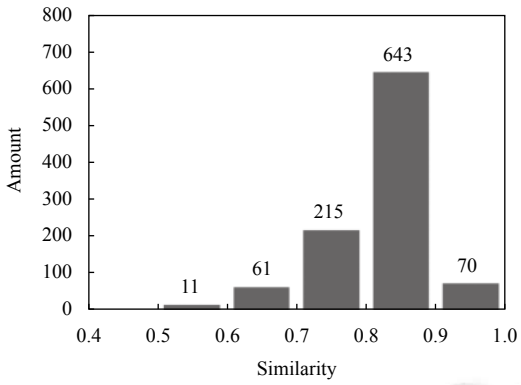


图 7 语义相似度分布图

5 知识存储和可视化

Neo4j 是由 Java 实现的开源 NoSQL 图数据库, 采用自由邻接特性的图存储结构, 能够提供更快的事务处理和数据关系处理能力. 经过信息抽取和知识融合获得的三元组知识使用 Cypher 语言存储到 Neo4j 中, 即完成了知识存储. Neo4j 虽然提供了一个查询与展示一体化的 Web 操作界面, 但 Neo4j 并没有接口允许该图形界面嵌入到公共网页中, 所以本文借助 Echarts 来实现知识图谱可视化[13], Neo4j 仅作为图数据库使用. Echarts 是基于 JavaScript 的开源可视化库, 其自带的关系类型图是在前端实现知识图谱可视化的常见选择, 并且还可以配合 JavaScript 实现力导向图、展现依赖关系、显示属性等功能. 基于 Echarts 实现的计算机学科评估知识图谱可视化如图 8 所示.

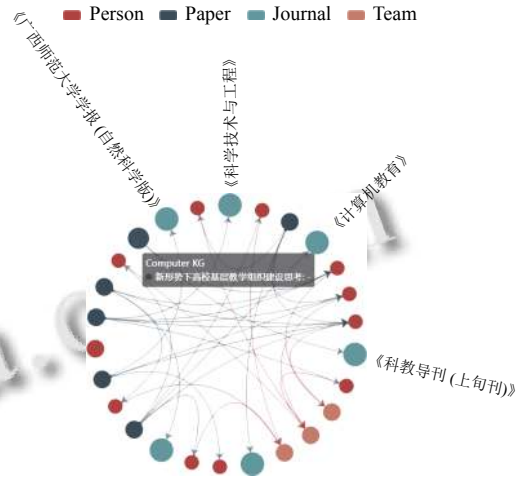


图 8 知识图谱可视化

除此之外, 计算机学科评估知识图谱还提供了命名实体识别、实体关系查询、实体路经查询等功能.

图 9 是本系统的实体路径查询功能的截图展示, 输入两个实体 (不需要输入关系), 查询可返回两个实体间的路径.

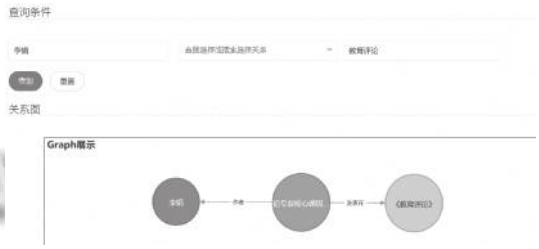


图 9 实体路径查询功能

6 结束语

知识图谱在国内有很大发展潜力, 特别是教育领域, 相较于国外还有较大差距. 本文使用 CIR 模型建模, 对于最常见的文本、表格格式数据, 从本体构建、信息抽取、知识融合、知识存储到图谱可视化, 介绍了一套完整的学科知识图谱构建流程. 通过套用此方法可以提高了构建学科知识图谱的效率, 此方法也可以用在其他学科知识图谱的构建过程中. 同时, 计算机学科评估知识图谱可以方便地查询学科评估信息, 为学科评估提供数据支持和更好的分析数据.

参考文献

1 张春霞, 彭成, 罗妹秋, 等. 数学课程知识图谱构建及其推理. 计算机科学, 2020, 47(S2): 573–578.

2 Jin HL, Li CJ, Zhang J, *et al*. XLORE2: Large-scale cross-lingual knowledge graph construction and application. Data Intelligence, 2019, 1(1): 77–98. [doi: 10.1162/dint_a_00003]

3 丁泽源, 杨志豪, 罗凌, 等. 基于深度学习的中文生物医学实体关系抽取系统. 中文信息学报, 2021, 35(5): 70–76.

4 袁荣亮, 姬忠田. 基于深度学习的网络信息资源知识图谱研究. 情报理论与实践, 2021, 44(5): 173–179.

5 Nosenko Y. Alta solution from Knewton as a tool of support for adaptive learning in mathematics. Educational Discourse: Collection of Scientific Papers, 2020, 28(11): 69–81.

6 林也莉. 面向多源数据的信息抽取方法研究 [硕士学位论文]. 上海: 华东理工大学, 2015.

7 Jia SB, E SJ, Li MZ, *et al*. Chinese open relation extraction and knowledge base establishment. ACM Transactions on Asian and Low-resource Language Information Processing, 2018, 17(3): 15.

8 曹勇刚, 曹羽中, 金茂忠, 等. 面向信息检索的自适应中文分词系统. 软件学报, 2006, 17(3): 356–363.

9 Sweta K, Amutha B. HMM (hidden Markov model) based speech to text conversion for regional language (TAMIL). Artificial Intelligent Systems and Machine Learning, 2011, 3(5): 320–325.

10 王荣波, 池哲儒. 基于词类串的汉语句子结构相似度计算方法. 中文信息学报, 2005, 19(1): 21–29.

11 黄承慧, 印鉴, 侯昉. 一种结合词项语义信息和 TF-IDF 方法的文本相似度量方法. 计算机学报, 2011, 34(5): 856–864.

12 曾斯炎, 周锦, 黄国华. 基于词频-逆文本频率和社区划分的图书推荐算法. 邵阳学院学报 (自然科学版), 2017, 14(2): 19–22, 37.

13 邓宇, 周卫强, 张振铭, 等. 基于名老中医医案的知识图谱构建. 湖南中医杂志, 2019, 35(7): 186–187.