基于关系代数的多源异构数据聚合模型研究

张军欢[1] 庞 正[1] 张 辉[2]
（1.北京航空航天大学经济管理学院，北京 100191；2.北京航空航天大学计算机学院，北京 100191）

摘要：科技资源已成为推动科技进步的关键因素。科技资源的孤岛问题严重阻碍了科技资源的流通及共享，多源数据聚合成为有效解决该问题的关键。针对论文和专利两种不同来源的异构数据展开聚合研究。首先，利用模式匹配方法计算出目标表的结构；其次，利用关系代数的方式对数据调解与整合过程进行建模；最后，利用模型对异构数据进行聚合，得到了聚合的XML数据。在模式匹配中，匹配属性的余弦相似度最高达到0.748，并且聚合结果具有较强的可解释性，验证了该模型的可行性与正确性。

关键词：科技资源；多源异构数据聚合；关系代数；模式匹配；属性相似度
中图分类号：G203 **文献标识码：**A **DOI：**10.3772/j.issn.1674−1544.2021.05.001

Aggregation Model of Heterogeneous Scientific and Technical Resources Using Relational Algebra

ZHANG Junhuan[1], PANG Zheng[1], ZHANG Hui[2]
(1.School of Economics and Management, Beihang University, Beijing 100191; 2.School of Computer Science, Beihang University, Beijing 100191)

Abstract: Scientific and technical resources have become a key factor in promoting scientific and technical progress. The isolated island problem of scientific and technical resources has seriously hindered the circulation and sharing of scientific and technical resources. The aggregation of multi-source data has become the key to effectively solving this problem. This article focuses on the study of the aggregation problem of heterogeneous data from two different sources of papers and patents. First, the structure of the target table is calculated using the pattern matching method, and then the data mediation and integration process is modeled using relational algebra, and finally the model is used. The aggregation of heterogeneous data is realized, which verifies the feasibility and correctness of the model.

Keywords: scientific and technical resources, Multi-source heterogeneous data integration, relational algebra, pattern matching, attribute similarity

作者简介：张军欢（1983—），男，北京航空航天大学副教授，硕士生导师，研究方向为人工智能和区块链在金融和经济的应用（通信作者）；**庞正**（2000—），男，北京航空航天大学学生，研究方向为科技资源管理；**张辉**（1968—）男，北京航空航天大学计算机学院教授，研究方向为人工智能、科技资源管理。

基金项目：重点研发计划项目“分布式科技资源体系及服务评价技术研究”（2017YFB1400200）；重点研发计划项目“跨平台科技资源聚合及规模化服务空间构建”（2018YFB1402904）。
收稿时间：2021 年 3 月 1 日。

0　引言

数据是一种重要的科技资源。近年来，随着数据密集型科研活动快速发展，数据管理的重要性日益上升，进而对科学数据管理和科学数据知识库提出了新的要求[1]，需要考虑更加多源化的数据进行知识库的构建。目前，大数据应用和智能决策的难点之一是多源异构数据融合问题，虽然在文化资源[2]、书目资源[3]等领域已经有了一些研究，但其理论部分依然匮乏，因而在已有理论基础上做进一步的探索是非常必要的[4]。在大数据时代，综合利用和挖掘多源异质异构数据能衍生出新的规律和价值，其实现基础就是数据聚合。数据聚合是一种价值链活动，是将信息收集并标识到更高级别的信息组中的过程[5]。该过程主要需要处理两个方面的问题：一是如何构建更高级别信息组的结构，在本文中更高级别的信息组是数据仓库中的数据表；二是如何将多种低层次信息收集并标识到高级别信息组中，在本文中多个异构数据源属于多个低层次信息收集源，该问题便转化为多源异构数据的整合写入问题。

对于数据表结构的构建，已有很多模式匹配的解决方案[6]，一般可以利用表中实例或属性名来解决数据表结构的构建问题[7]，在模式信息不可用或不足以用于模式匹配时，查找实例的对应关系是一种较好的方法[8]。基于实例的模式匹配方法对实例进行语法和语义分析，并确定属性之间的对应关系[9]。现在已有N-gram、正则表达式、潜在语义分析（LSA）、WordNet、同义词库等基于实例的模式匹配方法，但这些方法通常对匹配数据的实例值有一定限制。而对于数据的整合写入问题，通常是使用ETL工具来解决的。可是现有ETL工具中的ETL流程是利用不同的特定语言定义的[10]，缺乏通用理论建模方面的研究，如Santos等[11-13]提出了一种使用关系代数建模的方法，但该方法只针对单一数据表写入数据仓库的问题，缺少对于异构数据整合过程的建模。

目前，在多源异构数据聚合过程的理论建模研究较为零散，更多的是关注局部问题的解决，缺少专门针对多源异构数据聚合整个过程的理论建模研究。本文首先对文献[8]中提出的一种基于Word2Vec[14-16]的语义比较模式匹配方法进行扩充，在其基础上加入人工规则使得该模型可以进行实例值非字符串属性的模式匹配，进而能够在更大范围内利用模式匹配来辅助数据表结构的构建。在数据整合过程中，本文借鉴了文献[11]—文献[13]关系代数建模数据集成过程的思路，使用关系代数对二源异构数据的整合过程进行建模，从而为多源异构数据的聚合过程提供理论上的指导。

本文在第1节中详细介绍了构建二源异构数据聚合模型的过程；在第2节中介绍了使用论文期刊数据和专利数据对模型的实现过程，并结合人工判断对模型结果进行初步验证；在第3节中得出相关的结论。

1　二源异构数据的聚合模型设计

建模过程主要使用关系代数[17-18]的形式来表达，并参考文献[11]—文献[13]中的方法，结合多源数据整合任务中常用的数据清洗、调解与整合的方法。数据清洗是处理数据中的异常值、缺失值等问题，数据调解是统一数据主键形式的过程，数据整合是多源数据写入到同一数据仓库的过程。本文采用期刊论文和专利两种不同来源的异构数据来完成聚合任务，可将两个数据源分别记作S_p和S_{pa}。聚合多源异构数据首先需要对这些数据进行清洗，然后再利用属性匹配的方法构建聚合完成后的表（以下称为“目标表”）的结构[8]，最后通过数据调解与整合将异构数据写入目标表中。二源异构数据聚合的流程如图1所示。

1.1　相关数据结构定义

从S_p和S_{pa}两个数据源中提取并经过清洗后得到的数据$source_dimdata_{S_p}$和$source_dimdata_{S_{pa}}$分别定义如下。

从数据源S_p中提取的数据表为：

$$source\_dimdata_{S_p}=\left(BK_{S_p},Att_{p_1},\ldots,Att_{p_x}\right)\quad(1)$$

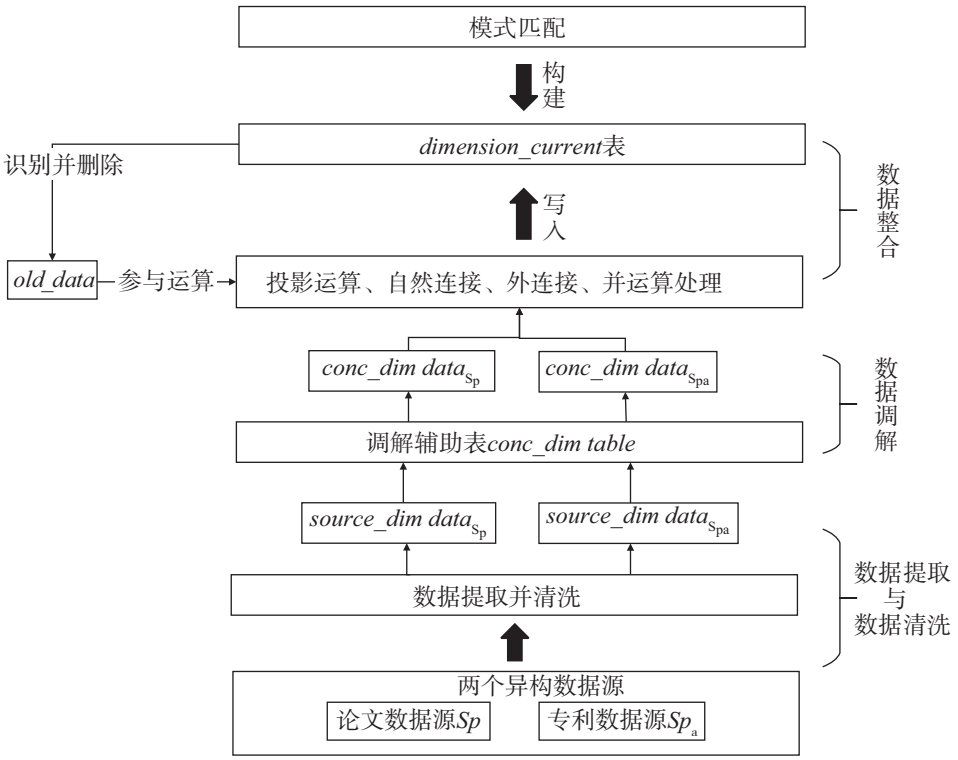

图 1　二源异构数据聚合流程

其中，BK_{S_p}为数据源S_p中数据表的主键，$Att_{p_1},\ldots,Att_{p_x}$表示数据源$S_p$中数据表的非主键属性，$p_1,p_2,\ldots,p_x \in N^+$。

从数据源S_{pa}中提取的数据表为：

$$source\_dimdata_{S_{pa}} = \left(BK_{S_{pa}}, Att_{pa_1},\ldots,Att_{pa_z}\right) \quad (2)$$

其中，$BK_{S_{pa}}$为数据源S_{pa}中数据表的主键，$Att_{pa_1},\ldots,Att_{pa_z}$表示数据源$S_{pa}$中数据表的非主键属性，$pa_1,pa_2,\ldots,pa_z \in N^+$。

1.2　使用属性匹配构建目标表结构

由于在进行属性匹配时不用关注某个属性是否为主键，为叙述方便，本节对$source_dimdata_{S_p}$和$source_dimdata_{S_{pa}}$两张表的结构重新进行如下标记（后文中仍使用 1.1 节中定义的标记方式）：

记表$source_dimdata_{S_p}$为：

$$source\_dimdata_{S_p} = \left(Att_{p_1},\ldots,Att_{p_X}\right) \quad (3)$$

其中，$Att_{p_1},\ldots,Att_{p_X}$为该数据表的全部$X$个属性，并且假设该表中共有$n$个元组，则第$i$个属性$Att_{p_i}$的值向量记为：

$$V_i^p = \left(v_{i_1}^p, v_{i_2}^p,\ldots,v_{i_n}^p\right) \quad (4)$$

其中，$v_{i_j}^p$为第j个元组中在属性Att_{p_i}上的分量。

记表$source_dimdata_{S_{pa}}$为：

$$source\_dimdata_{S_{pa}} = \left(Att_{pa_1},\ldots,Att_{pa_Z}\right) \quad (5)$$

其中，$Att_{pa_1},\ldots,Att_{pa_Z}$为该数据表的全部Z个属性，并且假设该表中共有$m$个元组，则第i个属性$Att_{p_i}$的值向量记为：

$$V_i^{pa} = \left(v_{i_1}^{pa}, v_{i_2}^{pa},\ldots,v_{i_m}^{pa}\right) \quad (6)$$

其中，$v_{i_j}^{pa}$为第j个元组中在属性Att_{pa_i}上的分量。

假设两表中所有值均为字符串，以数据表$source_dimdata_{S_p}$的任意一个属性Att_{p_i}为例，使用Word2Vec将表中任意属性Att_{p_i}值向量V_i^p的每个分量（实例值）转换为数值向量，若实例值为空则置为零向量，之后加权求和得到属性的数值向量VA_{p_i}，记为：

$$VA_{p_i} = \sum_{j=1}^{n} Word2Vec\left(v_{i_j}^p\right) \times \frac{K_j}{n} \quad (7)$$

其中，K_j为第j个实例值$v_{i_j}^p$在V_i^p中出现的次数，n为表中元组数。

利用该转换方法对两张表中所有属性进行操作，则可以得到论文属性Att_{p_i}的数值向量VA_{p_i}和专利属性Att_{pa_i}对应的数值向量VA_{pa_i}。

记Att_{p_i}和Att_{pa_j}之间的余弦相似度为：

$$S_{i,j}=\frac{VA_{p_i}\cdot VA_{pa_j}}{VA_{p_i}\cdot VA_{pa_j}} \tag{8}$$

其中，VA_{p_i}和VA_{pa_j}为向量VA_{p_i}和VA_{pa_j}的二范数。

设阈值为a，则当$S_{i,j}>a$时认为属性Att_{p_i}与属性Att_{pa_j}匹配，可以考虑在目标数据表中将这两个属性合并为一个属性。使用该方法计算出两张表之间的匹配属性对，以此为依据构建目标数据表的结构。

鉴于实例值不为字符串且属性的实例值通常使用字母、数字以一定规则编码而成的属性，称之为编码类属性。此类属性结构精简、信息密度大，并且缺少能够展现其语义的语料，难以通过Word2Vec等算法得到其对应数值向量。为解决这种情况，可以使用人工制定规则的方式进行处理。人工制定规则的具体方法需要根据实际情况而定。本文涉及的具体情形及制定的相应规将在第2.2节中介绍，在这里提供一种思想以供参考。

假定此处考虑的数据均有一定的含义，由于编码类属性的语义通常蕴含在其特定格式中，故要制定的规则即为格式的规则，识别出拥有相似格式的属性将得到匹配。

将两种方式结合，构建出具有相同关系模式的两张目标表*dimension_current*和*dimension_history*，定义如下：

$$dimension\_current=\left(SK,A_1,\ldots,A_m,DateFrom,DateTo\right) \tag{9}$$

$$dimension\_history=\left(SK,A_1,\ldots,A_m,DateFrom,DateTo\right) \tag{10}$$

其中，*SK*为目标数据仓库W中数据表的主键，$A_1,\ldots,A_m$为表中的非主键属性，$m\in N^+$，并且有：

$$\begin{aligned}&\{\text{close}=Att_{p_1},\ldots,Att_{p_x}\}\cup\\&\{\text{close}=Att_{pa_1},\ldots,Att_{pa_z}\}\subseteq\\&\{\text{close}=A_1,\ldots,A_m\}\end{aligned} \tag{11}$$

由于本文仅涉及目标数据仓库中的这两张表，忽略了其他可能存在的无关数据表。*dimension_current*用于记录数据仓库W中的最新数据，*dimension_history*用于记录数据仓库W中的历史数据，*DateFrom*用于记录数据采集的时间（在下文中也使用*Start_Date*表示），*DateTo*用于记录数据在*dimension_current*表中的最后时间（在下文中也使用*End_Date*表示）。

1.3　数据调解与整合

1.3.1　调解阶段

由于从S_p,S_{pa}中提取的数据之间主键的不同会影响目标数据仓库W中关系模式的统一，通过数据调解将原关系模式的主键映射为数据仓库中目标关系模式的主键。首先需要人工制定一张调解辅助表*conc_dimtable*。该表定义为：

$$conc\_dimtable=\left(\text{SK},\text{BK}_{S_p},BK_{S_{pa}}\right) \tag{12}$$

将两数据表分别与调节辅助表conc _ dimtable进行自然连接运算：

$$Temp1_p\leftarrow source\_dimdata_{S_p}\infty conc\_dimtable$$

$$Temp1_{pa}\leftarrow source\_dimdata_{S_{pa}}\infty conc\_dimtable$$

$Temp1_p$和$Temp1_{pa}$中的元组，可称之为匹配成功的数据。对于这些数据，通过投影运算分别提取所需属性：

$$conc\_dimdata_{S_p}\leftarrow\pi_{\left(SK,Att_{p_1},\ldots,Att_{p_x}\right)}\left(Temp1_p\right) \tag{13}$$

$$conc\_dimdata_{S_{pa}}\leftarrow\pi_{\left(SK,Att_{pa_1},\ldots,Att_{pa_z}\right)}\left(Temp1_{pa}\right) \tag{14}$$

此外，某些原因导致$source_dimdata_{S_p}$、$source_dimdata_{S_{pa}}$中出现了新的主键或主键属性，使得某些数据的主键在调解辅助表中不存在。这些数据无法出现在$Temp1_p$和$Temp1_{pa}$中，称之为未匹配成功的数据。首先提取这些数据：

$$new\_dimdata_{S_p}=source\_dimdata_{S_p}-\pi_{\left(BK_{S_p},Att_{p_1},\ldots,Att_{p_x}\right)}\left(Temp1_p\right) \tag{17}$$

$$new\_dimdata_{S_{pa}}=source\_dimdata_{S_{pa}}-\pi_{\left(BK_{S_{pa}},Att_{pa_1},\ldots,Att_{pa_z}\right)}\left(Temp1_{pa}\right) \tag{18}$$

之后通过人工增加$conc_dimtable$中元组的处理方式，使得$new_dimdata_{S_p}$、$new_dimdata_{S_{pa}}$能够与$conc_dimtable$表连接成功，进而将这部分数据分别写入$conc_dimdata_{S_p}$、$conc_dimdata_{S_{pa}}$中。

1.3.2 整合阶段

在$dimension_current$中可能已经存在一部分数据与新数据有所重复，因此整合阶段的任务可以分为两种。一是使用最新提取的数据对dimension_current中的数据进行更新，称之为更新数据任务；二是将目标数据仓库中不存在的新数据写入$dimension_current$中，称之为写入新数据任务。两个任务的过程有所不同。

（1）更新数据

首先识别需要被更新的数据old_data：

$$Temp2 \leftarrow \pi_{(SK)}\left(conc\_dimdata_{S_p}\right) \cup \pi_{(SK)}\left(conc\_dimdata_{S_{pa}}\right) \quad (19)$$

$$old\_data \leftarrow Temp2 \infty dimension\_current \quad (20)$$

之后把old_data从$dimension_current$表移入$dimension_history$中：

$$dimension\_current \leftarrow dimension\_current - old\_data \quad (21)$$

$$dimension\_history \leftarrow dimension\_history \cup \varepsilon_{(End\_Date=today()-1)}\left(\pi_{(*)}\left(old\_data\right)\right) \quad (22)$$

$$*=\{d \mid d \in \{\mathrm{SK}, A_1, \ldots, A_m, DateFrom\}\} \quad (23)$$

其中，$today()$指更新数据当天的日期，使用ε函数标记为属性End_Date赋值为$today()-1$（在下文中同理，不再赘述）。

假设按照数据源S_p和S_{pa}的顺序依次进行数据更新。

使用数据源S_p的数据对$dimension_current$中属性$Att_{p_1}, \ldots, Att_{p_x}$更新：

$$Temp3 \leftarrow \varepsilon_{(Start\_Date=today())}\left(\pi_{(*)}\left(old\_data\right) \infty \mathrm{conc\_dimdata}_{S_p}\right) \quad (24)$$

$$dimension\_current \leftarrow dimension\_current \cup Temp3 \quad (25)$$

$$*=\{d \mid d \in \{SK, A_1, \ldots, A_m, DateTo\} \wedge d \notin \{Att_{p_1}, \ldots, Att_{p_x}\}\} \quad (26)$$

使用数据源S_{pa}的数据对$dimension_current$中属性$Att_{pa_1}, \ldots, Att_{pa_z}$更新：

$$Temp5 \leftarrow \varepsilon_{(Start\_Date=today())}\left(\pi_{(**)}\left(old\_data\right) conc\_dimdata_{S_{pa}}\right) \quad (27)$$

$$dimension\_current \leftarrow dimension\_current \cup Temp4 \quad (28)$$

$$**=\{d \mid d \in \{\mathrm{SK}, A_1, \ldots, A_m, DateTo\} \wedge d \notin \{Att_{pa_1}, \ldots, Att_{pa_z}\}\} \quad (29)$$

（2）写入新数据

该任务仍假设按数据源S_p和S_{pa}的顺序进行处理，具体过程如下。

对于数据源S_p，首先从$conc_dimdata_{S_p}$中识别出$dimension_current$中不存在的新数据：

$$Temp6 \leftarrow conc\_dimdata_{S_p} - \pi_{(*)}\left(dimension\_current\right) \quad (30)$$

$$*=\{d \mid d \in \{SK, Att_{p_1}, \ldots, Att_{p_x}\}\} \quad (31)$$

由于从数据源S_p中提取的属性并不是全部的$A_1, \ldots, A_m$属性，需要增加除属性$Att_{p_1}, \ldots, Att_{p_x}$外在$A_1, \ldots, A_m$中的所有属性以及$End_Date$属性并将其设置为空值：

$$Temp7 \leftarrow \varepsilon_{(Start\_Date=today())}\left(Temp6\right) \times dimension\_current \quad (32)$$

最后将新数据写入$dimension_current$中：

$$Temp8 \leftarrow \pi_{(SK, A_1, \ldots, A_m, Start\_Date, End\_Date)}\left(Temp7\right) \quad (33)$$

$$dimension\_current \leftarrow dimension\_current \cup Temp8 \quad (34)$$

对于数据源S_{pa}，需要从$conc_dimdata_{S_{pa}}$中识别出$dimension_current$中不存在的新数据：

$$Temp9 \leftarrow conc\_dimdata_{S_{pa}} - \pi_{(**)}\left(dimension\_current\right) \quad (35)$$

$$*=\{d \mid d \in \{SK, Att_{pa_1}, \ldots, Att_{pa_z}\}\} \quad (36)$$

与S_p中的处理类似，增加除属性$Att_{pa_1}, \ldots, Att_{pa_z}$外在$A_1, \ldots, A_m$中的所有属性并将其设置为空值：

$$Temp10 \leftarrow \varepsilon_{(Start\_Date=today())}\left(Temp9\right) \times dimension\_current \quad (37)$$

最后将新数据写入：

$$Temp11 \leftarrow \pi_{(SK, A_1, \ldots, A_m, Start\_Date, End\_Date)}(Temp10) \quad (38)$$

$$dimension\_current \leftarrow dimension\_current \cup Temp11 \quad (39)$$

2　结果分析

本文将使用论文数据和专利数据来实现以上建模过程。

2.1　数据描述

本文使用的数据已经过清洗。其中，论文数据$source_dimdata_{S_p}$以XML格式存储，各叶节点含义如表1所示，其树结构如图2所示。由于数据中fundlist节点均为空值，在下文中计算目标表结构时不再考虑，但为不破坏数据的完整性，在最终目标表结构中仍会体现出来。专利数据$source_dimdata_{S_{pa}}$以EXCEL文档的形式存储，其属性及含义如表2所示，数据中摘要属性均为空值，处理方式与论文数据中空值节点的处理相同。

2.2　目标表结构的计算

论文期刊数据和专利数据中同时包含纯字符

表1　论文数据节点含义

节点名	含义	节点名	含义
paper_id	论文编号	issue	收录期刊期号
title	题目	journal_id	期刊编号
Keywords	关键词	issn	国际标准连续出版物号
language	使用语言	host_title	期刊名称
classification	中图分类号	cn	国内统一刊号
page_string	页数	author_id	作者编号
abstract	摘要	author_sequence	本文作者序号
issue_id	收录编号	author_name	作者姓名
year	收录年份	affiliation	作者所属机构
volume	收录期刊卷号	fundlist	论文所属基金

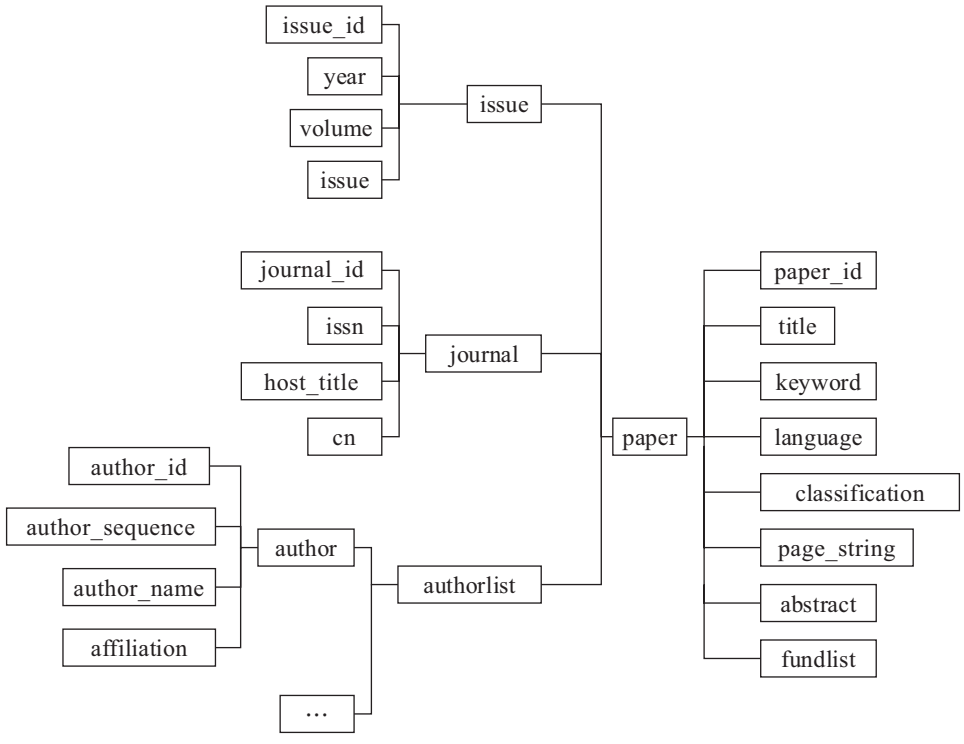

图2　论文数据结构

表 2　专利数据属性含义

属性名	含义	属性名	含义
申请号	专利申请编号	洛伦加号	洛迦诺分类号
申请日	专利申请日期	同族	同族号
公开号	专利公开编号	优先权号	优先权编号
公开日	专利公开日期	优先权日	优先权获得日期
专利类型	专利类型	当前法律状态	专利当前法律状态
申请人	专利申请人	法律状态公布日	法律状态公布日
国别	专利申请地所在国	专利权人	专利权拥有人
省份	专利申请地所在省份	发明人	专利发明人
城市	专利申请地所在城市	地址	发明人地址
主分类号	专利主分类号	代理人	专利代理人
IPC分类号	IPC分类号	代理机构	代理人所在机构
摘要	专利摘要	标题	专利标题

串属性和编码类属性，在 2.2.1 中采用第 1.2 节模型中的属性匹配方法对前者进行匹配，在 2.2.2 节中通过人工设定三类规则处理后者，最终计算出目标表*dimension _ current*的结构。由于本文只是简单验证，为了操作上的简洁省略了对模型中*DateFrom*,*DateTo*属性的记录。此外，XML 文件中子节点记录的信息必然与其父节点存在较强关联，因此本文认为当某一节点的任意子节点与某一属性产生匹配时，该节点整体也有较大的概率与该属性存在匹配，可以记录为该节点与该属性匹配。

2.2.1　纯字符串属性间的匹配

筛选论文数据和专利数据中纯字符串属性进行匹配。本文将专利属性申请人、专利权人、地址、发明人、代理人、代理机构中的人名与机构名分别组成人名和机构两个新属性来替代原有的 6 个属性。计算词向量时采用了预训练的 Word2Vec 模型。该模型使用百度百科等语料进行训练得到，计算相似度的结果如表 3 所示。

如果某一属性对的相似度同时为所在行和所在列的最大值，则认为该属性对为匹配属性；若相应属性已被其他匹配占据，则进行人工选取。匹配结果如表 4 所示，其中 50% 的论文纯字符串属性可以通过计算直接得到匹配，50% 的属性需要在计算结果的基础上进行人工选取，而对于 abstract 和 host_title 属性，本文认为没有专利的纯字符串属性可以与其匹配。

论文的 author_name 属性和专利的人名属性、affliation 属性及专利的机构属性匹配，就是将论文的 authorlist 属性和组成专利人名、机构的 6 个属性整合到一起。

2.2.2　编码类属性间的匹配

（1）日期类属性匹配

由于常有一些数据用于记录相关日期的属性，基于其功能的相似性，本文认为这类属性应该匹配到一起。这类属性通常有两种特点：一是

表 3　纯字符串属性余弦相似度

属性名	title	Keywords	abstract	host_title	author_name	affiliation
专利类型	0.360	0.331	0.415	0.249	0.013	0.231
省份	−0.146	−0.141	−0.099	−0.151	0.136	0.166
城市	−0.147	−0.155	−0.117	−0.100	0.112	0.233
法律状态	0.335	0.334	0.424	0.199	−0.029	0.189
机构	0.152	0.150	0.174	0.062	0.168	0.395
人名	0.186	0.183	0.155	0.177	0.675	0.266
标题	0.748	0.715	0.651	0.358	−0.025	0.291

属性名称中含有“日”“date”等字符，二是实例值的格式为“年/月/日”等。将这两种特点设定为识别日期类属性的规则。其结果见表5。

（2）类别类属性匹配

科技资源中一般会含有用于分类的数据，如“中图分类号”“IPC分类号”等。本文希望将这类属性匹配在一起。由于这类数据具有已知的特定格式，并且这些通用分类号通常都会存在于科技资源中，本文可以直接根据这些可能存在的分类号的特定格式识别这类数据。其结果见表6。

（3）ID类属性匹配

ID类属性是为管理方便而给管理对象分配的一个唯一标识符。利用这类属性可以搜索到唯一与其对应的数据记录。这类属性通常具有唯一性和特定的编码格式。由于编码格式在不同情境下相差极大，从一组实例值中通过发现可能存在的特定格式来识别出可能的ID类属性较为困难，本文仅使用两种规则来识别ID类属性：一是这类属性的名称中常含有“id”“号”等字符，二是这类属性的实例值一定具有唯一性。由于规则较为简单，为了能够与其他属性区分开，需要先实现其他规则，然后才能使用该规则识别剩余的属性。其结果见表7。

表4　纯字符串属性匹配结果

属性（论文属性）	匹配属性（专利属性）	相似度
title	标题	0.748
Keywords	专利类型	0.331
abstract	无	—
host_title	无	—
author_name	人名	0.675
affliation	机构	0.395

表5　日期类属性匹配结果

属性（论文属性）	匹配属性（专利属性）
issue_id,year,volume,issue	申请日、公开日、优先权日、法律状态公布日

表6　类别类属性匹配结果

属性（论文属性）	匹配属性（专利属性）
classification	主分类号、IPC分类号、洛加诺号

表7　ID类属性匹配结果

属性（论文属性）	匹配属性（专利属性）
paper_id	申请号、公开号、专利权号

至此，便完成了两种类型属性匹配的计算。论文数据中有32%的纯字符串属性，68%的编码类属性，共有74%的属性可以进行计算，可以进行计算的属性中86%的属性可以得到匹配。专利数据中有48%的纯字符串属性，52%的编码类属性，共有91%的属性可以进行计算，可以进行计算的属性中86%的属性可以得到匹配。

2.3　目标表结构的验证

在2.2中本文只是得到了匹配的属性，只有进行如检查各匹配属性合理性、加入未匹配属性、目标表中属性命名等人工调整才能得到最终的目标表结构。由于论文数据为多级结构，并且其中部分节点数量不确定，这使得XML的形式更适合表示此类数据。其中每一条记录作为一个resource节点，相关属性为其子节点，每个属性下的数据对应论文数据或专利数据中的匹配属性。resource的一级子节点如图3所示，即目标表的$SK, A_1, \ldots, A_m$属性，其中id为SK，m的值为12。

结果表明，本文提出的目标表结构计算方法可以对超过半数的属性进行计算，并且大多数可计算的属性都可以得到匹配并且经验证具有合理性，说明该方法计算相似度可以对目标表构建起到较好的辅助作用。但仍存在一些属性只能通过人工匹配来实现目标表结构的构建。其原因在于，属性匹配时通常希望将功能相近的属性匹配

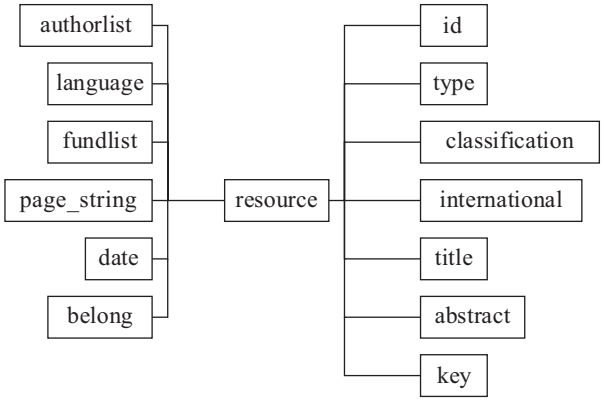

图3　resource一级节点结构

在一起，而这些属性或应与其匹配的属性的功能难以通过计算展现。

属性的功能一般包含在实例值语义或与整条记录间的关系中。前者在文中体现为编码类属性的语义识别困难，其语义只有通过对特定格式的转换才能进行识别，而这类属性通常又极为精简，目前难以通过一种普适性算法发现编码类属性实例值中可能存在的格式进而识别出相应的语义。对于后者，以文中论文属性“journal”与专利属性“国别”“省份”“城市”为例。这两组属性与论文或专利之间都有一种归属关系：论文归属于对应的期刊，而专利归属于对应的申请地点，但这种归属关系目前是难以通过计算得到的。因此，在一般情况下虽然属性匹配计算可以对目标表结构的构建起到一定的辅助作用，但不同程度的人工干预仍是必不可少的。

2.4 数据整合

在调解过程中，为了操作方便，本文将id的结构设计为（paper_id，申请号，公开号，优先权号）的形式，如果数据不存在其中的某一项则置空。这样，就相当于自动生成了模型中的辅助调制表*conc_dimtable*。整合结果以XML格式保存，部分结果如下。

```
<resource>
  <id>503994</id>
  <type>paper</type>
  <classification>TP391.43</classification>
  <international/>
  <title>多链ANN-HMM识别模型及其应用</title>
  <abstract/>
  <key>;隐马尔可夫模型;;人工神经网络;;混合识别系统;;多马尔可夫链</key>
  <page_string>1</page_string>
  <language>chi</language>
  <authorlist>
    <author>
      <author_id>1602044</author_id>
      <author_sequence>1</author_sequence>
      <author_name>姚丹霖</author_name>
      <affiliation>国防科技大学计算机科学系</affiliation>
    </author>
    <author>
      <author_id>1602045</author_id>
  <author_sequence>2</author_sequence>
  <author_name>陈火旺</author_name>
  <affiliation>国防科技大学计算机科学系</affiliation>
  </author>
  <author>
  <author_id>1602046</author_id>
  <author_sequence>3</author_sequence>
  <author_name>殷建平</author_name>
  <affiliation>国防科技大学计算机科学系</affiliation>
  </author>
  </authorlist>
  <date>
  <issue_id>26770</issue_id>
  <year>2000</year>
  <volume>13</volume>
  <issue>1</issue>
  </date>
  <belong>
  <journal_id>1024</journal_id>
  <issn>1003-6059</issn>
  <host_title>模式识别与人工智能</host_title>
  <cn>34-1089/TP</cn>
  </belong>
  <fundlist/>
</resource>
```

利用第1.3节中提出的模型将原本结构完全不同的二源数据聚合为拥有基本相同结构的数据，并且这种聚合明显具有高度可解释性。如聚合到authorlist节点下的论文作者和专利申请人等属性均包含了科技资源数据相关人的信息以及具有的相似功能。统一的数据结构极大地方便了数据的使用。研究人员可以通过相似的调用方法获取不同来源的数据。

3 结论

对科技资源进行数据聚合是解决科技资源多源异构问题的有效方法，良好的数据聚合有助于对多源异构科技资源高效地综合利用。本文成功地对二源异构数据聚合过程中数据仓库的表结构构建、数据调解与整合过程进行了建模，并利用该模型实现论文数据和专利数据二源异构数据的聚合，验证了其可行性。该模型没有局限于数据聚合中的数据整合部分，而是对数据聚合的整个流程进行了建模。这在一定程度上弥补了此前研究相对局限于部分流程的问题，使用更加完整的数据聚合模型可以为整个过程提供更完善的理论指导，使得在不同情境下的多源数据聚合能够更加方便地构建整个流程框架。

在未来工作中，改进模型中模式匹配的方法，减少属性匹配计算中对属性实例值类型的限制，建立结果更加可靠、更少人工处理的多源异构数据聚合模型。此外，将图书数据纳入实现时的考虑范围，验证该模型在更加复杂的科技资源数据聚合情境下的可行性。

参考文献

[1] 刘峰, 张晓林, 孔丽华. 科研数据知识库研究述评[J]. 现代图书情报技术, 2014(2): 25−31.

[2] 刘盼雨, 王昊天, 郑栋毅, 等.多源异构文化大数据融合平台设计[J/OL].华中科技大学学报(自然科学版): 1−8[2021−02−25].https: //doi.org/10.13245/j.hust.210216.

[3] 卫宇辉.基于细粒度聚合单元元数据的书目资源聚合研究[J].国家图书馆学刊, 2020, 29(6): 90−101.

[4] 唐伟, 翟国锋, 谷红娟. 大数据背景下的多来源数据融合研究[J]. 统计与管理, 2019(5): 6.

[5] FRANKLIN E W. Data fusion lexicon[R]. Washington DC: Joint Directors of Labs, 1991.

[6] IONESCU A. Reproducing state−of−the−art schema matching algorithms[D]. Delft: Delft University Of Technology, 2020.

[7] SAHAY T, MEHTA A, JADON S. Schema matching using machine learning[C]//2020 7th International Conference on Signal Processing and Integrated Networks (SPIN). Noida: IEEE, 2020: 359−366.

[8] NOZAKI K, HOCHI T, NOMIYA H. Semantic schema matching for string attribute with word vectors and its evaluation[J]. International journal of networked and distributed computing, 2019, 7(3): 100−106.

[9] ALWAN A A, NORDIN A, ALZEBER M, et al. A survey of schema matching research using database schemas and instances[J]. International journal of advanced computer science and applications, 2017, 8(10): 102−111.

[10] AWITI J, VAISMAN A A, ZIMÁNYI E. Design and implementation of ETL processes using BPMN and relational algebra[J]. Data & knowledge engineering, 2020, 129: 101837.

[11] SANTOS V, BELO O. Modeling ETL data quality enforcement tasks using relational algebra operators[J]. Procedia technology, 2013(9): 442−450.

[12] SANTOS V, BELO O. Modelling ETL conciliation tasks using relational algebra operators[C]//2014 European Modelling Symposium. Pisa: IEEE, 2014: 275−280.

[13] SANTOS V, BELO O. Using relational algebra on the specification of real world ETL processes[C]//2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing. Liverpool: IEEE, 2015: 861−866.

[14] MIKOLOV T, CHEN K, CORRADO G, et al. Efficient estimation of word representations in vector space[J]. arXiv preprint arXiv: 1301.3781, 2013.

[15] MIKOLOV T, YIH W, ZWEIG G. Linguistic regularities in continuous space word representations[C]//Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: Human language technologies. Atlanta: ACL, 2013: 746−751.

[16] MIKOLOV T, SUTSKEVER I, CHEN K, et al. Distributed representations of words and phrases and their compositionality[C]//Advances in neural information processing systems. Lake Tahoe: NIPS, 2013: 3111−3119.

[17] CODD E F. A relational model of data for large shared data banks[M]//Software pioneers. Berlin, Heidelberg: Springer, 2002: 263−294.

[18] CODD E F. Extending the database relational model to capture more meaning[J]. ACM Transactions on Database Systems (TODS), 1979, 4(4): 397−434.