计算机科学
COMPUTER SCIENCE

一种增量式本体模型与数据模式映射的图谱实例模型构建演化方法

单中原, 杨恺, 赵俊峰, 王亚沙, 徐涌鑫

引用本文

单中原, 杨恺, 赵俊峰, 王亚沙, 徐涌鑫.一种增量式本体模型与数据模式映射的图谱实例模型构建演化方法[J]. 计算机科学, 2023, 50(1): 18-24.

SHAN Zhongyuan, YANG Kai, ZHAO Junfeng, WANG Yasha, XU Yongxin. Ontology-Schema Mapping Based Incremental Entity Model Construction and Evolution Approach of Knowledge Graph [J]. Computer Science, 2023, 50(1): 18-24.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.220500205

一种增量式本体模型与数据模式映射的图谱实例模型构建演化方法

单中原[1,2]　杨　恺[1,2]　赵俊峰[1,2,3]　王亚沙[1,2,3]　徐涌鑫[1,2]
1 北京大学计算机学院　北京 100871
2 高可信软件技术教育部重点实验室　北京 100871
3 北京大学(天津滨海)新一代信息技术研究院　天津 300450
(1901213329@pku.edu.cn)

摘　要　在智慧城市领域中,随着信息化技术的不断深入,各信息系统产生的海量数据不断增长,这些多源异构数据之间的语义互通成为了城市智能应用开发需要解决的重要问题之一。构建知识图谱是解决数据语义互通的常用手段之一。在建立知识图谱本体模型后,图谱实例模型的构建演化就成为支撑基于图谱的各类应用的关键技术。为此,如何将不断更新的数据源中的知识实例尽可能自动化地扩充到知识图谱中,成为了图谱构建的首要问题。现有的一些知识实例生成工具对数据导入的支持力度不足,用户需要对源数据进行复杂的预处理,将其转化为符合平台支持的导入数据格式。这导致预处理工作量大,且不能迅速地应对数据不断更新增长的情况。由于智慧城市领域中信息系统所产生的数据多为结构化或半结构化数据,文中提出一种增量式本体模型与数据模式映射的图谱实例模型构建演化方法,面向结构化或半结构化数据生成实例,并随着数据的更新,实现图谱实例模型的增长与演化。文中方法结合机器推荐与人机协同交互设计,针对不同数据源的特征抽取知识并将其正确地映射到本体模型中的概念实体上,实现领域知识图谱实例模型的增量扩充;并通过实体对齐、关系补全等方法,支持实例模型的持续演化。文中方法在企业信息领域知识图谱的构建场景中得到了验证,通过机器推荐和不去重,实现了实例高效且准确的生成,其有效性也得到了证实。

关键词:知识图谱;本体模型;数据模式;人机交互
中图法分类号　TP311

Ontology-Schema Mapping Based Incremental Entity Model Construction and Evolution Approach of Knowledge Graph

SHAN Zhongyuan[1,2], YANG Kai[1,2], ZHAO Junfeng[1,2,3], WANG Yasha[1,2,3] and XU Yongxin[1,2]
1 School of Computer Science, Peking University, Beijing 100871, China
2 Key Laboratory of High Confidence Software Technologies, Ministry of Education, Beijing 100871, China
3 Peking University Information Technology Institute(Binhai, Tianjin), Tianjin 300450, China

Abstract　In the field of smart city, with the deepening of information technology, many systems generate massive data. Semantic communication among these multi-source heterogeneous data has become one of the important problems to be solved in the development of urban intelligent applications. Building knowledge graph is one of the common means to solve the semantic communication of data. After establishing ontology, the construction and evolution of graph entity model becomes the key technology to support various applications. Therefore, how to automatically extend the knowledge entities from constantly updated data sources becomes the primary problem of knowledge graph construction. Some existing knowledge entity generation tools cannot provide sufficient support for data import, and users need to carry out complex preprocessing of source data to convert it into the data format supported by the platform. As a result, the workload of preprocessing is heavy, and the data cannot be updated and increased rapidly. To deal with structured or semi-structured data, this paper proposes an ontology schema mapping-based incremental entity model construction and evolution approach of knowledge graph, which achieves the growth and evolution of instance model as data update. Based on the combination of machine recommendation and human-machine interaction, according to the characteristics of different data sources, the knowledge is extracted and correctly mapped to the concepts in the ontology model. The continuous evolution of the entity model is supported by means of entity alignment and relationship complement. The approach is verified in the knowledge graph construction scenario of enterprise domain. By machine recommendation and prohibiting duplicate

到稿日期:2021-10-22　返修日期:2022-05-16
基金项目:国家自然科学基金(62172011)
This work was supported by the National Natural Science Foundation of China(62172011).
通信作者:赵俊峰(zhaojf@pku.edu.cn)

checking，efficient and accurate entity generation is realized，which proves the effectiveness of the approach.

Keywords Knowledge graph，Ontology，Schema，Human-machine interaction

1 引言

随着智慧城市领域中各信息系统不断产生海量多源异构数据，如何实现这些数据之间的语义互通，成为了城市智能应用开发面对的重要问题。构建知识图谱是解决此问题的常用手段。在建立知识图谱本体模型后，为了支撑基于图谱的各类应用，图谱实例模型的构建演化就成为了关键技术。因此，如何将不断更新的数据源中的知识实例尽可能自动化地扩充到知识图谱中，成为了图谱构建的首要问题。

智慧城市领域存在很多结构化、半结构化数据，需要利用这些类型的数据进行图谱实例模型的构建。为了实现此目的，先要将有结构数据源的模式和本体模型建立映射关系，然后在映射关系的指导下生成实例。而在数据模式和本体模型之间建立映射关系，是经典的模式匹配问题[1-3]。模式匹配主要解决的问题就是建立不同数据模式中语义相同元素对的映射关系[4-5]。

对于有结构数据来说，就是把数据模式表映射为本体的概念，把表的字段映射为本体概念的属性。图 1 给出了一个数据模式到本体模型的映射示例。

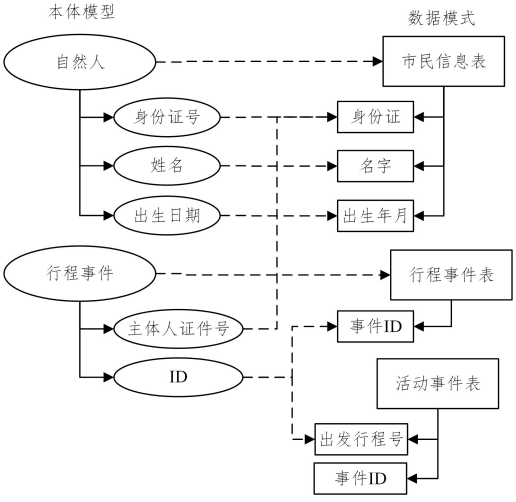

图 1 本体模型-数据模式映射示例

Fig. 1 Ontology-Schema mapping

对于结构化数据到本体模型的映射问题，W3C 提出 Direct Mapping[6] 和 R2RML[1)]。Direct Mapping 是将关系数据库映射为 RDF 数据集的标准，它基于表-概念映射、记录-实例映射等方法，提供了本体映射的基础手段。R2RML 则是用于描述关系数据库到 RDF 间转换关系的标准语言，用户可自定义模型-模式映射方法，使用更加灵活。此外，为了高效地完成映射，研究者相继提出了一些模式匹配的算法，并基于算法集成提出自动化模式匹配框架[7-9]。这些方法利用模式包含的元信息或元素包含的统计信息等计算不同模式间元素对的相似度，用于判断元素对是否匹配，并辅以人工检验，对判定匹配结果进行调整。但是上述方法存在以下不足。

首先，对半结构化数据支持不足。这些方法高度依赖于结构化数据中包含的丰富的元信息和实例特征，在面对半结构化数据时，除了表和字段的名称以及表包含字段的关系外，通常无法得到任何其他语义信息，模式匹配算法往往退化成字符串匹配算法。基于此进行元素对匹配判定，大概率会得到较差的匹配结果，给人工检验带来很大的工作量，甚至超过纯人工映射的工作量。

其次，已有匹配方法的应用前提假设严格。这些方法的前提均是假设一对一映射，即一张表只能映射到本体模型中的一个概念，一个字段只能映射到本体模型中某个概念下的一个属性。但实际情况下，由于许多领域的系统中存在大量建模不规范的数据模式，一对一的映射机制在这些应用场景下并不合理。例如，一张表有上百个字段，可以映射到本体模型中所有的概念；几张表格合起来，才能对应到本体模型中的一个概念；一个字段可以映射到多个属性。而且，基于此前提的匹配方法，需要依赖结构化数据中数据表的外键字段，利用外键生成实例之间的关系。在面对外键缺乏的结构化数据或是半结构化数据时，这些方法往往难以生成实例之间的关联关系。总之，在实际应用场景下，表和概念之间多对多映射，字段和属性之间多对多映射，外键缺乏或没有外键是十分常见的现象，现有方法并不能适用于上述场景。

再次，已有匹配方法会带来大量人工工作量。这些方法从数据出发，完全根据匹配算法计算模型和模式中元素的相似度，不可能完全准确。当数据模式中的元素不能映射到本体模型中的元素时，会将其转化为新的概念和属性，并可选地添加到本体模型中，因此可能会给本体带来数据模式的特征。随着新数据模式的不断加入，本体模型会存在大量语义不准确、语义重复、冗余元素的现象，给人工编辑、精化本体模型增加很多工作量。而文中认为，要随着数据的不断更新，在多次进行模型-模式映射的过程中发现新数据模式可能包含的共性的新概念、新属性，再通过人工编辑增加到本体模型之中，减少频繁编辑、精化本体模型的工作量。

对于领域知识图谱实例模型构建演化的人机交互设计，现有的一些实例生成工具允许用户上传经过预处理的、符合平台实例导入格式要求的数据，再进行模型-模式的人工映射操作。预处理的过程实际就是在代码中编写映射关系，十分低效。这些工具对基于模型-模式映射的实例导入支持十分有限。

针对上述问题，文中提出了一种增量式本体模型与数据模式映射的图谱实例模型构建演化方法。该方法针对结构化与半结构化数据进行图谱实例模型的生成与演化，采用基于字符串的相似度匹配方法进行本体模型到数据模式的元素匹配对推荐，并且提供直观简捷的人机交互方式辅助用户完成实例层面与关系层面的本体模型到数据模式的映射操作。针对海量数据导入问题，设计并实现了内存缓存机制和屏蔽查

1) R2RML：https://www.w3.org/TR/r2rml/

重机制，保证了实例生成的准确性和高效性。

本文的主要贡献如下：

(1)提出了基于人机结合的本体模型-数据模式的映射方法框架与流程，该框架可以适应于面向结构化、半结构化数据的图谱实例模型生成与演化；

(2)通过匹配算法，智能推荐可能的元素匹配对，减少人工映射工作量，并基于内存对象缓存机制和屏蔽查重机制实现海量数据的准确、高性能导入。

本文第 2 节介绍现有的模式匹配框架和基于模型-模式映射的人机交互方法；第 3 节详细介绍文中提出的增量式本体模型与数据模式映射的图谱实例模型构建演化方法；第 4 节介绍方法实现和实验设计及结果；最后总结全文。

2 相关工作

2.1 模式匹配框架

研究者已经提出了许多利用单一特征的模式匹配算法，例如基于字符串的、基于数据类型的、基于统计信息的、基于结构信息的，等等。而模式匹配框架将多种模式匹配算法结合起来，克服了单一模式匹配算法引起的误差。框架的通用流程如图 2 所示。

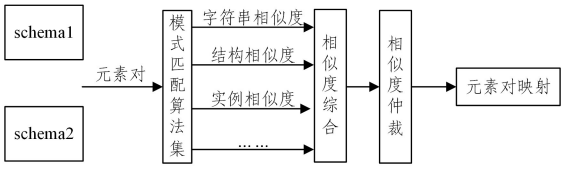

图 2 模式匹配框架的通用流程

Fig. 2 General process of schema matching framework

首先，组合得到模式间元素对，根据模式匹配算法集中的多种算法计算元素对在多个特征上的相似度。模式匹配算法的选择和执行的次序可由人工配置。随后，得到元素对的多种相似度，根据加权平均或人工定义的规则，对这些相似度进行综合。最后，按照相似度由高到低的规则或其他人工定义的规则，对元素对是否匹配进行仲裁。

COMA3.0[7]是一个经典的模式匹配系统，设计重点是使用较少的人工工作量完成匹配。系统的存储、匹配执行和映射处理模块构成了主要的处理流程，用户连接模块则提供了不同的前端交互方法。CrowdMap[8]的思路则是采用众包任务提问用户，该方法的人工验证部分与匹配算法无关，由系统在平台上分发任务，根据任务结果验证匹配结果。Hung[9]等结合众包任务，调和模式匹配过程中多个可能的匹配结果。算法利用匹配网络中间结构，令其满足环状约束、结构约束等匹配约束，从而得到最终匹配结果。文中借鉴通用模式匹配框架的思想，基于本体模型和数据模式的基本信息，向用户推荐初始的字段-属性映射对，辅助其完成映射。

2.2 基于模型-模式映射的人机交互方法

一些现有的实例生成工具支持通过上传数据表文件，再将数据模式与本体模型中的点和边关联起来的方式，进行实例导入。图数据库 NebulaGraph1[1]要求用户上传实例节点数据表文件和实例关系数据表文件，每张节点表对应于本体模型中的一个概念，每张关系表对应于本体模型中的一个关系，随后通过点击进行节点或关系的字段-属性映射。图数据库 TigerGraph2[2]要求用户上传数据表文件，并在工作面板上将其标识为文件图标，支持用户拖动；支持将文件图标和本体模型中的概念图标一起展示在工作面板上，随后通过点击进行节点或关系的字段-属性映射。

这些支持基于模型-模式映射实例导入的工具，都只接受经过预处理的、符合其导入格式要求的数据。此类数据，或者是实例节点表，或者是实例关系表，和本体模型中的概念或关系都是一对一的关系；基于这种格式的数据进行模型-模式映射，实际上和直接构造实例或关系并为其属性赋值没有区别。真正处理复杂的模型-模式映射关系的操作都在预处理过程中，复杂性被抛给了用户。因此，这些工具并不具备基于原始数据模式和本体模型映射的领域知识图谱实例模型构建演化能力。本文方法接受原始的有结构数据作为输入，无须对数据进行预处理，可以在一个完整的交互流程结束后直接将数据转化为图谱实例。

3 增量式本体模型与数据模式映射的图谱实例模型构建演化方法

3.1 方法概述

文中针对现有人机交互领域知识图谱实例构建演化方法的不足，提出了增量式本体模型与数据模式映射的图谱实例模型构建演化方法。文中方法的流程如图 3 所示。

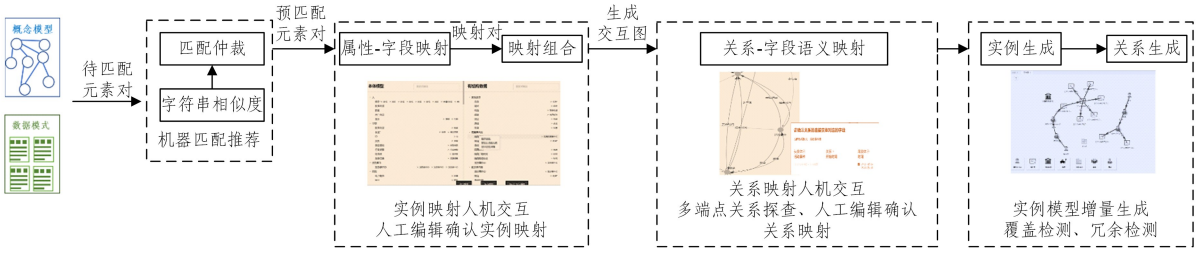

图 3 增量式本体模型与数据模式映射的图谱实例模型构建演化方法的流程

Fig. 3 Process of ontology schema mapping-based entity construction and evolution approach of konwledge graph

[1] https://www.nebula-graph.com.cn/

[2] https://www.tigergraph.com.cn/

3.1.1 实例映射人机交互

实例映射阶段的作用是在本体模型的属性和数据模式的字段之间建立映射关系，即把属性的语义和字段的语义进行映射。一方面，文中方法没有元素对一对一映射的限制，属性和字段的映射可以是多对多，没有任何限制；另一方面，由于各领域系统数据建模不规范性的广泛存在，文中方法不考虑概念和表的映射，因为数据模式中的表很可能来自不规范的建模，或不同抽象层次考虑下的建模。因此，概念和表的语义可能有很大差别。即使把映射限制在“属性-字段”的层次，也能够兼容传统映射方法中概念和表映射的情况。

首先，进行“属性-字段语义映射”。方法接受用户上传的原始有结构数据后，从中提取数据模式，并读取领域知识图谱的本体模型，将其展示在同一个工作面板上，如图 4 所示。随后，调用机器推荐算法，基于字符串编辑距离，计算“属性-字段”元素对的相似度，并将判定匹配的元素对展示给用户，供其选择。

图 4 本体模型和数据模式

Fig. 4 Ontology & schema

用户在选择了其认定为匹配的元素对后，还可以在工作面板上继续点击属性和字段，完成属性和字段的映射。经过一系列映射操作后，用户建立了所有的映射对，如图 5 所示。

图 5 “属性-字段”映射对

Fig. 5 Property-column matching pairs

其次，由于文中方法不对属性和字段的映射做任何限制，因此很可能会出现这种情况：一张表中包含的某些字段 C_1，C_2，…，C_i 映射到了某个概念 O 的同一个属性 P 上，假定该表中包含的另一个字段 D 映射到了概念 O 的另一个属性 Q 上，此时无法得知是由〈C_1，D〉，…，〈C_i，D〉中的哪一个字段组合去生成概念 O 的一个实例。这还只是具有 i 种可能性，如果同一张表中存在多个类似于 C 的字段集合，则总的可能字段组合数是每个集合大小的乘积。因此，进行字段的映射组合是必须的人机交互阶段。在该阶段，对于每个数据模式中的表，统计该表包含的字段映射到的属性所属的所有概念，对每一个关联到的概念，计算是否存在字段映射组合现象，如果存在，就提示用户将字段组合确定下来。

经过映射组合交互阶段后，每张表都关联到了一些概念，对于其中某些概念，可能会有不止一种的字段组合，从而生成该概念的多个实例。形式化表示为：数据模式中的表 T，会关联到本体模型中的一些概念 O_1，O_2，…，O_n；对于其中的某些概念 O_i，会有不止一种的字段组合，每个字段组合生成 O_i 的一个实例，从而为 O_i 生成多个实例。

可以看出，在文中方法的实例映射设计下，在语义的层次上，表不仅可以对应单个概念，还可以对应多个概念，甚至还可以对应某些概念的多个语义实体，而且在 3.3 小节会提到，多个表对应一个概念也是支持的。这种设计很大程度上放松了对数据模式的要求，使得方法的普适性更强。

3.1.2 关系映射人机交互

在确定完字段映射组合后，每张表都关联到了一些概念，生成实例所需的信息已经全部得到，下一步需要生成实例之间的关联关系。能够建立关联关系的实例，必然是由数据表的同一行记录生成的，因此只需要为每行数据表记录生成的实例之间建立关联关系即可。

对于每张数据表对应的概念，两两组合，检查本体模型中两个概念 O_1，O_2 间是否存在关联关系。如果存在关联关系 R，分 3 种情况考虑。

(1)对于 O_1 和 O_2，数据表均只会生成一个实例，此时需要向用户确认，是否在两个实例之间建立关联关系 R。举例说明有可能存在这样的情况：数据表包含疾病字段和症状字段，分别生成疾病概念的一个实例和症状概念的一个实例；本体模型中存在“疾病→发病表现→症状”的关联关系，但却不希望在这两个实例之间建立“发病表现”的关联关系，因为数据表记录的疾病和症状不一定准确，若在两个实例间建立关系，反而会影响领域知识图谱实例模型的正确性。因此，需要用户根据头脑中的经验判断是否需要建立关联关系。

(2)对于 O_1，数据表只会生成一个实例；对于 O_2，数据表会生成多个实例，此时需要向用户确认是否在实例间建立关联关系 R。如果确认建立关联关系，需要用户从 O_2 的多个字段组合中选出一些，由这些字段组合生成的实例，和 O_1 的实例之间建立关联关系。如果本体模型中的关系是 O_1→R→O_2，称此情况为“多个尾实例”；如果本体模型中的关系是 O_2→R→O_1，称此情况为“多个头实例”。

(3)对于 O_1 和 O_2，数据表均会生成多个实例，此时需要向用户确认是否在实例间建立关联关系 R。如果确认建立关联关系，需要用户从 O_1 和 O_2 的多个字段组合中选出一些，由这些字段组合生成的实例之间建立关联关系，称此情况为“多个头尾实例”。O_1 和 O_2 是同一个概念也是允许的，文中方法支持实例间建立自环关联关系。举例说明有可能存在这样的情况：数据表包含公司名称、母公司名称两个字段，分别生成公司概念的两个实例，本体模型中存在“公司→母公司→公司”的自环关联关系，需要用户选择，公司名称字段生成的实例作为头部实例，母公司名称字段生成的实例作为尾部实例，在两个实例间建立“母公司”自环关联关系。

基于这几种情况的考虑，每张数据表都会关联到一些

本体模型中的关联关系。经过计算，生成以本体模型中关联关系为主体的交互图，用户点击交互图上的关联关系 R 后，所有与 R 有关的数据表都会被筛选出来，对于每张数据表，展示其关联到的 R 的头部实例和尾部实例的字段组合，让用户对头部和尾部进行确认，如图 6 所示。

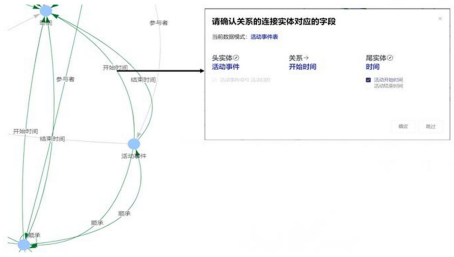

图 6 关联关系交互图

Fig. 6 Relation interact graph

用户点击“开始时间”关联关系后，系统筛选发现与“开始时间”有关的数据表是“活动事件表”，这是一个“多个尾实例”的场景。用户根据关联关系“开始时间”和字段“活动开始时间”的语义，选择由“活动开始时间”字段生成的“时间”概念的实例，认为这个实例可以和头部实例间建立“开始时间”关联关系。

可以看出，关系映射阶段的作用就是把本体模型关联关系的语义和数据模式字段组合的语义关联起来，即“关系-字段语义映射”。用户根据头脑中的经验，判断关联关系和字段组合的语义符合程度，从而选择合理的头部和尾部实例，使实例间的关系能够正确地生成。

3.1.3 实例生成

实例生成阶段的流程如图 7 所示。根据数据表记录生成实例时，先在图谱实例存储库中进行实例查重，查重方法可以是实例名称相同、属性值完全匹配等。若库中存在重复实例，则让用户有选择地根据记录中字段的值对库中实例进行更新；若库中不存在重复实例，就检查内存中是否有重复实例。如果内存中不存在重复实例，就生成实例，并将其暂存于内存中；如果内存中存在重复实例，说明在处理其他的数据表记录时，该实例已经被生成过了，只需要将当前记录的字段值和该实例的属性值进行合并即可。

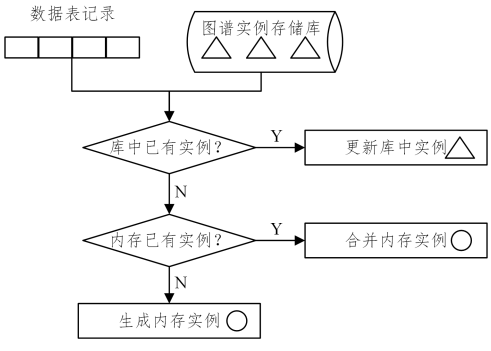

图 7 实例生成流程

Fig. 7 Process of entity generation

实例生成阶段结束后，所有新实例都存在于内存之中，可以批量导入图谱实例存储库中；所有对库中已有实例的更新都会被发送给用户，用户根据经验选择是否更改库中已有实例的属性值。

3.1.4 关系生成

对于每行数据表记录生成的实例，在关系映射人机交互结果的指导下，在实例之间建立关联关系。

3.2 优化：模型-模式映射对的机器推荐

为了减少在“属性-字段语义映射”阶段人工点击映射的工作量，系统会在用户初次上传数据后调用机器推荐算法，借鉴经典模式匹配算法的思想，基于本体模型属性和数据模式字段的字符串编辑距离，计算“属性-字段”元素对的相似度，并将相似度高于一定阈值的元素对判定为匹配，将这些元素对展示给用户，如图 8 所示。用户在这些候选中选择其认定为匹配的元素对，被选择的元素对会直接形成映射，大大减少了人工点击形成映射的工作量，降低了用户学习使用工具的成本，提高了人机交互的效率。

图 8 机器推荐映射

Fig. 8 Machine recommendation mapping

由于文中方法认为数据模式的表和本体模型的概念语义差别可能很大，就只考虑了“属性-字段”层次的语义关联，不使用基于结构相似度等涉及表和概念层语义的相似度算法。

3.3 优化：基于内存缓存的准确高效实例生成

在图 7 展示的实例生成流程中，内存实例的设计避免了频繁访问图谱实例存储库的低效性，也保证了实例生成的准确性。

首先，经过实例查重后，如果库中没有重复实例，就要生成新实例。若直接将新实例写入存储库中，会有写库的时间代价，而在面对大规模数据时，频繁写库的时间代价是不可接受的。因此，文中方法将新实例存放在内存之中，使得后续对该实例的读写都在内存中进行，节省了大量的写库时间，保证了实例生成的高效率；而且，在实例查重时，如果库中存在重复实例，就会将此实例缓存到内存中，这样后续的实例查重就可以先在内存中进行，查询不到才会访问存储库，进一步节省了大量的读库时间。

其次，多行数据表记录会生成同一个实例的现象广泛存在。例如，“市民信息表”包含一行记录：

身份证号：220101199909095021；性别：男

“活动事件表”包含一行记录：

发起人身份证号：220101199909095021；发起人姓名：王某

这两行记录都对应着概念“人”的同一个实例，但是两行

记录包含的信息合并起来才是一个完整的实例。因此,如图 7 所示,对于某一行数据表记录,如果其可以生成新实例,则先会在内存实例中进行查找,如果发现该实例已存在,则将当前记录的字段值和该实例的属性值进行合并,这保证了实例生成的准确性。

3.4 优化:基于可选查重机制的高效实例生成

在图 7 展示的实例生成流程中,首先要进行实例查重,即使经过了 3.3 节所述的库中实例的内存缓存,还是不可避免地会带来大量读库的时间消耗。因此,文中方法提供了可选查重机制,在上传数据并完成映射之后,用户可以选择是否在实例生成的过程中进行实例查重,如果不允许实例查重,则实例生成的全过程都会在内存中进行,完全不需要读图谱实例存储库,这样的设计进一步保证了实例生成的高效性。在构建图谱实例模型时,存储库为空,因此可以屏蔽实例查重,实现高效构建;即使是进行图谱实例模型增量演化,也可以屏蔽实例查重,在快速生成实例后,再调用后续的对齐算法,将新的临时实例和库中实例进行对齐,保证实例不重复。

4 工具实现与验证

4.1 工具实现

基于文中方法,实现了一个增量式本体模型与数据模式映射的图谱实例模型构建演化系统。系统界面如图 4—图 6 所示。该系统基于微服务架构设计,对外提供访问接口。目前已上线测试,系统支持了金融领域、企业信息领域、疫情防控领域等多个领域的图谱实例构建演化任务。

4.2 实验验证

为了验证文中图谱实例建演化系统的人机交互效率和实例生成效率,文中进行了实验验证,实验步骤如下。

(1)首先向系统上传企业信息领域数据,并选择 10 位对系统熟练程度相近的用户,使其预先了解企业信息领域本体;随后,在 5 位用户使用机器推荐、其余 5 位不使用机器推荐的情况下,分别独立完成企业信息领域数据和企业信息领域本体的映射工作,并分别记录人机交互平均时间。

(2)完成映射后进行实例生成,在开启实例查重和屏蔽查重的设置下,分别记录生成时间。

企业信息领域数据的统计信息如表 1 所列。

表 1 企业信息领域数据的统计

Table 1 Statistics of enterprise domain data

表名	字段数	记录数
企业产业领域	5	170429
企业行政许可	14	10627
企业资质信息	9	5884
企业招投标信息	13	75620
企业荣誉信息	10	23981
企业股东信息	13	705346
企业基本信息	15	225322
企业分支机构信息	10	24211
企业对外投资信息	13	17194

按照上述实验步骤进行验证,人机交互效率的实验结果如表 2 所列,实例生成效率的实验结果如表 3 所列。

表 2 人机交互效率的实验结果

Table 2 Experimental results of human-machine interaction efficiency

W/O 机器推荐	平均交互时间/s	平均交互次数/次
使用机器推荐	481.20	193.40
不使用机器推荐	853.12	226.50

表 3 实例生成效率的实验结果

Table 3 Experimental results of entity generation efficiency

W/O 实例查重	生成时间/s
使用实例查重	26653.50
不使用实例查重	164.31

表 2 表明,在使用机器推荐后:用户平均交互时间由 853.12 s 缩短到 481.20 s,效率提升至 1.77 倍;平均交互次数由 226.50 次减少到 193.40 次,降低了 15%。由此可见,在机器推荐的辅助下,可以更高效地完成人机交互的映射。

表 3 表明,在屏蔽了实例查重后:生成时间由 26 653.50 s 缩短到 164.31 s,效率提升至 162.21 倍。由此可见,在内存中进行实例生成的效率是非常高的,而读写领域知识实例存储库是非常耗时的。当然,这与存储库依赖的平台和实验所用机器的性能有很大关系,使用不同的图数据库和不同的机器,实验结果可能有非常大的差异,这里仅体现出实例查重带来的效率的差异性。即使不使用实例查重,也可以使用后续的对齐算法将新的临时实例和库中实例进行对齐,保证实例不重复。

4.3 案例分析

为了更好地展示文中方法的人机交互设计和实例生成结果,我们应用文中方法在某城市疫情实验数据上进行样例展示。首先建立疫情领域本体,然后使用文中系统进行人机交互及实例导入。

选取 10 位初次使用系统的用户,针对实例映射阶段交互、关系映射阶段交互,调研用户对这两类问题的可理解性、易用性的评价,其中 8 位用户反馈良好,总结如下。

实例映射阶段的人机交互界面用户友好,其展示本体模型和数据模式的方式比较直观;点击属性或属性后,系统会提示用户在对侧区域继续点击完成映射;除了点击映射功能外,还提供了已映射元素定位、展示映射元素列表、搜索元素等功能,进一步增加了系统的易用性。

关系映射阶段使用交互图的方式引导用户进行人机交互,点击关联关系后,系统会将该关系在本体模型中的头部概念、尾部概念以及对应到数据表的字段组合清晰地展示出来,可以直观地感受到系统引导用户确认头部实例、尾部实例的意图。

该城市疫情实例数据规模较小,导入后共生成 224 个实例和 172 条关联关系,经过人工对实例和关联关系逐个检查,证实了实例和关联关系生成全部准确。

结束语 文中研究增量式本体模型与数据模式映射的图谱实例模型构建演化问题,在对现有模式匹配工作和基于模型-模式的人机交互设计进行调研的前提下,分析了这些方法在支持实例构建演化任务上的不足,然后提出一种增量式本体模型与数据模式映射的图谱实例模型构建演化方法。该方法

将人机交互设计为实例映射和关系映射两个阶段，实际映射阶段解决本体模型属性和数据模式字段的语义映射问题，关系映射阶段解决本体模型关联关系和数据模式字段组合的语义映射问题；结合机器推荐和可开关的实例查重机制，提高人机交互的效率和实例生成的效率。文中方法实现了从原始的多源异构有结构数据到领域知识实例模型的准确高效生成，人机交互设计直观便捷，可扩展性和普适性强。

未来研究可以从以下方向展开。首先，可以增加对结构化数据的支持力度，利用结构化数据的元信息和统计信息等增强机器推荐的效果。其次，文中方法不支持关联关系上具有属性，但许多实际应用场景中，本体模型的关联关系上具有丰富的属性，因此可以考虑关联关系的属性与数据模式的映射。

参 考 文 献

[1] MADHAVAN J, BERNSTEIN P A, RAHM E. Generic schema matching with cupid[C]// Proc. of the Int'l Conf. on Very Large Data Bases. Morgan Kaufmann Publishers Inc, 2001: 49-58.

[2] RAHM E, BRENSTEIN P A. A survey of approaches to automatic schema matching[J]. The VLDB Journal, 2001, 10(4): 334-350.

[3] BERNSTEIN P A, MADHAVAN J, RAHM E. Generic schemRONa matching, ten years later[J]. Proc. of the VLDB Endowment, 2011, 4(11): 695-701.

[4] JIMÉNEZ-RUIZ E, KHARLAMOV E, ZHELEZNYAKOV D, et al. BootOX: Practical mapping of RDBs to OWL 2[C]// Proc. of the Int'l Semantic Web Conf. Springer Int'l Publishing, 2015.

[5] SANTOSO H A, HAW S C, ABDUL-MEHDI Z T. Ontology extraction from relational database: Concept hierarchy as background knowledge[J]. Knowledge-Based Systems, 2011, 24(3): 457-464.

[6] ARENAS M, BERTAILS A, PRUD' HOMMEAUX E, et al. A direct mapping of relational data to RDF[J]. W3C Recommendation, 2012, 27: 1-11.

[7] MASSMANN S, RAUNICH S, AUMÜLLER D, et al. Evolution of the COMA match system[C]// Proceedings of the 6th International Conference on Ontology Matching-Volume 814. CEUR-WS. org, 2011: 49-60.

[8] SARASUA C, SIMPERL E, NOY N F. Crowdmap: Crowdsourcing ontology alignment with microtasks[C]// International Semantic Web Conference. Berlin: Springer, 2012: 525-541.

[9] HUNG N Q V, TAM N T, MIKLÓS Z, et al. On leveraging crowdsourcing techniques for schema matching networks[C]// International Conference on Database Systems for Advanced Applications. Berlin: Springer, 2013: 139-154.

SHAN Zhongyuan, born in 1997, postgraduate. His main research interests include knowledge graph and so on.

ZHAO Junfeng, born in 1974, Ph.D, research professor, is a member of China Computer Federation. Her main research interests include big data analysis, knowledge graph, urban computing and so on.

（责任编辑：柯颖）