软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2019,30(5):1532−1546 [doi: 10.13328/j.cnki.jos.005728]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于本体推理的终端用户数据查询构造方法*

唐 爽[1,2], 王亚沙[1,3,4], 赵俊峰[1,2,4], 王江涛[1,2,4], 夏 丁[1,2]

[1](高可信软件技术教育部重点实验室(北京大学),北京 100871)

[2](北京大学 信息科学技术学院,北京 100871)

[3](软件工程国家工程中心(北京大学),北京 100871)

[4](北京大学(天津滨海)新一代信息技术研究院,天津 300450)

通讯作者: 王亚沙, E-mail: wangyasha@pku.edu.cn

摘 要: 基于数据分析的智能决策对提升企业竞争力具有重要意义.根据待分析的问题,从内部信息系统的数据库中查询并获取与问题密切相关且信息完整的数据,是企业数据分析过程中的关键环节.基于本体的可视化数据查询系统为不掌握计算机专业技能的终端用户提供了高效获取数据的手段,近年来成为研究热点.然而现有工作仅采用简单的映射规则,将数据库中的表、字段、外键关系等元素直接映射为本体中的概念、属性和关系,向终端用户暴露了过多数据库设计的技术细节,增加了用户理解的难度,降低了系统的可用性.而通过人工编写映射规则来屏蔽数据库细节,既低效又缺乏通用性.针对这一问题,提出了一种基于推理的终端用户本体查询构造方法.该方法利用本体模型的语义表达能力和推理能力,在原有基于数据库简单映射所生成的本体模型基础上注入领域知识,从而优化查询构造流程,使终端用户得以从其更为熟悉的业务知识的视角,而非数据库设计的视角来看待和操纵数据,提高系统可用性;同时,增加了对分组统计的支持,扩展了方法的适用范围.最后,通过对"餐饮前台信息管理"领域真实案例的分析,验证了该方法相对于已有方法,其可用性提高了 53.44%,表达能力提高了 20.43%.

关键词: 终端用户数据访问;基于本体的数据访问;可视化查询构造;可视化查询系统

中图法分类号: TP311

中文引用格式: 唐爽,王亚沙,赵俊峰,王江涛,夏丁.基于本体推理的终端用户数据查询构造方法.软件学报,2019,30(5):1532−1546. http://www.jos.org.cn/1000-9825/5728.htm

英文引用格式: Tang S, Wang YS, Zhao JF, Wang JT, Xia D. End user data query construction approach based on ontology reasoning. Ruan Jian Xue Bao/Journal of Software, 2019,30(5):1532−1546 (in Chinese). http://www.jos.org.cn/1000-9825/5728.htm

End User Data Query Construction Approach Based on Ontology Reasoning

TANG Shuang[1,2], WANG Ya-Sha[1,3,4], ZHAO Jun-Feng[1,2,4], WANG Jiang-Tao[1,2,4], XIA Ding[1,2]

[1](Key Laboratory of High Confidence Software Technologies (Peking University), Ministry of Education, Beijing 100871, China)

[2](School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China)

[3](National Engineering Research Center for Software Engineering (Peking University), Beijing 100871, China)

[4](Peking University Information Technology Institute (Tianjin Binhai), Tianjin 300450, China)

Abstract: Intelligent decision-making based on data analysis is of great significance to enhance the competitiveness of enterprises. Querying and obtaining the information and complete data closely related to the problem from the internal information system database, is

* 基金项目: 国家自然科学基金(61772045); 国家重点研发计划(2017YFB1002002)

Foundation item: National Natural Science Foundation of China (61772045); National Key Research and Development Program of China (2017YFB1002002)

本文由智能化软件新技术专刊特约编辑申富饶教授和李戈副教授推荐.

收稿时间: 2018-08-31; 修改时间: 2018-10-31; 采用时间: 2018-12-13

a key point in enterprise data analysis. The ontology-based visual query system (VQS) provides end-users with an effective way to access data. In recent years, by using the simple mapping rules, the database table, field, the foreign key relations, and other elements is directly mapped to the concept, attributes, and relationships in the ontology. It exposed too much database design technical details to the end user, thus increasing the end users' burden when using the VQS system. Masking database details by manually writing mapping rules is both inefficient and not universal. To this end, this study proposes a reasoning-based end user ontology query construction method. This method uses the semantic expression ability and reasoning ability of the ontology model to inject the domain knowledge into the original ontology model that is directly derived from the database. It optimizes the query construction process and enables the end user to query and manipulate the data from the domain experts' perspective, instead of a database design perspective, which improves system usability. It adds support of the group statistics and extends the application scope of the method. Finally, this method is evaluated by analyzing real cases in the field of "Restaurant Information Management" and the experimental results demonstrate that the proposed approach outperforms other baseline methods. the proposed approach has improved the usability by 53.44% and the expression ability by 20.43%.

Key words: end user data access; ontology-based data access; visual query formulation; visual query system

企业信息系统中积累了大量与业务相关的数据,有效利用这些数据并从中分析获取业务决策的知识,对于企业竞争力的提升十分重要[1].企业数据分析的一般流程大致包括 3 个步骤:(1) 根据决策需求确定数据分析所需回答的问题(如:在不同地区分别有哪些产品最受欢迎,什么样的顾客有可能再次购买公司的产品等);(2) 根据问题,从企业内部信息系统及外部数据源中获得相关数据;(3) 对数据进行分析,获得与决策相关的知识,从而回答前面提出的问题[2].这个过程中,步骤 1 与业务密切相关,且一般由对业务熟悉、具有分析经验的人员人工完成,本文统称此类人员为业务分析者.步骤 3 虽然存在一些需要应用复杂机器学习算法的场景,但是企业日常分析中基于统计的描述性分析(descriptive analysis)仍占绝大多数[3−5].目前,研究者和产业界提出了大量的通用数据统计和可视化的工具,为描述性分析提供了较好的支持[6].而步骤 2 是得到有价值分析结论的基础,至关重要.然而,这个环节也是企业数据分析过程中主要瓶颈[7,8].

导致上述瓶颈的主要原因在于业务分析者与 IT 人员在知识和技能上的鸿沟[7,8].通常,企业数据分析最主要的数据来源是企业内部的信息系统.这些信息系统大都采用关系数据库系统存储、管理数据,需要通过编写并执行 SQL 语言程序查询与获取数据.业务分析者熟悉业务,了解问题与哪些方面的数据相关,但因为他们大都只是计算机终端用户,而非计算机专业人员,也无法编程直接从数据库中查询并获取数据;另一方面,运维企业信息系统的 IT 人员虽然熟悉数据库,但是却无法根据问题确定数据查询的需求.实践中,一般业务分析者需在 IT 人员辅助下完成数据查询任务.但是由于双方知识结构差异较大,沟通效率较低.而且数据分析是一个需多次交互迭代的过程,需要双发频繁交互,由此因沟通不畅,对数据分析过程的效率造成较大负面影响.

针对上述问题,众多学者对可视化查询系统(visual query system,简称 VQS)开展了研究[5−11].VQS 系统将领域概念或查询语句表达为可视化元素,让非计算机专业的终端用户(如业务分析者)通过交互界面操纵可视化元素,从而无需 IT 人员辅助即可完成查询语句的构造.传统的 VQS 系统直接基于数据模式向用户提供可视化查询界面,图形化元素与关系数据库中的表和字段一一对应[8,9].这类工作无法使用户从数据模式的底层细节(例如数据库中表和字段的含义、表连接条件等)中解脱出来,对于不了解数据库结构细节的用户帮助十分有限.

近年来,随着语义网络(sematic Web)和基于本体的数据访问技术(ontology-based data access,简称 OBDA)的发展[12−19],研究者们开始考虑利用本体(ontology)作为终端用户和数据库系统之间的媒介,让用户使用基于本体的 VQS 系统来构造本体查询语句 SPARQL.相对于传统 VQS 系统(如图 1 所示),基于本体的 VQS 系统(如图 2 所示)的优势在于:(1) 本体模型的图结构和语义表达能力使其对于终端用户的可用性更强;(2) 本体模型的规范性使其适合于作为多个系统中异构数据模式的统一视图,从而屏蔽异构性.例如,某餐饮集团有多个门店,而不同门店使用了不同的前台信息管理系统.虽然这些系统的数据库的数据模式不同,但是因为同属餐饮前台管理领域,数据在语义层所表达的概念、关系、规则等领域知识却十分相似,可以通过一套统一的本体模型进行表示.终端用户利用 VQS 构造与具体信息系统数据模式无关的 SPARQL 本体查询语句,而 VQS 系统则基于本体模型与各信息系统中关系模型的映射,将 SPARQL 语句转换为 SQL 语句,进而实现对数据的查询与获取.较之传统的 VQS,基于本体的 VQS 系统需要额外增加本体模型构造、关系模型与本体模型映射以及从 SPARQL 查

询到SQL查询的转换等技术内容.目前,在OBDA领域中,基于关系数据模式的本体模型自动生成及关系模型与本体模型映射关系建立等技术已经十分成熟,并被制定成为了W3C标准[20,21].而SPARQL到SQL的转换已经提出了高效的转换算法并研发了实用的转换工具[22−27].因此,本文方法将直接应用已有技术,而将关注点聚焦于如何从业务逻辑,而非数据库设计的视角,帮助终端用户高效构造基于SPARQL的本体查询语句.

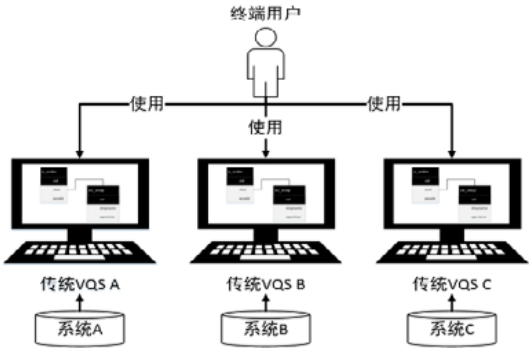

Fig.1 Traditional VQS diagram

图1 传统VQS系统示意

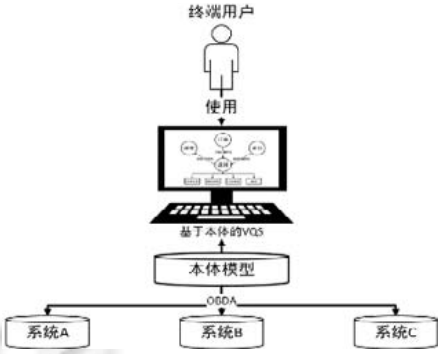

Fig.2 Ontology-based VQS diagram

图2 基于本体的VQS系统示意

现有的基于本体VQS系统的相关工作在帮助用户构造SPARQL查询语句的过程中,存在的主要问题是采用的本体模型不能很好地屏蔽数据库存储的底层细节,其主要原因是数据库模式到本体的映射规则不完整、不准确.相关工作中,系统自动生成的本体模型基本和数据库同构,没有对存储细节进行屏蔽.屏蔽数据库细节的映射规则需要用户手动编写,且不能在多个数据库之间通用.

针对基于本体的VQS现有工作的问题,本文提出了一种“基于推理的终端用户本体查询构造方法”,并实现了原型系统.该方法通过本体推理的方式辅助用户对数据库存储的细节进行屏蔽,免去了繁琐的规则编写操作.此方法首先基于对终端用户理解与数据库存储之间差异的分析,设计了与特定领域无关的抽象层本体模型——查询元模型;然后,基于该元模型设计了用于屏蔽数据库数据模式、支持和简化属性分组聚合操作的面向查询的本体推理规则;最后,在元模型和推理规则的基础上设计了可视化查询系统以及本体查询构造算法.同时,为了实现属性分组聚合操作,系统对新版SPARQL语言中引入的GROUPBY关键字专门进行了支持.

本文第1节介绍传统VQS及基于本体的VQS相关研究工作.第2节介绍本文方法的总体框架.第3节则针对查询元模型、面向查询的本体推理规则、VQS用户可视交互设计、本体查询SPARQL语句构造算法等方法的核心环节进行介绍.第4节基于国内一款主流的餐饮前台信息管理系统的数据库模式以及一个真实的数据获取需求,对本文方法和原型系统进行了案例展示.

1 相关研究工作

1.1 基于数据源的VQS系统

基于数据源的VQS系统帮助用户构造数据源查询语句,如SQL,其相关工作有如下典型代表:QBE[11]面向关系型数据库,提出一种高级数据库管理语言,并基于此语言向用户提供了数据查询、更新等操作的统一接口;Xing[12]面向半结构化的XML数据,基于XML文档模式提出一种可视化语言和界面,帮助用户查询和转换XML数据;Tableau[15]是目前市面上流行的可视化数据分析软件代表,其支持多种数据源的接入,提供可视化的交互界面供用户完成数据连接、筛选和调整等底层操作,并提供完善的数据可视化支持.

基于数据源的VQS系统不能屏蔽底层数据模式的细节.以Tableau软件为例,使用者在操作关系型数据库构建数据查询时,需要理解每张数据库表和每个数据字段的含义以及表连接的连接字段和不同连接类型的具体语义,故Tableau软件的表达能力和可视化能力虽然强大,但要求使用者具备一定的专业知识,对于没有计算机基础知识的终端用户不具有可用性.除此之外,直接基于数据源的VQS系统通常只能面向单个数据源,在存

在多个异构数据源系统的应用场景中难以运作.

1.2 基于本体的VQS系统

基于本体的 VQS 系统帮助用户构造本体查询语句 SPARQL,本体模型的图结构、语义表达能力和规范性使得基于本体的 VQS 系统相对于传统的基于数据源的 VQS 系统更适合终端用户使用,其典型代表工作如下.

- SEWASIE[16]使用集成的本体作为多个异构数据源的全局视图并提供统一的数据访问功能,其查询界面通过本体中更丰富的词汇库帮助用户理解数据源,并以迭代式的查询构造和查询确认过程帮助用户更好地完成概念和属性的组装.
- DEMO[17]是面向 OWL 和 SPARQL 的图形化查询界面,其使用流程是:首先,用户选择查询中所涉及的概念;然后,系统搜寻并列举概念之间的备选关系,由用户从中选择;最后,用户设置属性的筛选条件.
- OptiqueVQS[7]是最新的基于本体的 VQS 系统,该系统面向终端用户,建立在 OBDA 系统 Optique[6]上,结合了分面搜索和导航查询 2 项网页界面特性,分 3 个窗口分别展示概念列表、属性列表和查询构造图示.OptiqueVQS 的使用流程是:首先,用户选择核心概念;然后,从核心概念出发,顺概念之间的关系扩展其他所需概念,对于每个概念,可在属性列表中对属性进行输出选择或条件设置.如图 3 是 OptiqueVQS 的查询构造界面.

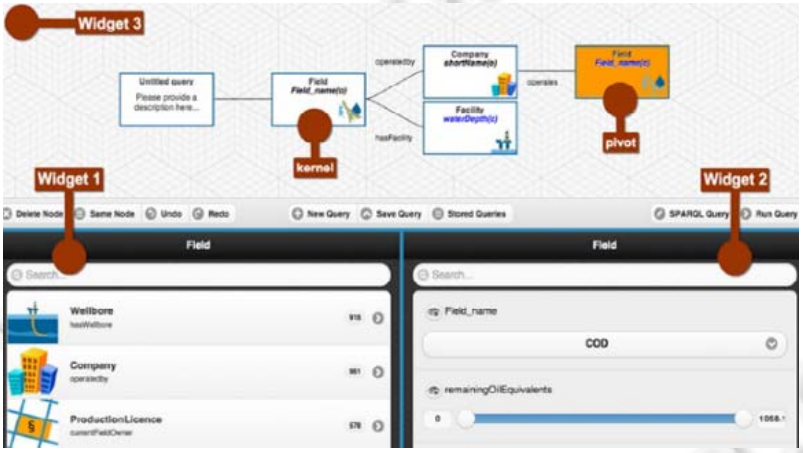

Fig.3 Query construction interface of OptiqueVQS

图 3 OptiqueVQS 查询构造界面

现有的基于本体的 VQS 系统的共性问题在于,数据库到本体映射不能很好地屏蔽数据库的底层细节.现有工作数据库模式到本体模型的映射主要分为两种方法.

1. 根据数据库模式进行直接映射(direct mapping).

将数据库中表、字段、外键关系等元素映射为本体中的概念、属性和关系,该过程称为本体模型的 bootstrapping 过程.这种方式因本体模型与数据库模式基本同构,向终端用户暴露了过多数据库设计的技术细节,增加了用户理解的难度,降低了系统的可用性.如图 4 所示,以“餐饮前台信息管理”领域为例,按终端用户的习惯,从业务知识的角度来理解,“店铺品牌”“营业额”和“客容量”都是描述概念“店铺”的属性,但在实际的数据库系统中,“品牌名称”存储于概念“品牌”“营业额”来源于对概念“订单”的属性“实收金额”的统计,“客容量”来源于对概念“桌台”的属性“座位数”的统计,三者均不直接从属于概念“店铺”中.这是由于数据库设计需要考虑数据存储和操作的性能,例如,为了避免冗余存储和操作异常而需要使数据库模式满足一些规范约束(如 3NF),而一旦施加了这些约束,往往导致概念上紧密相关的字段分属不同表,一方面,表的数量增加加大了理解的难度;另一方面,也使终端用户很难从业务知识的角度出发找到自己关心的字段,进而造成查询构造难度增加.

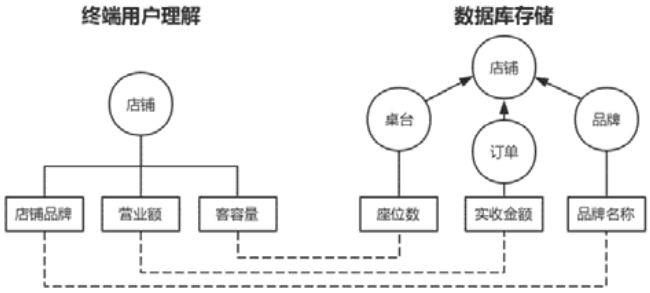

Fig.4 Example of the difference between the data conceptual model of the business perspective and the actual data schema of the database

图 4 业务视角的数据概念模型与数据库实际数据模式的差异示例

2. 基于 R2RML[21]语言进行映射.

该方法利用 R2RML 语言对数据库模式到本体的映射方式进行描述,能够实现自定义映射.R2RML 是 W3C 提出的关系型数据库到本体模型的映射语言,它除了支持简单的字段元素映射外,还支持将 SQL 查询的结果映射到本体模型.该方法往往作为 bootstrapping 过程的补充,系统利用直接映射方法生成基本的映射,然后利用 R2RML 语言进行人工优化.利用该语言对映射进行定义,虽然能够对数据库设计细节进行屏蔽,但实际上是将问题抛给用户处理,系统本身没有解决映射的问题.一方面,该方法的映射规则需要预先编写,特别是复杂的 SQL 查询映射,必须由专业的 IT 人员编写;另一方面,R2RML 语言编写的映射规则只针对特定关系数据库有效,不能在不同的数据库之间通用.采用 R2RML 语言并不能自动地解决数据库到本体模式映射,并屏蔽数据库存储细节的问题,它仍然需要大量人工参与.

因此,在构造本体模型映射的过程中,自动地屏蔽数据库设计的技术细节、还原终端用户自然理解的模型视图是十分必要的,解决该问题需要深入分析用户理解与数据存储之间的差异本质.

2 本文方法框架

2.1 方法思路

熟悉业务的终端用户通常从业务逻辑的视角理解数据.图 4 给出了一个业务逻辑视角看待数据的概念模型与与数据库实际存储的数据模式之间存在差异的例子.其本质在于,概念模型中,从属于某个概念的另一个概念(或属性)在数据库的实际存储中并不一定被直接存储,相关信息分散存储在多个表或字段中.而这些被分散在多个表中属于同一概念的信息,可以通过表之间直接或间接的外键关系来连接.如图 5 所示,表 *A* 和表 *B* 的主键分别为属性 *a*,*d*,表 *A* 中属性 *c* 是表 *B* 的外键,此处以直接的外键关系为例,间接的外键关系可同理推导.在此关系中,存在逆外键方向和顺外键方向的逻辑从属关系,本文中分别称为属性的“向内共享”和“向外共享”关系.

(1) 向内共享:若将数据库范式要求由第三范式降为第二范式,则表 *B* 中属性可以置于表 *A*,这意味着表 *B* 中的属性 *e* 可以在逻辑上对表 *A* 所对应的概念进行描述.“向内共享”所描述的就是属性 *e* 逆外键方向逻辑从属于表 *A* 的这种关系,如图 4 中“品牌名称”与“店铺”.

(2) 向外共享:表 *A* 与表 *B* 因为外键关系形成多对一关联,前者可按后者(即外键 *c*)进行分组,若表 *A* 的属性 *b* 的数据类型为数值型,则表 *A* 按外键 *c* 分组并对属性 *b* 施加聚合函数(总计、平均值、最大值、最小值等)后属性 *b* 可以在逻辑上对表 *B* 所对应的概念进行描述.“向外共享”所描述的就是聚合后的属性 *b* 顺外键方向逻辑从属于表 *B* 的这种关系,这也是分组统计需求的基本结构,如图 4 中“实收金额”与“店铺”.

为了更好地帮助用户构造查询,我们应该尽量屏蔽属性与概念之间的物理从属关系,还原符合用户自然理解的逻辑从属关系,同时对属性的聚合和分组提供支持.利用本体的推理能力,我们可以完成此项任务:设置相

应的本体推理规则,根据实际存在的物理从属关系和概念之间的外键关系推导出属性与概念之间的“向外共享”和“向内共享”关系.而这样的推理规则必须脱离具体实例,建立在与特定领域无关的抽象本体模型之上,从而具备通用性,故本文方法充分利用了本体的层次抽象能力,首先设计了与特定领域无关的抽象层本体模型——查询元模型;然后,基于该元模型设置用于表达“向外共享”和“向内共享”关系的面向查询的本体推理规则.本文首先对数据库直接映射生成的本体模型进行分析,通过继承关系将本体模型抽象到查询元模型层次,抽象方式是,让本体模型中的类继承查询元模型中的“查询对象”、本体模型中的属性继承查询元模型中的“查询属性”,通过这样的过程,当用户在本体模型上进行查询操作的时候,用户查询的本体模型中的属性将会被算法看作为“查询属性”,用户查询的本体模型中的类会被看作为“查询对象”,从而将用户查询也抽象到查询元模型层次.抽象到查询元模型后,定义在查询元模型上的推理规则就能应用到本体模型中.本文系统基于查询元模型和推理规则实现可视化查询算法,在提升可用性和表达能力的同时,还有以下几点主要优势.

(1) 避免数据转换带来的信息损失.在查询元模型层次只是进行本体推理和查询语句构造,不会进行数据转换,最终的查询还是在本体模型对应的底层数据库中进行.

(2) 本体模型直接接入,提高灵活性.系统通过将本体模型抽象到查询元模型来进一步屏蔽数据库细节,该过程不需要对映射算法进行修改.因此,系统的数据通过本体模型直接接入,数据库到本体模型的映射可以使用任意算法工具实现,可以支持多个数据库到同一本体模型的映射.

(3) 本体模型只需一次接入,查询操作无额外成本.本体模型接入到系统后,系统会将其抽象到查询元模型,并利用推理规则进行查询指标推理,接入完成后,用户的查询操作不会再次进行抽象和推理,不会带来额外操作成本和计算成本.

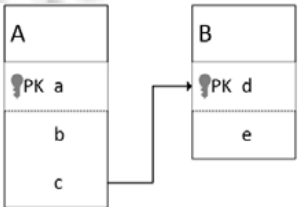

Fig.5 Foreign key relationship example

图 5 外键关联示例

2.2 方法框架

本文方法的框架如图 6 所示,基于本体的 VQS 系统以及查询构造算法建立在查询元模型和面向查询的推理规则之上,其中,VQS 利用推理规则所推导的导出关联关系可帮助用户屏蔽数据存储细节、方便用户完成查询构造,并生成用户输入,然后由查询构造算法根据用户输入还原原始的查询结构并生成相应的 SPARQL 查询语句.对于任意的领域相关的本体模型,只要以继承查询元模型的方式接入系统,即可支持面向查询的本体推理规则的运行、VQS 系统的操作和本体查询构造算法的执行.

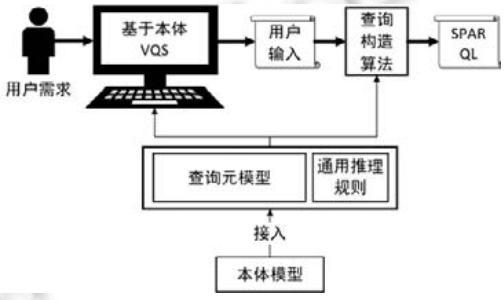

Fig.6 Method framework of this paper

图 6 本文方法框架

3 详细实现

3.1 查询元模型

为了脱离特定领域和具体实例来设置通用的推理规则,并对属性的聚合提供模型上的支持,本文方法首先设计了仅与查询行为相关、与特定领域无关的查询元模型,该元模型仍用 OWL 进行表达,如图 7 所示.其中,“查询属性”概念用于描述用户所关心的数据库表项,如“品牌名称”;“查询属性”所属的数据库表则抽象为“查询对象”概念,如“品牌”.“查询属性”与“查询对象”之间除了物理的“从属”关联,也有通过推理规则表达的逻辑上的“向内共享”关联;对于可聚合的一类“查询属性”,模型中使用“查询指标”概念对其统计值进行表达,如“营业额”(“实收金额”的统计值).“查询指标”与“查询对象”之间也有通过推理规则表达的逻辑上的“向外共享”关联.

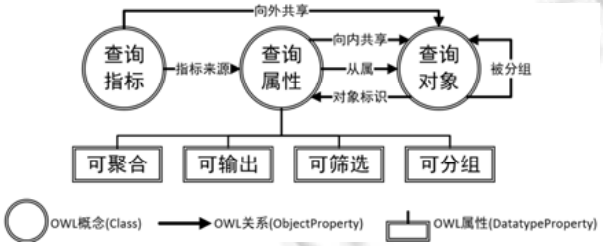

Fig.7 Query meta model diagram

图 7 查询元模型示意图

关于查询元模型中,“向内共享”“向外共享”关系在数据库中的表达,主要通过两个途径获取:第 1 种途径是通过面向查询的本体推理规则计算所有可能的“向内共享”“向外共享”关系,由用户选择所需的关系进行标注和命名,这种方式优点是无需额外信息,只需要提供原始数据库模式就能通过人工辅助的方式建立映射,缺点是用户需要对数据库模式和查询需求有一定的了解;第 2 种途径是在已经拥有一些已知的映射关系后,对于同一领域的新数据库,使用模式匹配方法自动生成映射关系,这种方式的优点是高效、无需用户操作、对用户无要求,缺点是自动生成的映射关系可能会有误,需要手动校正.

3.2 面向查询的本体推理规则

基于查询元模型的本体推理规则以查询元模型中的抽象概念、抽象关系和抽象属性作为基本元素来进行构建,运行在查询元模型之上,主要用于表达“向外共享”和“向内共享”所描述的并不直接存在于原始本体模型中的导出关系.这些本体推理规则可以帮助 VQS 系统向终端用户适当屏蔽数据库实际存储模式,还原终端用户自然理解的模型视图,并优化终端用户的查询构造流程.

以“向外共享”关系的运用为例,其对应的面向查询的本体推理规则的内容见表 1,通过该推理规则可以在“查询指标”和“查询对象”之间建立“向外共享”关联.

Table 1 Example of query-oriented ontology reasoning rules: Out share

表 1 面向查询的本体推理规则示例:向外共享

面向查询的本体推理规则示例——向外共享:
prefix:〈http://www.semanticweb.org/pku/ontologies/catering〉 [*rule_out_share*: (?*i* *rdfs*:*subClassOf*:查询指标) (?*p* *rdfs*:*subClassOf*:查询属性) (?*o* *rdfs*:*subClassOf*:查询对象) (?*ou* *rdfs*:*subClassOf*:查询对象) (?*i*:指标来源 ?*p*) (?*p*:从属 ?*o*) (?*o*:外键关联 ?*ou*) ⇒ (?*i*:向外共享 ?*od*)]

如图 8 所示,对于用户需求“查询各品牌的总营业额”,按照现有工作的构造方案,用户首先必须了解营业额的计算来源于对概念“订单”的属性“实收金额”的聚合;其次,必须依照数据库存储模式构造出用户需求中所涉及的概念(“订单”“店铺”“品牌”)、概念间的连接关系、聚合属性(“实收金额”)以及分组属性(“品牌名称”).这套

操作要求使用者熟悉数据库实际的存储模式,且具备一定的关系代数理论知识,对于没有计算机基础的终端用户来说难度较大,系统可用性差.而按照本文方法,系统首先发现查询指标“营业额”来自查询属性“实收金额”,查询中,另一个本体中的类“品牌”是一个查询对象;接着,系统通过查询元模型对“向外共享”关系的表达,在查询指标“营业额”和查询对象“品牌”之间建立了导出关系,用户可以直接从查询对象“品牌”出发选择查询指标“营业额”完成查询构造,而实际完整的本体查询结构将由系统依据对应的面向查询的推理规则反向推导而构造得到.本文方案构造流程清晰、构造内容简洁且符合终端用户的自然理解,系统可用性强.

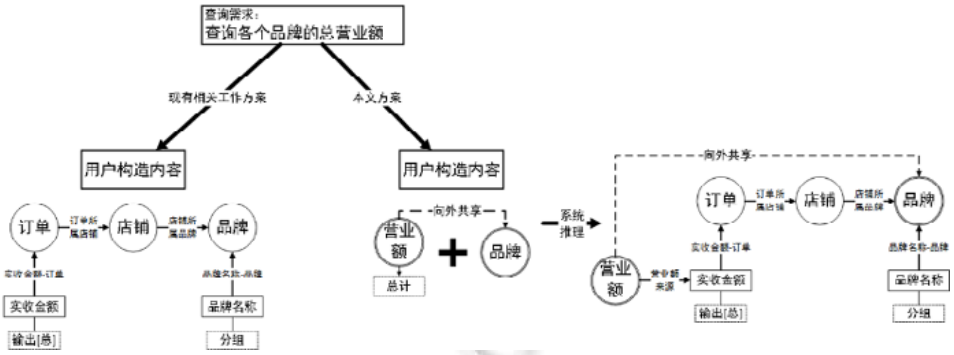

Fig.8 Example of the application of query-oriented ontology reasoning rules

图 8 面向查询的本体推理规则的运用示例

3.3 用户可视化交互设计

本文方法中,终端用户与系统的交互界面如图 9 所示,界面中主要包含查询指标树(左 1 栏)、查询对象图(左 2 栏)、查询属性表(右 2 栏)、查询元素表(右 1 栏)4 个子窗口.其中,查询指标树窗口负责展示“查询指标”;查询对象图窗口负责以图形式展示“查询对象”;查询属性表窗口负责展示与用户所点选的“查询对象”相关的“查询属性”;查询元素表窗口负责记录用户所选择的元素,对于用户所选的“查询指标”可点击选择指标的聚合方式.

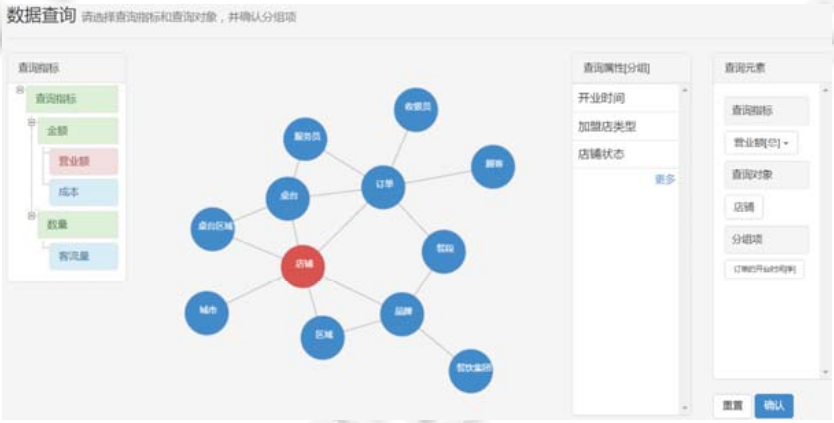

Fig.9 Interactive interface of data query system

图 9 数据查询系统的交互界面

在该界面中,用户可以通过选择查询指标和查询对象进行配合实现属性的聚合,选择查询属性实现属性的输出、筛选和分组,并最终完成用户输入.系统利用面向查询的本体推理规则所推导的导出关系帮助用户屏蔽实际的数据存储模式并优化查询构造流程,使得终端用户可以操作少量元素来表达包括分组统计查询在内的较为复杂的查询结构,系统可用性和表达能力较强.最终的用户输入 *user_input* 整合为五元组(*mname*,*ind*,*gprop*,

oprop,*fcond*)保存于系统,其中,*mname* 为所查询模型名称,*ind* 为用户所选指标集合,*gprop* 为用户所选分组属性集合,*oprop* 为用户所选输出属性集合,*fcond* 为用户所选筛选条件集合.随后,系统运行本体查询构造算法完成本体查询语句 SPARQL 的构造.

3.4 本体查询SPARQL语句构造算法

本体查询构造算法负责将用户输入转换为 SPARQL 语句,SPARQL 语句中最核心的组成成分为图模式 WHERE 部分,为了构造图模式,我们需要确定查询所涉及的所有“查询对象”以及它们之间的“外键关联”关系,本体路径搜索子算法将用于处理此任务.

本体路径搜索子算法以用户输入 *user_input* 和本体模型对象 *ont_mgr* 为输入,输出为查询所涉及的所有“查询对象”之间的“外键关联”关系的列表,即连接这些“查询对象”的路径.算法的思路是:从“查询对象”有向图中提取所有入度为 0 的“查询对象”组成起始对象集合,接下来遍历起始对象集合,对每个“查询对象”都以之为起点开始以广度优先搜索顺序遍历“查询对象”有向图,对于遍历过程中所访问到的任意“查询对象”,检查其是否与用户输入中任意“查询属性”具有“从属”关系,若有,则将该次遍历的“外键关联”关系加入结果列表.本体路径搜索子算法的伪代码如下:

算法 1. 本体路径搜索子算法(*searchOntPath*).

输入:用户输入 *user_input*,本体模型对象 *ont_mgr*.

输出:“外键关联”关系列表 *path_list*.

变量:已访问的“查询对象”集合 *visited*,宽度优先搜索队列 *queue*,起始对象集合 *start_objs*,用户所查询的模型名称 *model*

```
1.  算法开始
2.  初始化变量 visited,queue,start_objs,model
3.  if 用户输入的指标集合 ind 为空:
4.      start_objs=从 model 的查询对象有向图中所有入度为 0 的“查询对象”集合
5.  Else
6.      start_objs=用户输入的指标集中指标来源属性所从属的“查询对象”集合
7.  for each 起始对象 obj in start_objs
8.      从 obj 开始进行宽度优先搜索,遍历“查询对象”有向图中的“外键关联”edge 以及对应的对象
9.      对于每一个遍历到的“查询对象”cur_obj,检查用户输入的分组属性、输出属性和筛选条件集合
10.     if 用户输入的任意“查询属性”属于 cur_obj
11.         将 edge 加入 path_list
12. return path_list
13. 算法结束
```

SPARQL 语句中最核心的组成成分为图模式 WHERE 部分,该部分描述了查询所涉及的概念、概念之间的关系、概念的属性及对属性的筛选,需要最先进行处理,图模式构造子算法将用于完成图模式部分的构造任务.

图模式构造子算法以用户输入 *user_input*、本体模型对象 *ont_mgr* 和“外键关联”关系列表 *path_list* 为输入,输出为本体查询的图模式部分 *where*,并将图模式中使用的变量 *id* 写回用户输入 *user_input* 的输出属性和分组属性中,用于构造 SPARQL 语句中的其他部分.算法的思路是:首先遍历“外键关联”关系列表 *path_list*,为遍历到的每个“外键关联”关系和“查询对象”类都生成相应图模式三元组,并记录所有遍历到的“查询对象”组成列表;然后再遍历该“查询对象”列表,检查用户输入中从属于各“查询对象”的“查询属性”,针对这些“查询属性”生成相应图模式三元组,并判断是否需要添加筛选条件;最后返回本体查询中的图模式部分字符串.每个“查询对象”和“查询属性”都将生成唯一变量 *id*.图模式构造子算法的伪代码如下:

算法 2. 图模式构造子算法(*genSparqlWhere*).

输入:用户输入 *user_input*,本体模型对象 *ont_mgr*,“外键关联”关系列表 *path_list*.

输出:图模式 *where*.

变量:已访问的“查询变量”集合 *visited*

1. 算法开始
2. 初始化变量 *visited*,*where*
3. **for each** 外键关联 *edge* **in** 关系列表 *path_list*
4. *dom*=外键关联的主“查询对象”
5. *ran*=外键关联的从“查询对象”
6. **if** *dom* 未被访问过
7. 为 *dom* 生成唯一 *id*,并生成三元组添加到 *where* 末尾
8. **if** *ran* 未被访问过
9. 为 *ran* 生成唯一 *id*,并生成三元组添加到 *where* 末尾
10. 为 *edge* 生成三元组并添加到 *where*1 末尾
11. **for each** 查询对象 *obj* **in** *visited*
12. 遍历用户输入的分组属性、输出属性以及筛选条件集合中元素对应的“查询属性”*prop*
13. **if** *prop* 从属于 *obj*
14. 为 *prop* 生成唯一 *id*,并生成三元组添加到 *where* 末尾
15. 为 *where* 添加结尾
16. **return** *where*
17. 算法结束

完整的本体查询构造过程由本体查询构造算法负责.本体查询构造算法以用户输入 *user_input* 和本体模型对象 *ont_mgr* 为输入,输出为 SPARQL 查询语句字符串.

算法的思路是:首先,调用本体路径搜索子算法 *searchOntPath* 获得构造查询所需的“外键关联”关系列表 *path_list*;然后调用图模式构造子算法 *genSparqlWhere* 生成本体查询语句中最核心的图模式部分;接下来遍历用户输入中的“查询指标”、“输出属性”和“分组属性”来生成 SPARQL 语句中的其余部分;最后,将 SPARQL 各部分组装为整体字符串返回.本体查询构造算法的伪代码如下:

算法 3. 本体查询构造算法(*genSparql*).

输入:用户输入 *user_input*,本体模型管理器 *ont_mgr*.

输出:SPARQL 查询语句 *sparql*.

变量:查询模式 *select*,结果修饰 *groupby*

1. 算法开始
2. *sparql*=“.”
3. 运行本体路径搜索子算法得到输出 *path_list*
4. 运行图模式构造子算法得到输出 *where*
5. 初始化 *select*,*groupby*
6. **for each** 查询指标 *ind* **in** 用户输入的查询指标集
7. 为 *ind* 对应的查询属性生成查询项添加到 *select* 末尾
8. **for each** 输出属性 *prop* **in** 用户输入的输出属性集
9. 为 *prop* 生成查询项添加到 *select* 末尾
10. **for each** 分组属性 *prop* **in** 用户输入的分组属性集
11. 为 *prop* 生成查询项添加到 *select* 末尾
12. 为 *prop* 生成修饰项添加到 *groupby* 末尾
13. 为 *select* 和 *groupby* 添加结尾

14. *sparql=select+where+groupby*
15. **return** *sparql*
16. 算法结束

4 案例展示

本节以“北京宴品牌 2016 年各店铺各季度的总营业额”查询需求为例,展示本文 VQS 系统的使用操作.

终端用户(查询用户)首先需要选择所查询领域的本体模型,如在本例中,用户选择“餐饮前台信息管理”模型,选择了对应的领域模型后,用户就能开始构造查询模式.

终端用户点击进入“餐饮前台信息管理”领域的查询构造界面,如图 10 所示,用户按照需求点选指标“营业额”、点选对象“店铺”,并设置按属性“订单开台时间(季度)”分组.分组构造结束后是查询构造阶段,用户按照需求对属性“订单开台时间(年份)”设置筛选条件“等于:2016”,对属性“品牌名称”设置筛选条件按“是:北京宴”.至此完成查询构造,然后点击“确认”提交查询.

Fig.10 Data access: Design group

图 10 数据访问:分组构造

用户完成操作后,系统将会运行完整的数据访问流程获得查询结果,并进入数据展示界面,包含查询结果、SPARQL 查询、SQL 查询和数据可视化这 4 部分,具体如图 11、图 12 所示.

```
SELECT (SUM(?dp0) AS ?v_dp0) ?dp3 ?v_d2
WHERE {
        ?c0 rdfs:type:订单.
        ?cl ndfs:type:店铺.
        ?c2 rdfs:type:品牌.
        ?c3 rdfs:type:订单的实收金额.
        ?c4 rdfs:type:品牌的品牌名称.
        ?c5 rdfs:type:订单的开台时间.
        ?c6 rdfs:type:店铺的店铺名称.
        ?c0:订单店铺 ?cl.
        ?cl:店铺品牌 ?c2.
        ?c0:订单-实收金额 ?c3.
        ?c3:数据属性-订单-实收金额 ?dp0.
        ?c2:品牌-品牌名称 ?c4.
        ?c4:数据属性-品牌-品牌名称 ?dp1.
        FILTER regex(?dp1,“北京宴”)
        ?c0:订单-开台时间 ?c5.
        ?c5:数据属性-订单-开台时间?dp2.
        FILTER (YEAR(?dp2)=2016)
        ?cl:店铺-店铺名称?c6.
        ?c6:数据属性-店铺-店铺名称?dp3.
}
GROUP BY ?dp3 (QUARTER(?dp2) AS ?v_d2)
```

(a) SPARQL 构造结果

```
SELECT
        SUM(o_order_history.total)AS nc0_sum_total,
        sls_shop.shopname,
        QUARTER(o_order_history.newtime) AS nc1_quarter
FROM
        o_order_history
JOIN
        sls_shop
ON
        o_order_history.slsid=sls_shop.sid
JOIN
        sls_brand
ON
        sls_shop.bid=sls_brand.bid
WHERE
        sls_brand.brandname=‘北京宴’
AND
        YEAR(o_order_history.newtime)=2016
GROUP BY
        sls_shop.shopname,
        QUARTER(o_order_history.newtime)
```

(b) SQL 转换结果

Fig.11 Example of the contructed SPARQL and SQL queries

图 11 SPARQL 与 SQL 查询语句构造示例

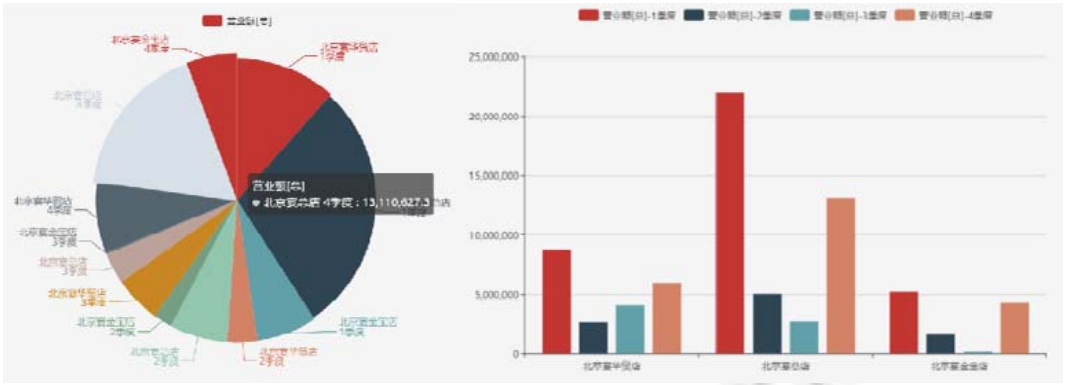

Fig.12 Data display: Sector and histogram

图 12 数据展示:扇形图和柱状图

5 实验验证

5.1 实验设定

本节对本文方法进行评估的方法为实验测试,主要评估指标为方法的可用性和表达能力.由于用户对数据库和计算机系统的了解程度非常难以量化,评估系统易用性将结合可用性和表达能力进行.在表达能力相同的情况下,可用性越高(操作越简单),则易用性越强.实验对比对象为目前最优秀的基于本体的可视化查询系统OptiqueVQS[7].实验所使用的实验材料来自于“餐饮前台信息管理”领域中,一款主流餐饮前台管理系统——“餐行健”系统的后台数据报表管理平台,本节从其中随机选取了 40 项数据查询实例,分别按照本文方法及OptiqueVQS 方法完成本体查询构造任务.为了保证用户对数据库以及计算机专业知识了解程度相同,实验中,两种 VQS 系统由同一用户操作.在实验开始之前,让用户充分熟悉系统的界面和操作,避免 UI 设计影响用户操作,用户在进行查询时,允许用户多次进行同一查询实例的查询,我们记录其操作最简单的一次操作.

5.2 实验指标

实验的主要评估指标是可用性和表达能力.

- 方法的可用性是指通过系统完成查询的操作复杂程度.

VQS 系统的操作主要有点击选择(点选)、翻页查找(翻页)以及输入信息(输入).通常情况下,认为输入操作最为复杂,翻页操作次之,点选操作比较简单.同时,对于 VQS 系统来说,可视化操作意味着鼠标操作是主要操作方式,因此点选操作占绝大部分,翻页操作较少,而输入操作一般出现在特殊查询条件的输入.不同系统操作次数一般相同,以“北京宴品牌 2016 年各店铺各季度的总营业额”为例,本文方法完成查询需要 15 次点选操作和 2 次输入操作,没有翻页操作.因此在考虑评估操作复杂程度时,我们主要是比较点选操作的数量.但为了区分不同操作的复杂性不同,在设置操作积分时,我们为点选设置积分为 1 分;翻页操作为 2 分,因为翻多页需要多次点击,但为了排除列表顺序对积分的影响,我们按实验过程中统计的平均翻页次数 2 次计算翻页操作的积分;而输入操作为 3 分,这是因为输入操作比其他操作复杂,但数量较少且不同系统差异较小,不应该在总积分中占较大比重.对于同样的查询需求,可用性高的 VQS 系统用户操作较为简单,反之较为复杂.

- 方法的表达能力是指通过系统可视化操作构造的查询语句的能力.

可视化查询方法相对于语句查询方法在简化用户查询操作难度的同时,往往会带来表达能力的损失,即部分查询语句无法通过可视化操作的方式构造.不同的 VQS 系统一般具有不同的表达能力,表达能力高的 VQS 系统能构造出更多复杂查询语句,能更好地满足用户的查询需求.

两种指标的量化计算步骤如下.

1. 数据查询实例数量 N=40.

2. 对于第 i 条查询实例,在查询构造所需的操作中点选次数记为 C_i、翻页次数记为 P_i、输入次数记为 I_i.给点选、翻页和输入操作分别设置 M_c=1,M_p=2,M_I=3 的操作积分,积分高的操作更为复杂.再根据所选系统能否正确构造出查询语句来设置正确性标志 A_i:

$$A_i=\begin{cases}0, & \text{可以正确构造}\\ 1, & \text{无法正确构造}\end{cases}.$$

3. 若VQS系统不能正确构造查询语句,则假设用户最后可以通过输入正确查询语句的方式更正查询,实验中首先构造出最接近查询目标的查询,然后增加一次更正操作(输入操作).

4. 对于第 i 条查询实例,方案的操作积分 Q_i 的计算公式为

$$Q_i=C_i\times M_c+P_i\times M_p+(I_i+A_i)\times M_I.$$

其含义为:根据操作次数计算出每种操作的总积分,再将点击、翻页、输入这 3 种操作的积分求和作为最终操作积分.

5. 对于第 i 条查询实例,方案的表达积分 E_i 的计算公式为

$$E_i=\frac{C_i\times M_c+P_i\times M_p+I_i\times M_I}{O_i}=\frac{O_i-A_i\times M_I}{O_i}.$$

其含义为:除更正操作外,其他操作总分占操作积分的比例.因此,如果可以正确构造查询语句,表达积分为1;不能正确构造查询语句,则积分小于 1.由于 Q_i 与要构造的查询的复杂程度基本正相关,这保证了在无法正确构造的情况下,要构造的查询越简单,积分越接近于 0,即惩罚无法正确构造的简答查询.

6. 记两种 VQS 系统所有查询实例中操作积分最高为 $Q_{\max}$.

7. 方案可用性的计算公式为

$$Acc=\left(1-\frac{\sum_{i=1}^{N}O_i}{O_{\max}\times N}\right)\times 100.$$

8. 方案表达能力的计算公式为

$$Exp=\left(\frac{\sum_{i=1}^{N}E_i}{N}\right)\times 100.$$

5.3 实验结果

实验结果中,两种评估指标如图 13 所示.

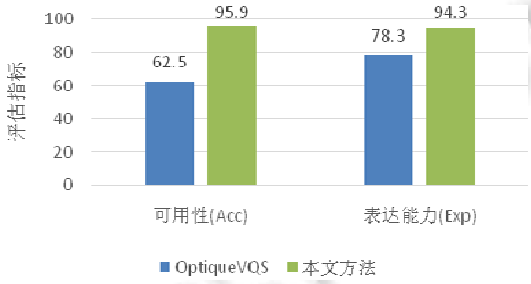

Fig.13 Method evaluation results of this paper

图 13 本文方法评估结果

对于实验结果的分析如下.

(1) 本文方法的可用性量化值高于OptiqueVQS方法,说明终端用户利用本文方法进行查询构造所需要的操作复杂度更低.形成这样的实验结果的原因在于:本文方法充分利用了本体模型的推理能力来优化查询构造流程,用户可以依赖由面向查询的本体推理规则所推导出的导出关系来完成查询构造,减少

了构造查询所需要确认的模型元素,而实际的完整查询结构则由系统根据对应的推理规则反向推导得到,用户不需要了解数据库系统的实际存储模式.该结果验证了本文方法具有更强的可用性.

(2) 本文方法的表达能力量化值高于OptiqueVQS方法,该差异主要来源于本文方法对终端用户的分组统计需求进行了支持,而OptiqueVQS方法未能对之进行相应支持.实验共40个查询实例,其中4个查询实例本文方法无法正确构造,其原因在于这些查询需要对多条查询语句的结果进行组合得到;12个查询实例OptiqueVQS方法无法正确构造,除前面提到的4个查询实例外,还有8个查询实例中涉及到了分组统计,而OptiqueVQS无法可视化构造分组查询.该结果验证了本文方法具有更强的表达能力.

6 结 论

本文以帮助终端用户构造本体查询语句SPARQL为基本目标,首先调研了相关的研究工作,分析并总结了相关工作的主要内容与不足之处,不足之处主要在于直接暴露数据库实际存储模式且不能支持分组统计需求;然后,针对终端用户的真实数据访问需求以及现有相关工作的不足,提出并实现了一种基于推理的终端用户本体查询构造方法;最后,通过“餐饮前台信息管理”领域的实际案例对本文方法进行了验证.

本文方法面向没有计算机基础的终端用户,充分利用本体模型的语义表达能力和推理能力来优化终端用户的查询构造流程,帮助用户脱离数据库的实际存储模式细节;同时,对终端用户的分组统计需求提供了支持,填补了现有相关工作的不足,具备更强的可用性和表达能力.

本文工作未来的一个研究方向是实现对复杂统计指标的支持.在实际应用场景中,终端用户所关心的一些复杂统计指标无法直接单次构造得到,需要对多条查询进行组合,例如店铺的“客均营业额”指标需要由店铺的“营业额”和“客流量”这两项指标联合计算得到,而本文方法尚不支持对多条查询进行组合.故本文未来需要研究“多条查询的组合和运算”,首先,在模型上增加元素对其进行表达;然后,在系统中添加相应功能,实现对单条查询之间类似四则运算的组合机制.

References:

[1] Dickson GW, Desanctis G. Information Technology and the Future Enterprise: New Models for Managers. Prentice Hall, 2000.

[2] Hameed M, Qamar U, Akram MU. Business intelligence: Self adapting and prioritizing database algorithm for providing big data insight in domain knowledge and processing of volume based instructions based on scheduled and contextual shifting of data. In: Proc. of the IEEE Future Technologies Conf. IEEE, 2016. 1168−1175.

[3] Berndtsson M. Analyzing business intelligence maturity. Journal of Decision Systems, 2015,24(1):37−54.

[4] Chaudhuri S, Dayal U, Narasayya VR. An overview of business intelligence technology. Communications of the ACM, 2011,54(8): 88−98.

[5] Duan L. Business intelligence for enterprise systems: A survey. IEEE Trans. on Industrial Informatics, 2012,8(3):679−687.

[6] Giese M, Ozcep O, Rosati R, *et al*. Optique: Zooming in on big data. Computer, 2015,48(3):60−67.

[7] Soylu A. Experiencing OptiqueVQS: A multi-paradigm and ontology-based visual query system for end users. Universal Access in the Information Society, 2016,15(1):129−152.

[8] Baeza-Yates R, Ribeiro-Neto B. Modern Information Retrieval. New York: Addison-Wesley Publishing Co., 1999.

[9] Giese M, Calvanese D, Haase P, *et al*. Scalable end-user access to big data. In: Proc. of the Big Data Computing. 2013. 205−244.

[10] Catarci T. Visual query systems for databases. Journal of Visual Languages & Computing, 1997,8(2):215−260.

[11] Zloof MM. Query by example: A data base language. IBM Systems Journal, 1977,16(4):324−343.

[12] Erwig M. Xing: A visual XML query language. Journal of Visual Languages & Computing, 2003,14(1):5−45.

[13] Jiménez-Ruiz E, Kharlamov E, Zheleznyakov D, *et al*. BootOX: Practical mapping of RDBs to OWL 2. In: Proc. of the ISWC. 2015. 113−132.

[14] Civili C. Mastro studio: Managing ontology-based data access applications. Proc. of the VLDB Endowment, 2013,6(12):1314−1317.

[15] Tableau Website. http://tableau.com

[16] Catarci T, Mascio TD, Franconi E, *et al*. An ontology based visual tool for query formulation support. In: Proc. of the ECAI 2004. IOS Press, 2004. 308−312.

[17] Barzdins G, Liepins E, Veilande M, *et al*. Ontology enabled graphical database query tool for end-users. In: Proc. of the Databases and Information Systems V—Selected Papers from the 8th Int'l Baltic Conf. (DB&IS 2008). Tallinn: DBLP, 2008. 105−116.

[18] Lu JJ, Zhang YF, Miao Z. Principles and Techniques of Semantic Web. Beijing: Science Press, 2007 (in Chinese).

[19] McBride B. Jena: A semantic Web toolkit. IEEE Internet Computing, 2002,6(6):55−59.

[20] Consortium WWW. A direct mapping of relational data to RDF. 2012. https://www.w3.org/TR/rdb-direct-mapping/

[21] Souripriya Das O. R2RML: RDB to RDF mapping language. 2011. https://www.w3.org/TR/r2rml/

[22] Prud'Hommeaux E, Seaborne A. SPARQL query language for RDF. 2007. https://www.w3.org/TR/rdf-sparql-query/

[23] Press R. Ontology and database mapping: A survey of current implementations and future directions. Journal of Web Engineering, 2008,7(1):1−24.

[24] Spanos DE, Stavrou P, Mitrou N. Bringing relational databases into the semantic Web: A survey. Semantic Web, 2012,3(2): 169−209.

[25] Kogalovsky MR. Ontology-based data access systems. Programming and Computer Software, 2012,38(4):167−182.

[26] Stardog website. http://stardog.com

[27] Bizer C, Seaborne A. D2RQ—Treating non-RDF databases as virtual RDF graphs. In: Proc. of the ISWC 2004. LNCS 3298, 2004.

附中文参考文献:

[18] 陆建江,张亚非,苗壮.语义网原理与技术.北京:科学出版社,2007.

唐爽(1995－),男,湖北巴东人,硕士生,CCF 学生会员,主要研究领域为智慧城市.

王江涛(1987－),男,博士,助理教授,CCF 专业会员,主要研究领域为群智感知,普适计算,移动计算.

王亚沙(1975－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为普适计算,大数据分析技术.

夏丁(1992－),男,硕士,主要研究领域为普适计算.

赵俊峰(1974－),女,博士,副教授,CCF 高级会员,主要研究领域为软件工程,软件复用,Web 服务,普适计算,大数据分析技术.