



## 基于本体分割的语义图概要方法

王 艺,王 英

(西南大学 计算机与信息科学学院,重庆 400715)

**摘 要:** 语义图概要的目的是提取语义图的关键信息,形成原数据集的概要模型以解决大规模语义图的理解、查询、应用难题。为提升现有语义图概要方法效率,提出一种基于本体分割的概要方法。通过本体分割算法对语义图进行分割生成扩展子图。采用形式概念分析对每个扩展子图生成元素的偏序格(又称特征集格)。在此基础上,由所有子图的特征集格形成了原语义图的概要。在关联开放数据集和 Berlin SPARQL Benchmark 数据集上的实验结果表明,该方法具有较好的可扩展性,有效提高了概要方法的效率。

**关键词:** 语义图;知识图谱;关联开放数据;语义图概要;形式概念分析

开放科学(资源服务)标志码(OSID):



中文引用格式:王艺,王英.基于本体分割的语义图概要方法[J].计算机工程,2021,47(10):67-74.

英文引用格式:WANG Y, WANG Y. Approach of semantic graph summarization based on ontology partition[J]. Computer Engineering, 2021, 47(10): 67-74.

## Approach of Semantic Graph Summarization Based on Ontology Partition

WANG Yi, WANG Ying

(School of Computer and Information Science, Southwest University, Chongqing 400715, China)

**[Abstract]** Semantic graph summarization is to extract key information from semantic graphs, and generate a summarized model of the original data set to solve problems in understanding, querying, and using large-scale semantic graphs. In order to improve the efficiency of current summarization algorithms, this paper proposes an approach of generating summaries based on ontology partition. The ontology partition algorithm is used to divide the semantic graph into sub-graphs. Then for each sub-graph, its partially ordered lattices (also named characteristic set lattices) of elements are generated using formal concept analysis. On this basis, the characteristic set lattices of all sub-graphs form the summarization of the original semantic graphs. The approach is tested on the Linked Open Data (LOD) dataset and the Berlin SPARQL Benchmark datasets. Results show that the proposed approach exhibits excellent scalability, and a significant improvement in summarization efficiency.

**[Key words]** semantic graph; knowledge graph; Linked Open Data (LOD); Semantic Graph Summarization (SGS); Formal Concept Analysis (FCA)

DOI: 10.19678/j.issn.1000-3428.0059083

### 0 概述

知识图谱是基于语义技术的大规模语义图知识库<sup>[1-2]</sup>。截至2019年3月关联开放数据云(Linked Open Data Cloud)已有1 239个数据集和16 147条链接,均以资源描述框架(Resource Description Framework, RDF)三元组形式对数据进行描述<sup>[3]</sup>。例如,Facebook的开放内容协议<sup>[4]</sup>、谷歌的知识图谱<sup>[5]</sup>、DBpedia<sup>[6]</sup>、Wikidata<sup>[7]</sup>等知识库,其包含的三元组已达数百万条,甚至上亿条。语义图广泛应用在不同领域<sup>[2]</sup>,如教育领域已相继启动关联开放数据(Linked Open Data, LOD)项目

mEducator<sup>[8]</sup>、Open University<sup>[9]</sup>、LAK Dataset<sup>[10]</sup>等,旨在共享教育数据(如课程数据、统计数据、教育资源(视频和文档等))。

由于语义图的规模巨大、结构复杂且缺乏标准模式,用户及应用程序开发人员面临理解、使用等难题。语义图概要(Semantic Graph Summarization, SGS)是解决该难题的技术,也是当前语义图领域的研究热点<sup>[11]</sup>。SGS将大规模语义图压缩为一个缩略图,保留原有信息的基础上压缩其规模。SGS依赖于一系列技术,包括有向图特征提取、统计方法、模式挖掘、代数结构等。例如,基于形式概念分析(Formal Concept Analysis,

基金项目:西南大学教育教学改革研究项目(2019JY048);第47批留学回国人员科研启动基金。

作者简介:王 艺(1978—),女,副教授、博士,主研方向为语义网、知识工程;王 英,讲师、博士研究生。

收稿日期:2020-07-28 修回日期:2020-10-15 E-mail: blackandgreens@163.com

FCA)<sup>[12]</sup>的语义图概要方法<sup>[13-15]</sup>是一种利用代数结构构建语义图概要的方法。通过定义语义图中的相应元素为形式概念,并建立概念之间的偏序关系形成一个偏序格。

虽然各种SGS方法提供了针对大规模语义图的概要模型,能够大幅提升存储和查询效率。由于语义图规模巨大,使得计算语义图概要开销增大,尤其当语义图数据在固定周期更新后,需要重新计算语义图概要。

为提高针对大型语义图SGS的计算效率问题,本文提出一种基于本体分割的SGS方法。根据一定标准将语义图分割成多个子图,以达到降低原语义图规模的目的。同时利用FCA方法计算每个语义子图的SGS,使所有语义子图的SGS构成原图的概要。

## 1 相关工作

语义图以RDF三元组形式存储,由于缺乏标准模式,大规模语义图结构复杂,使得提升查询、使用效率、理解语义图成为当前亟需解决的难题。SGS通过对语义图进行压缩生成原语义图的结构和模式,以解决大规模语义图的使用问题<sup>[11]</sup>。当前SGS方法主要分为基于图结构的方法和基于代数结构的方法。

### 1) 基于图结构的方法

根据RDF三元组所形成有向图的结构,对图中的节点进行划分或提取形成概要节点,所有概要节点及相应的边构成原语义图概要。文献[16]提出节点间的“强”和“弱”等价关系,将等价的节点归纳为一个节点,进而形成语义图的熵图,即概要图。文献[17]提出一种称为 $k$ -概要图的方法,通过对语义图节点集合进行划分得到 $k$ 个子集,每个子集作为一个概要节点。概要节点间的边设置了权重,描述原语义图中概要节点之间边的数量。文献[18]定义类节点的中心度和频率,以确定本体模式(类及类之间的层次关系)中类的重要性。中心度通过计算类所关联属性边的权重获取;频率用于衡量类在各本体中的使用比率。选取 $k$ 个最重要的类及其相关的类作为本体模式概要。文献[19]提出一种 $d$ -概要,描述语义图不超过 $d$ 步邻接节点的关联模式,语义图的所有 $d$ -概要即是语义图概要。通过定义 $d$ -概要的信息量指标,基于频繁子图挖掘技术,提出计算 $k$ 个最大化信息量的 $d$ -概要算法。

### 2) 基于代数结构的方法

利用FCA构建语义图的代数结构,生成概念格作为语义图的概要。概念格本质是概念之间的偏序关系。文献[13]提出一种基于FCA的模型——特征集格(Characteristic Set Lattice, CSL)。根据三元组 $(s, p, o)$ 的主语和谓语实体生成特征集概念,进而形成特征集格,实现对语义图的概要。文献[14]中FCA被用于对语义图类节点和属性边进行分类,分别形成类节点集合与属性边集合上的偏序集,最后生成模式概念格作为语义图的概要。文献[15]提出一种扩展的FCA模型——G-FCA,增加了概念格中对语义图三元组宾语实体的描述功能。

上述两类SGS方法均能在压缩原语义图规模的同时保留某些关键信息,实现帮助用户理解及提升

查询效率的目的。目前语义图规模巨大,包含的三元组已达到数百万,甚至上亿条,使得计算语义图概要的开销巨大。例如,基于FCA概要生成算法的时间复杂度为 $O(N^3)$ <sup>[13]</sup>,其中 $N=\max\{|S|, |P|\}$ , $S$ 为图中RDF三元组的主语节点集合, $P$ 为属性集合。文献[16]提出基于等价划分节点的方法,其时间复杂度为 $O(|V_i| \cdot (|V_i| + |P|^2))$ ,其中: $V_i$ 为语义图的实例节点集合; $P$ 为属性集合。文献[19]提出基于模式挖掘的方法,其时间复杂度为 $O(N_s \cdot |V| \cdot |E|)$ ,其中: $N_s$ 为概要的规模,即概要中包含的节点和边数; $V$ 为语义图的节点集合; $E$ 为边集合。因此,对于大型语义图中上述两类SGS方法的时间复杂度较高,当语义图数据在固定周期更新后,上述SGS方法需要重新计算概要图。因此,SGS方法的效率亟需进一步提升。

基于本体分割的CSL概要方法如图1所示。首先利用本体分割算法对语义图的节点进行划分,进而生成扩展子图。本体分割的目的是将大规模本体分割为多个子本体,以解决本体查询、维护、重用等问题<sup>[20-22]</sup>。基于网络模型的本体分割方法将本体视为有向图,依据其结构进行划分,最终得到多个子图,即子本体。例如,文献[23]将本体模式转化为依赖关系图,基于依赖关系的强弱,将图的节点集进行自动划分再生成连通的子图。该方法的优点在于其划分过程使用大型网络分析工具Pajek<sup>[24]</sup>的提取线岛(Line Island)功能,时间复杂度控制在 $O(n\sqrt{n})$ ,其中 $n$ 为图的节点数。语义图的本质就是本体,因此利用本体分割方法对大型语义图进行分割是可行的。其次针对每个由分割生成的扩展子图,基于FCA构造相应的格概要。所有子图的概要组成了原语义图的概要。

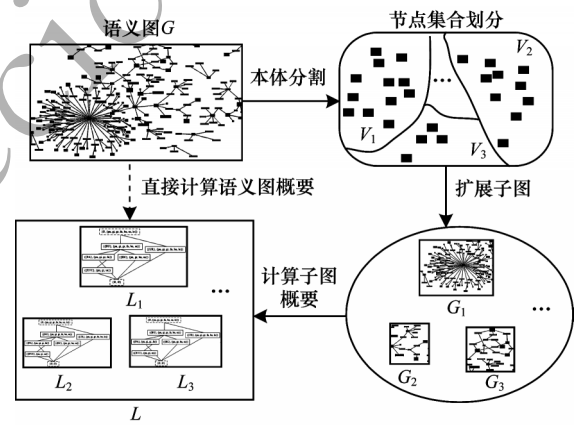


图1 基于本体分割的CSL概要

Fig.1 CSL summary based on ontology partition

SGS方法是直接生成语义图的概要(如图1虚线箭头所示),而本文是利用本体分割算法先划分语义图,再构造各语义子图的概要。SGS方法的复杂度主要由计算等价、相似节点、计算元素间的偏序产生,而本文方法经过分割步骤,针对语义子图进行计算概要图,大幅度降低了计算SGS的时间消耗。虽然本体分割步骤有一定的计算开销,但其复杂度可控制在 $O(n\sqrt{n})$ ,使得计算SGS的总体效率高于一一般SGS方法。

## 2 基于本体分割的 SGS

### 2.1 语义图的 CSL 概要

设  $G=(V,E,P,\lambda)$  是一个语义图,其中:  $V=I \cup L \cup B$ ,  $I$  为 URI 的集合;  $L$  为字面量集合;  $B$  为空白节点集合。  $E=\{(u,v) | u,v \in V\}$  是属性边的集合。  $\lambda$  是标号函数,  $\lambda: E \rightarrow \wp(P)$ , 其中:  $\wp(P)$  为属性集合的幂集。  $G_i=(V_i,E_i,P_i,\lambda_i)$  是语义图  $G$  的实例图, 其中  $V_i \subseteq V$  是所有实例、字面量和空白节点构成的集合,  $E_i \subseteq E$  是  $V_i$  之间的关系。

**定义 1** (形式上下文) 表示为  $X=(S,P_i,R)$ , 其中  $S$  是所有  $G_i$  中三元组主语构成的集合, 称为实体集合,  $P_i$  是  $G_i$  中所有三元组属性构成的集合,  $R$  是  $S$  和  $P_i$  中元素的关系, 即  $R=\{(s,p) | \exists o, (s,p,o) \in G_i\}$ 。

**定义 2** (特征集合) 给定形式上下文, 对任意  $s \in S$ ,  $CS(s)=\{p | \exists o, (s,p,o) \in G_i\}$ ,  $CS(s)$  是实体  $s$  的特征集合。

**定义 3** (特征集概念) 给定形式上下文,  $c=(S',T)$  被称为一个特征集概念, 需满足以下条件: 1)  $S' \subseteq S$ ; 2)  $T \subseteq P_i$ ; 3) 对任意的  $s \in S'$ ,  $CS(s)=T$ 。

令  $C$  表示  $X$  中所有的特征集概念集合, 则  $|C|$  是  $X$  中特征集概念的个数, 其取值范围为:  $1 \leq |C| \leq \min\{|V_i|, |\wp(P_i)|\}$ 。若  $|C|=1$ , 表明  $\forall v_i, v_j \in V_i$ , 且  $v_i \neq v_j$ , 均有  $CS(v_i)=CS(v_j)$ , 即  $G_i$  中所有实例有相同的特征集合。若  $|C|=\min\{|V_i|, |\wp(P_i)|\}$ , 表明  $\forall v_i, v_j \in V_i$ , 且  $v_i \neq v_j$ , 当  $|V_i| \leq |\wp(P_i)|$  时,  $CS(v_i) \neq CS(v_j)$ , 即不同的实例有不同的特征集合; 当  $|V_i| > |\wp(P_i)|$  时, 不同实例的不同特征集合达到最大值  $|\wp(P_i)|$ 。因此,  $|C|$  的数值描述了语义图数据的规范性, 数值越小, 则语义图数据越规范; 反之, 语义图数据越不规范。

**例 1** 语义图的形式上下文及特征集格如图 2 所示。由英国学术公开数据服务<sup>[8]</sup>提供的 LOD 数据如图 2(a)。  $X=(S,P_i,R)$ , 其中  $S=\{BU, RVC, UR, HC, PA\}$ ,  $P_i=\{pn, gi, gr, fn, bn, lo, sn\}$ , 从图 2(a) 可以看出, 矩阵显示了关系  $R$ , “ $\times$ ” 表示实体和属性之间有关系。例如, 与 RVC 实例有关的关系为:  $(RVC, pn), (RVC, gi), (RVC, sn)$ 。实体 RVC 的特征集合  $CS(RVC)=\{pn, gi, sn\}$ 。图 2(a) 包含的特征集概念集合  $C=\{c_1, c_2, c_3, c_4, c_5\}$ , 其中  $c_1=(\{BU\}, \{pn, gi, gr, fn, bn, sn\})$ ,  $c_2=(\{RVC\}, \{pn, gi, gr\})$ ,  $c_3=(\{UR\}, \{pn, gi, gr, bn, lo\})$ ,  $c_4=(\{HC\}, \{pn, gr, bn, sn\})$ ,  $c_5=(\{PA\}, \{pn, gi, gr, fn\})$ 。本例中, 特征集合数  $|C|=|V_i|=5$ , 即不同的实例有不同的特征集合, 表明该语义图数据不规范。

设  $\subseteq$  为特征集概念元素  $T$  之间的包含关系, 则  $(C, \subseteq)$  是一个偏序集。一般情况下, 由于缺乏最大元和最小元, 该偏序集不是格。为了使  $(C, \subseteq)$  成为一个格, 需要增加两个元素到  $C$  中, 分别是:  $(\emptyset, P_i)$  和  $(\emptyset, \emptyset)$ , 其中  $(\emptyset, P_i)$  为格的最大元,  $(\emptyset, \emptyset)$  为格的最小元。该偏序格被称为特征集格 CSL。

**定义 4** (CSL 概要) 给定语义图  $G$ , 设  $(C, \subseteq)$  为特征集概念集合上定义的偏序集。令  $C'=C \cup (\emptyset, P_i) \cup (\emptyset, \emptyset)$ , 则  $(C', \subseteq)$  为一个偏序格, 该偏序格称为语义图  $G$  的 CSL 概要。

当  $|C|=1$  时, 语义图数据是规范的, CSL 概要只有  $|C|+2=3$  个元素:  $(\emptyset, P_i), (V_i, CS(V_i))$  和  $(\emptyset, \emptyset)$ 。当  $|C|=\min\{|V_i|, |\wp(P_i)|\}$ , CSL 概要包含了  $|C|+2$  个元素。

因此, 数据的规范性可由 CSL 概要包含的元素反映, 数据越规范, CSL 概要包含的元素越少, 反之则越多。

**例 2** 例 1 中的  $(C, \subseteq)$  是一个偏序集, 其诱导的格如图 2(b) 所示。该格是图 2(a) 所示 LOD 数据的 CSL 概要, 包含  $|C|+2=7$  个元素。

	BU	RVC	UR	HC	PA	实例缩写对照:
pn	x	x	x	x	x	BU: 伯恩茅斯大学 RVC: 皇家兽医学院 UR: 雷丁大学 HC: 海斯洛普学院 PA: 亚伯大学
gi	x	x	x		x	
gr	x		x	x	x	
fn	x				x	
bn	x		x	x		
lo			x			
sn	x	x		x		属性缩写对照: pn: provider_name gi: GTR_ID gr: group fn: flat_name_number bn: building_name_number lo: locality sn: street_name

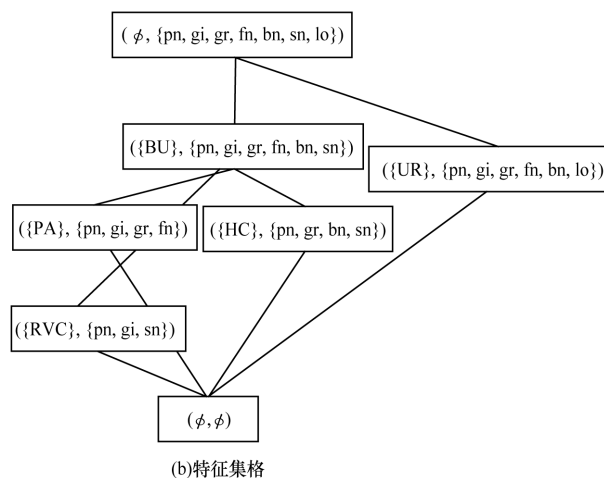
```

@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix oo: <http://purl.org/openorg/> .
@prefix vcard: <http://www.w3.org/2006/vcard/ns#> .
@prefix ospost: <http://data.ordnancesurvey.co.uk/ontology/postcode/> .
@prefix lprov: <http://id.learning-provider.data.ac.uk/terms#> .
@prefix spatialrelations: <http://data.ordnancesurvey.co.uk/ontology/spatialrelations/> .

<http://id.learning-provider.data.ac.uk/ukprn/10007779>
  a foaf:Organization ; lprov:UKUniversity ;
  rdfs:label "The Royal Veterinary College" ;
  skos:notation "10007779"^^<http://id.learning-provider.data.ac.uk/ns/UKPRNSchemeDatatype#> ;
  oo:primaryContact <http://id.learning-provider.data.ac.uk/ukprn/10007779#contact> ;
  ospost:postcode <http://data.ordnancesurvey.co.uk/id/postcodeunit/NW10TU> ;
  vcard:adr <http://id.learning-provider.data.ac.uk/ukprn/10007779#address> .
<http://id.learning-provider.data.ac.uk/ukprn/10007779#address>
  a vcard:Address ;
  vcard:postal-code "NW1 0TU" ;
  vcard:street-address "ROYAL COLLEGE STREET" .

```

(a)形式上下文及示例三元组



(b)特征集格

图 2 语义图的形式上下文及特征集格

Fig.2 Formal context and characteristic set lattice of semantic graph



## 2.2 语义图分割

语义图分割是通过划分大型语义图规模为合适的子图,从而达到提高语义图概要计算效率的目的。

**定义5(语义图分割)** 设  $G_I=(V_I, E_I, P_I, \lambda_I)$  是实例图,  $\pi=\{V_1, V_2, \dots, V_s\}$  为  $G_I$  的一个分割, 需满足3个条件: 1)  $\emptyset \neq V_i \subseteq V_I$ ; 2)  $V_i \cap V_j = \emptyset$ ; 3)  $\bigcup_{i \in \{1, 2, \dots, s\}} V_i = V_I$ 。

**定义6(语义图的扩展子图)** 设  $G_I=(V_I, E_I, P_I, \lambda_I)$  是  $G=(V, E, P, \lambda)$  的实例图,  $\pi=\{V_1, V_2, \dots, V_s\}$  为  $G_I$  的分割, 称  $G_i=(W_i, E_i, P_i, \lambda)$  ( $i=1, 2, \dots, s$ ) 为  $V_i$  所诱导的扩展子图, 需满足3个条件: 1)  $W_i=V_i \cup VB_i$ , 其中  $VB_i=\{v | \exists u \in V_i, (u, v) \in E\}$  为边界节点集合; 2)  $E_i=\{(u, v) | (u, v) \in E \text{ 且 } u, v \in V_i\}$ ; 3)  $P_i=\{p | \lambda(u, v)=p \text{ 且 } u, v \in V_i\}$ 。

上述定义是关于语义图的分割, 其本质是对语义图节点集合进行划分。集合的划分方案数较多, 选取合适的划分标准是对语义图进行合理分割的首要问题。文献[23]提出一种划分本体模式的方法, 通过将类之间的层次关系转换为赋权图(又称依赖图), 再进行图的线岛划分。该划分算法的本质是将依赖紧密的节点划分在一个块中。由于本文考虑的是大型语义图的实例图划分, 无法直接使用文献[23]所提出的算法, 因此改进了该本体模式划分算法。

首先将语义实例图转换为依赖图; 其次根据依赖图将节点集合划分为线岛, 作为节点的分割; 最后计算扩展子图。

**定义7(语义图的依赖图)** 给定  $G_I=(V_I, E_I, P_I, \lambda_I)$  是实例图, 有向图  $G_I^D=(V_I, E_D, w^e)$  是依赖图, 其中:  $E_D$  为依赖图的边集合;  $w^e$  为边上的权重集合。若  $(v_i, v_j) \in E_I$ , 则  $(v_i, v_j) \in E_D$  且  $(v_j, v_i) \in E_D$ 。每条边  $(v_i, v_j)$  有预设权重  $w_{ij}^p$ , 它的值由其关联的属性确定(默认值为1)。对于任意的  $(v_i, v_j) \in E_D$ , 其依赖图的权重  $w(v_i, v_j)$  如式(1)所示:

$$w(v_i, v_j) = \frac{w_{ij}^p + w_{ji}^p}{\sum_k (w_{ik}^p + w_{ki}^p)} \quad (1)$$

定义7中每条边的预设权重用来描述不同属性的重要性, 可由用户定义, 其默认值为1。若所有预设权重为1, 并注意到依赖图的所有边都是对称出现的, 则式(1)可简化为式(2), 其中  $d^+(v_i)$  表示  $v_i$  的出度, 如式(2)所示:

$$w(v_i, v_j) = \frac{1}{d^+(v_i)} \quad (2)$$

式(1)和式(2)计算了节点  $v_i$  到  $v_j$  的依赖强度, 其原理是基于社会网络结构洞理论中比例强度关系网络的相关结论<sup>[25]</sup>。以式(2)为例, 若节点  $v_i$  只与  $v_j$  邻接, 则  $v_i$  完全依赖于  $v_j$ , 这时  $w(v_i, v_j)=1$ ; 若节点  $v_i$  与包括  $v_j$  的  $k$  个节点邻接, 则  $v_i$  依赖于  $v_j$  的程度与其出

度成反比, 即  $w(v_i, v_j)=1/k$ 。因此, 式(1)及式(2)体现了节点之间的依赖强弱程度。在后面的本体分割步骤中, 这样的权重设置可以将依赖紧密的节点划分在一个分块中。

**定义8(依赖图的线岛)** 设  $G_I^D=(V_I, E_D, w^e)$  是依赖图,  $L_s \subseteq V_I$  是一个线岛当且仅当  $L_s$  能诱导一个  $G_I^D$  的连通子图, 存在一个  $G_I^D$  的生成树  $T=(V_T, E_T, w^e_T)$ , 满足:

$$\max \{w(v, v') | (v, v') \in E_D \wedge ((v \in L_s \wedge v' \notin L_s) \vee (v \notin L_s \wedge v' \in L_s))\} < \min \{w(u, u') | (u, u') \in E_T\} \quad (3)$$

**例3** 语义图分割过程及扩展子图如图3所示。从图3可以看出, 一个语义实例图  $G_I$ , 包含9个三元组, 9个实体。首先将  $G_I$  转换为依赖关系图。设所有边的预设权重为1, 根据式(2)可得依赖图各边的权重。节点间的依赖关系和权重。例如, 以节点  $v_4$  为起点的4条边的权重为  $w(v_4, v_1)$ 、 $w(v_4, v_2)$ 、 $w(v_4, v_3)$ 、 $w(v_4, v_5)$ , 根据式(2)其权重均为0.25, 表示  $v_4$  依赖于  $v_1, v_2, v_3, v_5$  每个节点的程度是0.25。依据依赖图权重所体现的节点连接强弱, 由定义8可将语义图的节点划分为两个节点集:  $V_1=\{v_1, v_2, v_3, v_4\}$  和  $V_2=\{v_5, v_6, v_7, v_8, v_9\}$ 。由该节点的划分可按定义6得到相应的扩展子图  $G_1$  和  $G_2$  (注意到  $G_2$  包含一个边界节点  $v_5$ )。

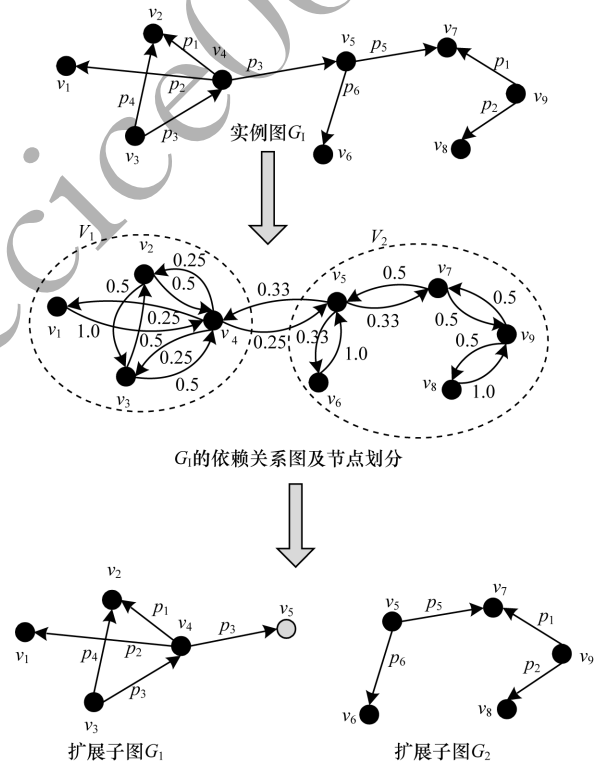


图3 语义图分割过程及扩展子图

Fig.3 Semantic graph segmentation process and extended sub-graphs

## 2.3 基于本体分割的 CSL 概要

基于本体分割的 CSL 概要包括如下定义:

**定义9**(基于本体分割的 CSL 概要) 设  $G_i=(V_i, E_i, P_i, \lambda_i)$  是语义图  $G$  的实例图,  $\pi=\{V_1, V_2, \dots, V_s\}$  为  $G_i$  的分块,  $G_i(i=1, 2, \dots, s)$  为  $V_i$  对应的扩展子图。由  $G_i$  定义的 FC 为  $X_i=(S_i, P_i, R_i)$ , 则针对每个  $X_i$  可构造相应的 CSL:  $L_i(i=1, 2, \dots, s)$ 。由各子图的 CSL 构成的集合  $L=\{L_1, L_2, \dots, L_s\}$  为语义图  $G$  的概要。

**例4** 语义图经分割为两个扩展子图  $G_1$  和  $G_2$ , 根据2.2节的方法, 可分别产生  $L_1$  和  $L_2$  为  $G$  的 CSL 概要。 $G_1$  的特征集概念为  $c_1=(\{v_3\}, \{p_3, p_4\})$  和  $c_2=(\{v_4\}, \{p_1, p_2, p_3\})$ , 相应 CSL 为  $L_1=\{((\emptyset, \emptyset), c_1), ((\emptyset, \emptyset), c_2), (c_1, (\emptyset, P_1)), (c_2, (\emptyset, P_1))\}$ ;  $G_2$  的特征集概念为  $c_3=(\{v_5\}, \{p_5, p_6\})$  和  $c_4=(\{v_6\}, \{p_1, p_2\})$ , 相应的 CSL 为  $L_2=\{((\emptyset, \emptyset), c_3), ((\emptyset, \emptyset), c_4), (c_3, (\emptyset, P_2)), (c_4, (\emptyset, P_2))\}$ , 其中  $P_1=\{p_1, p_2, p_3, p_4\}$ ,  $P_2=\{p_1, p_2, p_5, p_6\}$ 。

设  $X=(S, P, R)$  为由  $G_i$  定义的 FC, 则该语义图的 CSL 概要大小为  $|S|+2$ , 其中 2 是添加的最大元和最小元。当  $|S|$  较大时, 构造 CSL 所花时间和空间复杂度较高。因此, 针对整个语义图, 构造 CSL 时间复杂度相当高。本体分割后使得  $|S_i|$  远小于  $|S|$ , 从而大幅度提升了 CSL 的构造效率。

### 3 算法分析

#### 3.1 算法步骤

算法1描述了基于本体分割方法计算语义图  $G$  的 CSL 概要的主要步骤。

**算法1** 计算语义图的 CSL 概要

输入 SG //语义图文件, 格式可为 rdf, ttl, 或者 nt

输出  $L_i(i=1, 2, \dots, s)$  //  $L_i$  是第  $i$  个扩展子图的 CSL 概要

```

1. G=parse(SG)
//G 是三元组构成的有向图
2. Gi=DiGraph()
//Gi 是实例图, 初始化其为有向图
3. for (s, p, o) in G:
4. if p != RDF.type:
5. Gi.add_edge(s, o, label=p) //生成实例图 Gi
6. DepGi=Di.Graph()
//DepGi 是依赖图
7. for u in Gi.nodes:
8. w=Gi.degree(u)
//w 是节点 u 的总度数
9. for v in Gi.neighbors(u)
//v 是 u 的邻接节点
10. DepGi.add_edge(u, v, weight=1/w) //生成依赖图的
//边及权重
11.  $\pi=\{V_1, V_2, \dots, V_s\} \leftarrow \text{Partition}(\text{DepG}_i, \text{LineIsland})$ 
//利用 Pajek 的划分功能, 分割 DepGi
12. for Vi in  $\pi$ :
13. ExGi=DiGraph() //ExGi 是 Vi 的扩展子图
14. for u in Vi:
15. if Gi.out_degree(u)==0:

```

```

16. ExGi.add_node(u)

```

```

17. else:

```

```

18. for v in Gi.neighbors(u):

```

```

19. ExGi.add_edge(u, v)

```

```

20. for i=1 to s:

```

```

21. Li ← ComputeLattice(ExGi) //计算扩展子图的 CS 概要

```

```

22. Return Li (i=1, 2, ..., s)

```

算法1的第1步对输入的语义图文件进行解析, 可使用 RDFLib 库的 parse 函数完成解析, 其结果  $G$  为三元组构成的有向图。第2~5步是从  $G$  中提取实例图  $G_i$ 。其中, 第2步初始化  $G_i$  为有向图, 第3~5步中, 对每个三元组  $(s, p, o)$ , 若属性  $p$  不是 rdf:type, 则表明该三元组是实例之间的关系, 故加入到  $G_i$  中。第6~10步生成  $G_i$  的依赖图  $G_i^D$ 。第6步初始化  $G_i^D$  为有向图。第7~10步是对于  $G_i$  中的每个节点  $u$ , 对其所有邻接的节点  $v$ , 增加边  $(u, v)$  到依赖图  $G_i^D$  中, 且相应边的权重  $w_e=1/w$ , 其中  $w$  为  $u$  的出度。第11步根据依赖图  $G_i^D$  对节点进行划分, 得到划分  $\pi$ , 由  $s$  个分块  $\{V_1, V_2, \dots, V_s\}$  构成。Partition 函数使用了 Pajek 的“划分”功能, 其中划分选择方法为线岛。第12~19步, 根据节点划分结果生成每个划分块  $V_i$  的扩展子图  $ExG_i$ 。对于  $V_i$  中的每个节点  $u$ , 若  $u$  的出度为 0, 则将  $u$  加入  $ExG_i$  中; 若  $u$  的出度不为 0, 则对  $u$  的所有邻接节点  $v$ , 将边  $(u, v)$  加入  $ExG_i$  中。第20~21步, 计算每个扩展子图的 CSL 概要  $L_i$ , ComputeLattice 函数的任务是生成特征集概念形成的偏序格。本文采用文献[13]的算法实现该偏序格的计算。

#### 3.2 算法时间复杂度分析及比较

##### 3.2.1 算法1时间复杂度

算法1主要分为5个任务: 1) 提取语义图的实例图  $G_i$ ; 2) 生成  $G_i$  的依赖图  $G_i^D$ ; 3) 生成节点划分  $\pi$ ; 4) 计算所有划分块的扩展子图  $ExG_i$ ; 5) 计算每个  $ExG_i$  的 CSL 概要  $L_i$ 。

任务1的时间复杂度为  $O(|P|)$ , 其中  $|P|$  为语义图属性集合的基数。任务2的时间复杂度为  $O(|V_i| \cdot \Delta)$ , 其中  $|V_i|$  是实例图  $G_i$  的节点集合基数,  $\Delta$  为  $G_i$  的节点最大度。任务3使用 Pajek 工具完成划分功能, 其所有算法的时间复杂度最高为  $O(n\sqrt{n})$ , 其中  $n$  为图的节点数。因此, 利用 Pajek 生成实例图  $G_i$  的节点集合划分  $\pi=\{V_1, V_2, \dots, V_s\}$ , 时间复杂度最高为  $O(|V_i| \sqrt{|V_i|})$ 。任务4的时间复杂度为  $O(|V_i| \cdot \Delta^+)$ , 其中  $\Delta^+$  是  $G_i$  节点的最大出度。任务5的时间复杂度为  $O(|P| \cdot |C_m|^2)$ , 其中  $|C_m|=\max\{|C_1|, |C_2|, \dots, |C_s|\}$  是所有扩展子图  $G_i^E$  的特征集概念数最大值。在一般情况下,  $|P|$  和  $\Delta$  均远小于  $|V_i|$ , 且  $|C_i| \leq |V_i|$ 。综上所述, 算法1的时间复杂度为  $O(|P| \cdot |C_m|^2)$ 。

### 3.2.2 算法时间复杂度对比分析

为了与语义图概要类似方法和标准方法进行对比,本文基于FCA的概要<sup>[13]</sup>和基于语义图节点等价划分的概要方法<sup>[16]</sup>(基于熵图的概要)作为算法时间复杂度比较对象。

基于FCA的概要方法时间复杂度为 $O(N^3)$ ,其中: $N=\max\{|S|,|P|\}$ ;  $S$ 为语义图中RDF三元组主语节点集合;  $P$ 为属性集合。基于熵图的概要方法通过确定节点间等价关系,进而对节点集合进行划分形成概要节点,并计算概要节点间的属性边,从而生成概要图。该方法时间复杂度为 $O(|V| \cdot (|V|+|P|^2))$ 。

从3.2.1节可知,本文所提算法的复杂度为 $O(|P| \cdot |C_m|^2)$ 。经过本体分割后,  $|C_m|$ 的数值控制在较小的范围,对于大型语义图,  $|C_m|$ 的值远小于 $|V|$ 、 $|V|$ 和 $|P|$ ,因此相比上述两种概要方法,算法1具有较高的效率和可扩展性。

### 3.2.3 语义图数据结构敏感性分析

基于FCA的概要方法和基于熵图的概要方法对语义图数据规范性敏感,而算法1对语义图数据规范性不敏感。

基于FCA的概要方法对数据规范性敏感,当数据规范时,即 $|C|=1$ ,其概要图只包含一个概要节点,时间复杂度为 $O(|V|)$ ;当数据完全不规范时,即 $|C|=\min\{|V|,|\rho(P_i)|\}$ ,概要图包含 $|C|$ 个概要节点,时间复杂度为 $O(N^3)$  ( $N=\max\{|V|,|P|\}$ )。

基于熵图的概要方法对数据规范性敏感,当数据规范时,即 $|C|=1$ ,则所有实例节点都是等价的,概要图只有一个概要节点,时间复杂度是 $O(|E|)$ ,其中 $E$ 为语义图属性边的集合;当数据完全不规范时,即 $|C|=\min\{|V|,|\rho(P_i)|\}$ ,则每个实例为独立的概要节点,时间复杂度达到最大,即 $O(|V| \cdot (|V|+|P|^2))$ 。

算法1的时间复杂度取决于 $|P|$ 和 $|C_m|$ ,经过本体分割步骤,  $|C_m|$ 控制在合理范围,因此算法1对数据的规范程度不敏感。

## 4 实验与结果分析

### 4.1 数据集

本文使用的数据集如表1所示,包括南安普顿大学提供的LOD<sup>[26]</sup>数据集(数据集I)和Berlin SPARQL Benchmark (BSBM)<sup>[27-28]</sup>数据集(数据集II)。数据集I包含南安普顿大学相关数据,例如校历、教学楼信息、校内餐饮、停车场、公开课视频、在线培训课程等与该校相关信息。数据以RDF/XML、Turtle、JSON格式发布。数据集II是电商产品数据,包含商家、产品、用品评价等信息,是Turtle格式。该数据集由BSBM提供的程序自动生成,可根据需要

生成不同规模的数据集。本文所用BSBM数据集分别为BSBMData 4M(4 017 700个三元组)和BSBMData 10M(10 044 250个三元组)。

表1 数据集参数信息

Table 1 Datasets parameters information

数据集	来源	三元组数	节点数	属性数
数据集I	南安普顿大学 LOD	88 020	37 248	829
数据集II	BSBM	4 017 700	1 419 700	39
		10 044 250	3 547 500	39

### 4.2 实验1

实验1的目的是验证在计算CSL概要时,基于分割再计算与不分割直接计算的效率。实验1使用数据集I,采用Python语言,单机环境Windows10 64位专业版, CPU Intel酷睿 i7-3740QM 2.7 GHz,内存32 GB。

#### 4.2.1 基于本体分割的CSL概要

本文使用本体分割算法,并调用Pajek对语义图分割为30个分块,预处理及分割过程耗时0.49 s。在数据集I语义图的分割结果如图4所示。从图4可以看出,最小的分块包含107个节点,最大的分块包含3 365个节点。在数据集I上语义子图的CSL概要如图5所示,其包含的特征集概念数(6~87),以及特征集之间的关系,即CSL中哈斯图的边(简称特征边)为(1~178)。

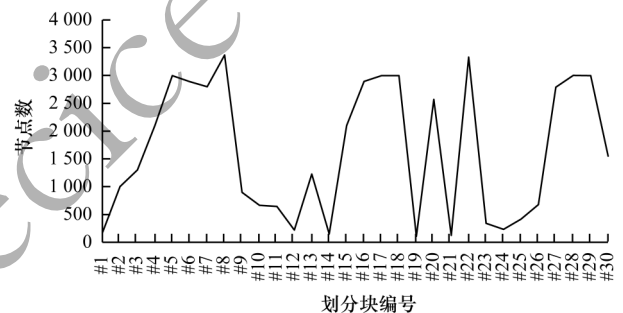


图4 在数据集I语义图的分割结果

Fig.4 Segmentation results of semantic graph on the dataset I

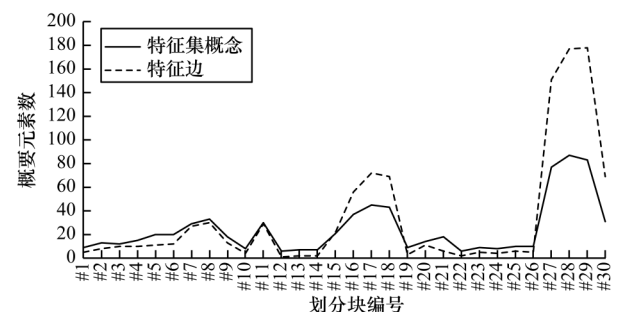


图5 在数据集I语义子图的CSL概要

Fig.5 CSL summaries of semantic sub-graphs on the dataset I



4.2.2 直接生成的 CSL 概要

在数据集 I 直接生成语义图的 CSL 概要,其概要结果包含特征集概念 371 个,特征边共有 620 条,程序耗时 17.05 s。

两种方法的结果对比如图 6 所示。基于本体分割的 CSL 方法总耗时为 10.36 s,是分割耗时(0.49 s)与分块计算 CSL 耗时(9.87 s)之和。直接生成 CSL 方法的耗时 17.05 s。从时间复杂度分析,基于本体分割的 CSL 方法明显优于直接生成的 CSL 方法,时间效率提升了 39%。由于本体分割可以离线进行,如果不包括本体分割耗时,本文算法的耗时节省了 42%。从生成的特征集概念和特征边分析,基于本体分割的 CSL 概要生成了更多的特征集概念和特征边。因此,基于本体分割的 CSL 方法明显提升了 CSL 概要的效率。

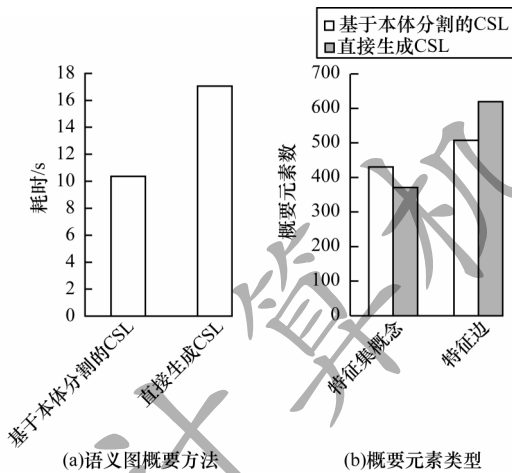


图 6 两种概要方法的结果对比  
Fig.6 Results comparison between two summarization methods

4.3 实验 2

将本文方法与其他代表性的语义图概要方法进行比较,以验证本文方法的有效性。本文选取基于熵图的概要<sup>[16]</sup>作为比较对象,其原因是利用节点等价关系进行概要生成熵图,且该方法提供了相应的工具,验证数据集为公开的 BSBM 数据集。实验 2 的环境与实验 1 相同。在不同数据集,两种方法的耗时情况如图 7 所示。两种概要情况对比如表 2 所示。在 BSBMData 4M 数据集,基于本体的分割方法和基于熵图的方法分别耗时 15 s 和 21 s,前者效率提升了 29%;对于 BSBMData10M 数据集,基于本体的分割方法和基于熵图的方法分别耗时 42.5 s 和 53 s,前者效率提升了 20%。在时间耗时方法,基于本体分割的方法要优于基于熵图的概要方法,两组数据效率平均提升 25%。说明先对大规模数据集进行划分后再运算能有效提高概要方法的效率。

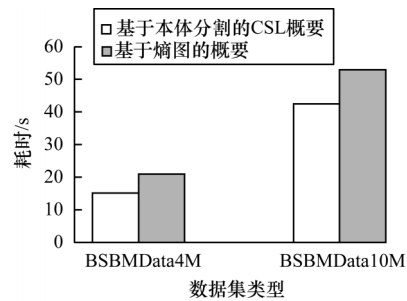


图 7 在两种数据集上两种概要方法的耗时对比  
Fig.7 Consumption results comparison between two summarization methods on two kinds of data sets

表 2 两种概要方法参数对比  
Table 2 Parameters comparison of two kinds of summarization methods

数据集 II	基于本体分割的 CSL 概要			基于熵图的概要	
	划分数	特征集概念均值	特征边均值	概要图节点数	概要图边数
BSBMData4M	10	27	41	22	63
BSBMData10M	15	33	49	25	79

从表 2 可以看出,在两个不同规模的数据集,基于熵图的概要方法能大幅缩减语义图节点数量。在总体概要结果上,基于熵图的概要方法得到了更小的概要图;在每个划分的子图概要情况上,两种方法得到了数量接近的概要节点。由于 BSBM 数据集由程序自动生成电商产品数据规范性高<sup>[28]</sup>,每个产品(实例)基本都包含商家、产品、用品评价等信息(不同的属性仅有 39 个),两个数据集规模不同,但结构相同。由 3.2.3 节算法对语义数据规范性敏感分析可知,语义图中大量的节点是等价的,且 BSBMData 4M 和 BSBMData10M 规模相差虽大,由于数据集的规范性导致概要节点数量差距不大。而本文方法对于数据是否结构规范不敏感,且经过分割步骤后,概要图规模能够控制在合理范围。

5 结束语

本文提出一种基于本体分割的 SGS 方法,将大型的语义图划分为多个子图,每个子图生成特征集格 CSL 概要。实验结果表明,从算法时间复杂度分析,相比基于 FCA 的概要和基于熵图的概要方法,本文方法对语义数据的规范程度不敏感,具有较好的可扩展性。在 LOD 数据集和 BSBM 数据集上,本文方法能有效提高概要方法的效率。下一步将对并行计算各语义子图 CSL 概要的算法进行研究,以进一步提升算法效率。

## 参考文献

- [1] 漆桂林, 欧阳丹彤, 李涓子. 本体工程与知识图谱专题前言[J]. 软件学报, 2018, 29(10): 2897-2898.
- QI G L, OUYANG D T, LI J Z. Preface to the topic of ontology engineering and knowledge graph[J]. Journal of Software, 2018, 29(10): 2897-2898. (in Chinese)
- [2] 王昊奋, 丁军, 胡芳槐, 等. 大规模企业级知识图谱实践综述[J]. 计算机工程, 2020, 46(7): 1-13.
- WANG H F, DING J, HU F H, et al. Survey on large scale enterprise-level knowledge graph practices[J]. Computer Engineering, 2020, 46(7): 1-13. (in Chinese)
- [3] MCCRAE J P. Linked data cloud [EB/OL]. [2020-06-20]. <https://www.lod-cloud.net/>.
- [4] THE FACEBOOK GROUP. The open graph protocol [EB/OL]. [2020-06-18]. <https://ogp.me/>.
- [5] GOOGLE. Google knowledge graph [EB/OL]. [2020-06-17]. <https://developers.google.cn/knowledge-graph/>.
- [6] DBPEDIA ASSOCIATION. Dbpedia [EB/OL]. [2020-06-25]. <https://wiki.dbpedia.org/>.
- [7] GITHUB. Wikidata [EB/OL]. [2020-06-17]. <https://github.com/Wikidata/Wikidata-Toolkit>.
- [8] ECONTENTPLUS PROGRAMME. mEducator [EB/OL]. [2020-06-20]. <http://www.meducator.net/>.
- [9] KNOWLEDGE MEDIA INSTITUTE. Open University [EB/OL]. [2020-06-25]. <http://data.open.ac.uk/>.
- [10] The Society for Learning Analytics Research (SoLAR). LAK Dataset [EB/OL]. [2020-06-21]. <https://solaresearch.org/initiatives/dataset/>.
- [11] ČEBIRIĆ Š, GOASDOUÉ F, KONDYLAŠIS H, et al. Summarizing semantic graphs: a survey[J]. The VLDB Journal, 2019, 28: 295-327.
- [12] ALAM M, NAPOLI A. An approach towards classifying and navigating RDF data based on pattern structures[C]//Proceedings of International Conference on Formal Concept Analysis. Berlin, Germany: Springer, 2015: 33-48.
- [13] GONZÁLEZ L, HOGAN A. Modelling dynamics in semantic web knowledge graphs with formal concept analysis[C]//Proceedings of World Wide Web Conference. Berlin, Germany: Springer, 2018: 1175-1184.
- [14] REYNAUD J, ALAM M, TOUSSAINT Y, et al. A proposal for classifying the content of the Web of data based on FCA and pattern structures [C]//Proceedings of International Symposium on Methodologies for Intelligent Systems. Berlin, Germany: Springer, 2017: 684-694.
- [15] FERRE S. A proposal for extending formal concept analysis to knowledge graphs [C]//Proceedings of International Conference on Formal Concept Analysis. Berlin, Germany: Springer, 2015: 271-286.
- [16] ČEBIRIĆ Š, GOASDOUÉ F, MANOLESCU M. Query-oriented summarization of RDF graphs [C]//Proceedings of the VLDB Endowment. New York, USA: ACM Press, 2015, 8(12): 2012-2015.
- [17] RIONDATO M, GARCIASORIANO D, BONCHI F, et al. Graph summarization with quality guarantees [J]. Data Mining and Knowledge Discovery, 2017, 31(2): 314-349.
- [18] PAPPAS A, TROULLINO G, ROUSSAKIS G, et al. Exploring importance measures for summarizing RDF/S KBs [C]//Proceedings of European Semantic Web Conference. Berlin, Germany: Springer, 2017: 387-403.
- [19] SONG Q, WU Y, LIN P, et al. Mining summaries for knowledge graph search [J]. IEEE Transactions on Knowledge and Data Engineering, 2018, 30(10): 1887-1900.
- [20] 赖雅, 王润梅, 徐德智. 基于参考点的大规模本体分块与映射[J]. 计算机应用研究, 2013, 30(2): 469-471.
- LAI Y, WANG R M, XU D Z. Anchor-based large-scale ontology partitioning and mapping [J]. Application Research of Computers, 2013, 30(2): 469-471. (in Chinese)
- [21] CUI Y, QIAO L, QIE Y. Ontology management and ontology reuse in web environment [C]//Proceedings of the Monterey Workshop. Berlin, Germany: Springer, 2016: 1-10.
- [22] KHAN Z C, KEET C M. Dependencies between modularity metrics towards improved modules [C]//Proceedings of European Workshop on Knowledge Acquisition, Modeling and Management. Berlin, Germany: Springer, 2016: 400-415.
- [23] SUÁREZ-FIGUEROA M C, GÓMEZ-PÉREZ A, MOTTA E, et al. Ontology engineering in a networked world [M]. Berlin, Germany: Springer, 2012.
- [24] MRVAR A, BATAGELJ V. Analysis and visualization of large networks with program package Pajek [J]. Complex Adaptive Systems Model, 2016, 4: 2-8.
- [25] GOLDENBERG J, HAN S, LEHMANN D R, et al. The role of hubs in the adoption process [J]. Journal of Marketing, 2009, 73(2): 1-13.
- [26] University of Southampton. Open data service [EB/OL]. [2020-06-25]. <http://data.southampton.ac.uk/>.
- [27] BIZER C, SCHULTZ A. Berlin SPARQL Benchmark (BSBM) [EB/OL]. [2020-06-20]. <http://wifo5-03.informatik.uni-mannheim.de/bizer/berlinsparqlbenchmark/>.
- [28] BIZER C, SCHULTZ A. The Berlin SPARQL Benchmark [J]. International Journal on Semantic Web and Information Systems, 2009, 5(2): 1-24.

编辑 薛晋栋