《智能信息处理》课程作业

基于形式概念分析的本体构建

曹芮

作业	分数[20]
得分	

2021 年 11 月 25 日

基于形式概念分析的本体构建

曹芮

(大连海事大学 计算机技术，大连 116026)

摘 要 针对传统本体构建方法依靠人工费时费力、主观干扰较大、对隐含概念和关系提取不足等问题,提出基于形式概念分析构建本体的方法。根据本体构建数据源的结构化程度,将这些构建方法分为 3 类,即基于结构化资源、基于非结构化资源和异构资源的合并本体构建方法。针对这 3 种类别,分析和阐述代表性的本体构建方法的优缺点,在比较结果中发现基于形式概念分析构建本体具有较大的改进空间,结合具体应用领域构建时需要在对象和属性的取舍、针对不同语言特点构建形式背景等问题上作进一步研究。

关键词：形式概念分析；本体构建

Ontology construction based on formal concept analysis

Rui Cao

(Computer technology, Dalian maritime university, Dalian 116026, China)

Abstract To solve these problems during traditional ontology construction like relying on artificial,wasting time and energy,subjective interference,lack of latent concept extraction and so on,researchers propose methods of ontology construction based on Formal Concept Analysis(FCA). According to the structure level of data resource using in ontology construction,these methods are divided into three classes: methods based on structured resource,methods based on unstructured resource,combine ontology construction based on different structure resource. This paper analyses a series of representational methods of these three classes respectively about their advantages and disadvantages,and finds that there is big improvement space on the choice betw een objects and attributes as well as making context aiming at different language combined with the constrction of the specific application domain.

Keywords formal conceptual analysis; ontology building

1 概述

随着语义 Web,[1]和信息大爆炸的到来,大规模抽取并表示信息的系统研究变得越发重要。近年来,本体学习逐渐为研究人员熟知,原因是获取信息较为简单且能提供可共享的高级结构。此外,由于本体能够概念化地描述事物的特征并在它们之间建立逻辑关系，这种结构化的可共享信息被广泛应用，目前主要集中在信息检索、人工智能、信息抽取、异构信息系统集成、语义 Web 等领域。但是,作为一种较为抽象的概念表达方式,本体在具体应用中受到一些挑战:本体在描述庞大的信息并对其概念化时难度较大;随着本体应用领域实体的多样化,本体描述语言相应也变得更需要具有兼容性。

研究人员提供了许多经典本体构建方法,如 Tove 法、Idef-5 方法、Kactus 工程

法、Methontology、Sensus 法、骨架法、七步法等。这些方法都有自己的特点和适用领域,再加上本体构建本身也没有统--标准,因此难以在不同领域本体的构建中保持一致'。客观上,本体构建是- -件复杂且费时的过程。而对领域专家来说,从给定的数据和文本中发现本体十分困难,需要一种能够半自动获取本体的方法,降低本体构建的复杂度和成本。

观察到本体和形式概念分析(Formal ConceptAnalysis, FCA)都是对概念的形式化表达,并且其表现形式都是概念和关系组成的层级结构,所以基于 FCA 构建本体具有可行性,并且具有以下特点:(1)概念格算法的研究已经较为成熟,在基于 FCA 构建本体的过程中,原本依赖人工的初始本体构造可以转化为概念格构造,实现了本体构建的半自动化; (2)概念格中的概念是算法自动从形式背景中获取,并按照序关系形成格结构,避免了传统本体构建中人工主观因素的干扰; (3)FCA 同时关注对象和属性,而本体只注重属性,将FCA引入本体构建,丰富了本体概念关系提取方法,发现更多隐含概念关系;(4)本体在视觉上像“树”,而概念格则像“网”,树中的节点非此即彼,网中的节点四通八达,通过使用概念格表示本体,可以使本体更像- -张 “网”,增加节点知识的互联性。

本文根据数据源的结构化程度,将基于 FCA 的本体构建方法分为 3 类: ()基于结构化资源进行本体构建; Q)基于非结构化资源进行本体构建;B)将结构化和非结构化资源合并进行本体构建。其中,结构化资源主要包括关系数据库或主题词表;非结构化资源是指没有固定结构的数据,例如纯文本,在使用这类资源构建本体时,必须先对文本资源进行自然语言处理(Nature Language Processing,NLP),去除冗余信息, 并且最大限度地保留用户感兴趣的内容,以使得机器理解文本并从中获取知识,使构建好的领域本体实现对领域概念和领域关系的高度覆盖。

2 FCA 和本体中的概念

2.1 形式概念分析

形式概念分析理论是德国数学家 Wille 教授在 1982 年提出的+ ,用于概念的发现、排序和显示,并且在 1999 年 Ganter 对形式概念分析理论的早期成果作了总结 5)。文献[6]指出:FCA 不会像其他数据分析方法那样粗粒度减少给定的信息,并且能够包含所有数据细节。其在本体构建过程中的概念提取和关系提取(分类关系和非分类关系)部分的应用被许多学者研究。

形式概念分析, 也称为概念格, 其基本思想是基于对象与属性之间的关系, 根据这一关系来建立一种概念层次结构, 其中每个概念都是对象与属性的统一体。另外, 概念格通过Hasse图生动简洁地体现了这些概念之间的关系。设 U 与 V 是两个有限的非空集合, U 是对象的集合, V 是性质的集合, 对象与性质之间的关系用 R 来表示, R 是 U×V 的一个子集, (x, y)∈R 代表 x 具有性质 y, (U, V, R)称为一个形式背景。通过 R, 我们建立 x 与它所具有的所有性质之间的联系, x 具有的所有的性质集合 xR 为:

$$xR=\{Y\in V|xRy\}\leq V$$

同理, 所有 y∈Y 具有性质 y 的所有对象集合为:

$$Ry=\{x\in U|xRy\}\leq U$$

定义 1 设(U, V, R)是一个形式背景, 对于一个对象集合 X,它对应的性质集合为:

$$X^{*}=\{y\in V|\forall x\in X,(x,y)\in R\}=\{y\in V|X\subseteq Ry\}=\bigcap_{x\in X}xR$$

对于一个性质集合 Y, 它对应的对象集合为:

$$Y^{*}=\{x\in U|\forall y\in V,(x,y)\in R\}=\{x\in X|xR\subseteq Y\}=\bigcap_{y\in Y}Ry$$

即 X＊是 X 中元素所共同具有的最大性质集合, Y ＊是具有 Y 中所有性质的最大对象集合。

定义 2 令 L＝(U, V, R)是一个形式背景, 对于 X≤U, Y ≤V, 若 X＊=Y 且 Y＊=X, 则(X, Y)叫做 L 的一个形式概念,

其中 X=ez(X，Y)叫做这个形式概念的外延，y=in(X，Y)叫做这个形式概念的内涵。

定理 1 形式概念的并与交定义为：

$$\bigwedge_{t\in T}(X_t,Y_t)=(\bigcap_{t\in T}X_t,(\bigcup_{t\in T}Y_t)^{\cdot\cdot})$$

$$\bigvee_{t\in T}(X_t,Y_t)=((\bigcup_{t\in T}X_t)^{\cdot\cdot},\bigcap_{t\in T}Y_t)$$

其中，T 是一个指标集，任意一个(X，Y，)是一个形式概念。

注：有限个外延(内涵)的交还是一个外延(内涵)，但是外延(内涵)的并可能不是任何一个形式概念的外延(内涵)。

L 中的所有形式概念形成一个概念格，通常用

Hasse 图来表示一个概念格，Hasse 图的每一个节点表示一个形式概念，最上层的节点叫做顶层节点，其外延包含 U 中所有的对象；最底层的节点叫做根节点，其内涵包含 V 中所有的性质。形式概念分析的基本思想可以用以下的例子说明：

例 1 一个动物与自身习性之间的关系的形式背景可以用表 1 的表格来表示，图 1 是其对应的概念格。

Table 1 Formal context of some animals

表 1 动物习性形式背景

种类	a	b	c	d	e	f	g	h	i
1 leech	×	×					×		
2 bream	×	×					×	×	
3 frog	×	×	×				×	×	
4 dog	×		×				×	×	×
5 spike-weed	×	×		×		×			
6 reed	×	×	×	×		×			
7 beam	×		×	×	×				
8 maize	×		×	×		×			

图 1 中，Cl=(；a，b，c，d，e，f，g，h，i)，C2=(4；a，c，g，h，i)，C3=(3；a，b，c，g，h)，C4=(6；a，b，c，d，f)，C5=(7;a，c，d，e)，C6=(2，3；a，b，g，h)，C7=(3，4；a，c，g，h)，C8=(5，6；a，b，d，f)，C9=(3，6；a，b，c)，C10=(6，8；a，c，d，f)，C11=(1，2，3；a，b，g)，C12=(2，3，4；a，g，h)，C13=(5，6，8；a，d，f)，C14=(6，7，8；a，c，d)，C15=(1，2，3，4；a，g)，C16=(1，2，3，5，6；a，b)，C17=(3，4，6，7，8；a，c)， C18=(5，6，7；a，d)，C19=(1，2，3，4，5，6，7，8；a)

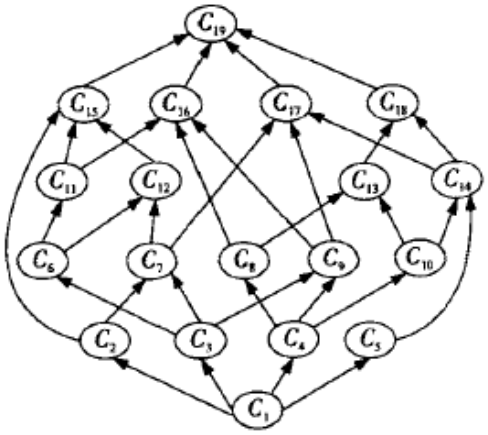


Figure 1 Concept lattice of Table 1

图 1 表 1 的概念格

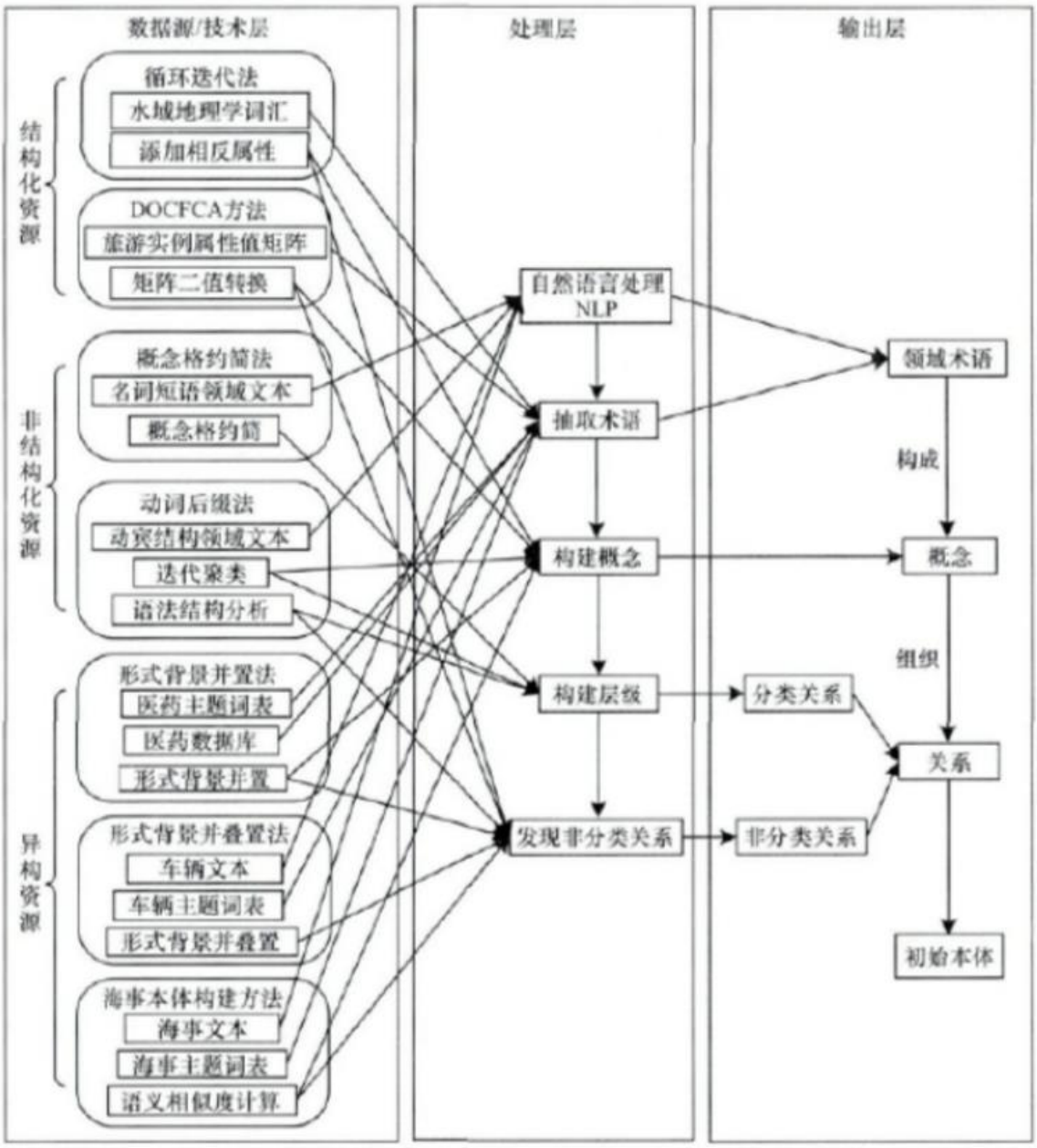
2.2 本体

Guber 于 1993 年给出了 Ontology 的定义”，本体是对概念模型明确的形式化说明，概念可以被理解为对世界或领域的抽象描述。文献[8]中总结了 Ontology 的 5 个基本建模元语。这些元语分别为:类(easses)，关系(relations)，函数(functions)，公理(axioms)和实例 Ginstances)，通常也把 elasses 写成 concepts;概念可以指任何事物;关系表示概念间的相互作用;函数是- -种特殊的关系，表示前 n-1 个元素唯一确定第 n 个元素;公理表示永真断言;实例表示元素”本体的结构可以表示为 0: = (C,≤C.R.8.≤R)。

3 基于 FCA 构建本体方法的三层结构

本文通过分析 7 种典型的基于 FCA 构建本体的方法.将基于 FCA 构建本体的方法体系划分为 3 个层次:数据源/技术层.处理层.输出层，并分别对 7 种方法中所使用的数据源和技术，在数据源/技术层-处理层以及处理层输出层之间建立联系，明确了基于 FCA 构建本体过程中的输人输出、因果关系以及构建流程，使原本离散的数据源、技术以及

相关处理之间有迹可循,有源可溯。基于 FCA 构建本体方法的层次如图 1 所示。.



基于 FCA 构建的 b 本体方法层次

4 基于结构化资源的本体构建方法

结构化资源作为-- 种本身就具有二维表结构的数据资源,在被用来构建本体时,省去了自然语言处理,可以简化构建形式背景的过程;其蕴含的分类信息可以为本体概念的提取提供参考;由于结构化资源间具有相似的结构,使得不同的结构化资源能够较为便利地合并(相对于非结构化资源和异构资源)。在基于结构化资源构建本体的基础上,分别提出了循环迭代本体构建方法和实例-属性-属性值矩阵本体构建方法。

4.1 循环迭代法

文献[10]认为,以传统分类学作为构建本体概念间关系的基础并按照包含关系来构造

概念间的层次,有以下 2 个弊端: (1)建立对象的层级体系时,一些对象仅按照分类学的序列来组织,但却不具有属性差异,这会在

题,必须用一种更优的方法描述概念和概念之间的
关系,而不是仅使用传统的分类学方法进行组织。
因此,在GACR项目中提出使用形式概念分析来构
造本体的方法,这种方法具有以下特点:
(1)概念由属性来描述;
(2)属性决定概念的层次,即层级体系不再仅由设计者定义;
(3)当不同概念具有相同属性时,认为这些概念等价;
(4)该方法可用于合作环境,多个设计者工作于一个本体,每个人都可对本体做改变,由管理者决定哪些改变被采纳。
具体步骤是:
(1)初始为空对象集合和空属性集合。
(2)向背景表中添加对象和属性。
(3)显示形式背景对应的概念格或其中的一部分。
(4) 用户可以在可视化的概念格的基础上做如下操作:

1)直接编辑(依据本体的实际需要)。添加或删除对象;添加或删除属性;从概念中添加或删除某一属性。

2)按照本体构建工具的提示编辑本体。当2个概念重合具有相同属性)时,要么将其合并成一个概念,要么通过给概念添加属性来加以区别(添加相反属性);FCA 能产生直接由属性构成的新概念,作为已有概念的父概念,但它们并不在背景表中显示。

3)重复整个过程,直到设计者满意为止。该方法是- -种分布式构建本体的手段,并且可以循环往复对本体进行完善,但其从无到有的本体构建机制使得这种方法不能有效地利用现有本体。

4.2 DOCFCA 方法

知识共享时带来问题; (2)一旦结构和位置已经被定义将很难再改变。为了解决这些问

2013 年提出了结合--种基于形式概念分析的领域本体半自动构建方法(Domain Ontology Con-struction based on FCA, DOCFCA)″,该方法的主要流程如图 3 所示。

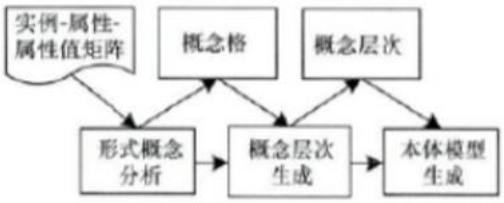


图 3 DOCFCA 流程

该方法依据概念格生成领域本体模型的主要思想是:

(1)去除底层节点,为每个概念命名。

2)根据概念格层次关系获取本体概念间的关系。

B)分别将概念格中的内涵和外延映射为本体中的概念属性和实例。.

4)扩充初始本体。

该方法与其他方法的不同之处在于,它增加了实例~属性-属性值矩阵向实例-属性二值形式的转变,拓宽了基于形式概念分析构建本体方法可用数据源范围。此外,该方法还将实例-属性属性值矩阵中具有相同属性的实例归并到同- -概念,避免概念冗余并且保证了领域本体最大可扩展性的构建原则。管形式概念分析在软件维护中有着广泛的成功应用,但其在应用领域深化、工具支撑等方面还需进行深入的研究.

5 结束语

目前基于FCA构建本体的方法研究星现出应用环境和技术手段的多样化,注重与当今信息网络环境发展趋势结合,并针对具体应用具体构建。如文献[21]借助云环境下的技术理念探索基于FCA的领域本体协作构建模式.提高本体构建的效率和质量:文献[22]

给出基于 FCA 和 Folk sonomy 的本体构建方法，为网络社区环境下通过社群分类法实现及时.灵活和人本的本体构建过程提供新的思路:文献[23]结合情报学领城本体构建实例说明 FCA 在本体构建中的应用:文献[24]提出了形式概念分析在基于非结构化资源的本体学习中的应用及其三维可视化展现形式。这些研究丰富了本体与其他相关技术的结合,拓宽了本体的应用领城。

本体的相关研究以本体构建为起点,逐步开始向外延伸,沿着“本体构建-本体合并-本体集成-本体对应”这-路径发展”。随着越来越多领域本体的成功构建,发现任何单一的本体都难以独自实现知识表达和知识复用，如何让这些具有重叠知识的本体相互关联，互相映射，而不是费时费力地重新构建更大的本体,需要对本体构建之后做进一-步研究。

本体合并是对同一领域的多个本体进行整合,从而构建该领城的统- .本体;本体集成则是将相关领域的不同本体融合在一起，重点对其差异进行互补;本体对应是指分布式本体协调,随着以上 2 种本体研

究构建出庞大的统--目标本体,并在当今网络环境下显示出低效和迟钝的缺点,如何在本体问建立通信和协调受到越来越多学者的关注。

本文将基于FCA构建本体的方法按照共资源结构化程度的不同分为 3 类.并找出本体构建过程中所使用的各种技术之间的相互影响和联系。通过分析,证实了基于 FCA 构建本体的方法具有主观影响小构建层级简单.知识互联性强等优点。但一些方法受其本身特性的局限并不适用于其他语言,并且在对异构资源构建本体时,FCA 表现出了操作的局限性。而本体构建的相关研究也在不断向外延伸，

如何解决基于FCA构建本体方法的不足,并将完善的方法广泛应用到本体合并、本体集成等相关领域是今后的研究方向。

参考文献：

[1]Ganter B，Wille R．Formal Concept Analysis：Mathematical Foundations
[M]．Springer-Verlag，1996.

[2]Birkhoff G.Lattice Theory[M].USA:American Mathematical Society，1940.

[3]Tilley T，Cole R，Becker P，eta1．A survey of formal concept analysis support for software engineering activities[A]．LNCS 3626：Formal Concept Analysis [C]．Berlin：Springer，2005．250—271.

[4]Li B，Sun XB，Leung H，Zhang S．A survey of code-based change impact analysis techniques【J】.Journal of Software Testing，Verification and Reliabiblity，2013，23(8)：613—646.

[5]ToneUa P．Using a concept lattice of decomposition slices for program understanding and impact analysis [JJ．IEEE Transactions on Software Engineering，2003，29(6)：495—509.