

基于 OWL 描述本体的语义信息抽取

梁晓涛, 谢荣传

(安徽大学 计算机科学与技术学院, 安徽 合肥 230039)

摘要:文中描述了一种基于 OWL 本体抽取可以语义 Web Agent 理解的语义数据的方法, 在抽取过程中先将 OWL 本体模型转换成 OSM 本体, 然后生成抽取规则, 并进一步完善本体, 最后将抽取出的数据加上语义标记, 转换成语义 Web Agent 可以接收的 RDF 格式, 从而可以促进语义 Web 的发展。

关键词:语义 Web; 资源描述框架; Web 本体语言; 面向对象系统模型; 数据抽取

中图分类号: TP311

文献标识码: A

文章编号: 1005-3751(2006)01-0062-04

Semantic Information Extraction Based on Ontology Described in OWL

LIANG Xiao-tao, XIE Rong-chuan

(Coll. of Computer Sci. and Tech., Anhui Univ., Hefei 230039, China)

Abstract: Describes a data extraction method based on ontology described in OWL, and the result can be understood by semantic web agent. In the process, first translate the OWL ontology model to OSM ontology model, and then generate extraction rules and improve the ontology. Finally, add semantic mark to the data, and translate the extracted data to RDF format which is accessible to semantic web agent. So the method can improve the development of semantic web.

Key words: semantic Web; RDF; OWL; OSM; data extraction

0 引言

随着 Internet 上信息的飞速增长, 一方面为查找所需信息提供可能, 另一方面也为海量非结构化数据中准确查找信息增加了难度, 因为目前 Web 上的信息是计算机可读的(readable), 而不是可理解的(understandable)。Tim Berners-Lee 提出了语义 Web 的思想^[1], 语义 Web 不是一种全新的 Web, 它是对现有 Web 的扩展, 其目标是通过智能的软件代理(Agent)完成异构和分布平台的互操作, 这就要求对 Web 信息添加一定的语义信息使 Agent 能够理解 Web 上的信息。当前阻碍语义 Web 应用的一个问题是缺少具有语义标记的数据源, 文中对如何抽取用户感兴趣的内容, 并且给这些内容加上语义标记做了说明。

1 系统结构

基于 OWL 的 RDF 信息抽取系统结构如图 1 所示。

图中首先从 OWL 定义的本体开始, 将其转变为面向对象系统模型(Object-oriented Systems Model), Web 页面中包含多条记录, 经过过滤器将其变成结构基本一致的

数据, 规则生成器通过将这些数据与 OSM 中的对象集进行匹配生成提取规则, 把提取出的数据存入关系数据库, 再将其格式化成语义 Web Agent 可以接受的 RDF 数据。

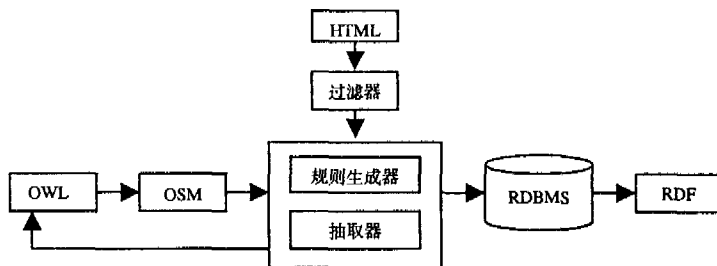


图1 系统结构图

2 OWL 本体

本体是共享概念模型的形式化规范说明, 它的目标是捕获相关领域的知识, 提供对该领域知识的共同理解, 确定该领域内共同认可的词汇, 并从不同层次的形式化模式上给出这些词汇(术语)和词汇之间相互关系的明确定义。

下面是用于描述语义信息的机制:

①RDF(Resource Description Framework)^[2]是一种专门用于表达 Web 上资源的语言, 它提供了一种用于表达信息, 并使其能够在应用程序间交换而不丧失语义的通用框架, 以 XML 语法为基础, 用 Web 标识符(uri/uris)标识资源, 用特定属性和属性值的陈述(statement)描述资源。RDF 定义了一个基本的数据模型, 包括下面 3 种对象类

收稿日期: 2005-04-10

作者简介: 梁晓涛(1981-), 男, 安徽涡阳人, 硕士研究生, 研究方向为数据库与 Web 技术; 谢荣传, 副教授, 研究方向为数据库、XML 数据管理、多媒体技术。

型:

* 资源(resources)。任何一个可以描述的事物都是资源,比如:网页,一本书等;

* 属性(properties)。用来描述资源的特定方面,特征、关系等;

* 陈述(statements)。资源加上特定属性以及属性值的集合,一个陈述有 3 个独立的组成部分:主体(subject)、谓词(predicate)、客体(object);

②RDF Schema 是描述资源中属性和类的词汇表,并带有资源和属性泛层次化的语义,如: `rdfs:subClassOf`, `rdfs:subPropertyOf` 用来描述类、属性之间的关系, `rdfs:domain`, `rdfs:range` 用来描述一个属性的定义域与值域。但是其表达能力不足,而且在逻辑推理方面有缺陷。

③OWL(Web Ontology Language)^[3]是 W3C 本体论工作小组提出的 Web 本体论语言规范,有 3 个子语言:OWL Lite,OWL DL,OWL Full,这 3 种语言的语义表达能力是递增的。其中 OWL Lite 除了具有 RDFS 特性外,还具有简单的属性约束能力,比如描述属性特性(传递性、对称性、互逆性等)以及对属性的基数进行约束;OWL DL 在 OWL Lite 的基础上引入了类型分割,它要求一个属性要么为对象属性(`owl:ObjectProperty`,表示两个类的实例之间的关系);要么为数据类型属性(`owl:DatatypeProperty`,表示类实例和 XML datatype 之间的关系),其语义描述能力相当于描述逻辑,能够保证推理系统最大限度地计算出所有结论;OWL Full 包含了所有的 OWL, RDFS 词汇,比如类之间的操作(交、并、补)等,此外它还允许在一个预定义的词汇表上添加词汇,能够提供最大限度的知识描述能力。用户可以根据需要具体选择某种语言。

3 OSM 本体

OSM(Object-Oriented Systems Model)是一种基于对象建模的概念模型语言,包含 3 个子模型:对象关系模型、对象行为模型和对象交互模型,既然人们只在乎数据的结构,所以仅使用对象关系模型。在 OSM 中,对象集是一组具有相同性质的对象,根据对象的类型可以分为词汇性质的(Lexical)和非词汇性质的(non-Lexical);关系集是表示两个或多个对象集之间的关系;参与约束表示对象参与某一关系的最大与最小数;一般/特殊(Generalization/Specialization)两个对象集之间用“:”隔开表示前者是后者的特殊,即“isa”。

图 2 是一个简单的 OSM 对象模型,其中 `Book[1:1] has Name[0:1]` 的意思是说一个 Book 准确包含一个 Name (最多是一个,最少也是一个),而一个 Name 最多只能是一个 Book 的 Name。

4 OWL 本体模型到 OSM 本体模型的转换

既然 OWL 是以属性为中心的语言^[4],那么就从 OWL 的属性及其约束开始进行转换,然后再到类。文献

[5]中描述了 DAML 模型到 OSM 模型的转换。

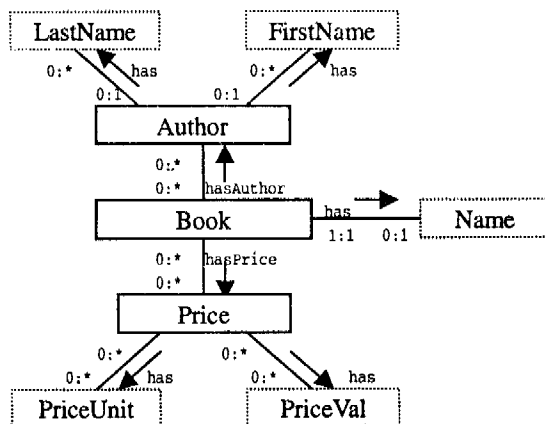


图 2 OSM 对象模型

4.1 OWL 属性的转换

OWL 定义的属性有两种基本类型:对象属性和数据类型属性,首先确定属性的类型。如果是对象属性类型,那么就要在属性的定义域和值域对应的对象集之间创建关系集,关系集的名称就取这个对象属性的名称;如果是数据类型属性,就创建一个词汇(Lexical)对象集和一个关系集,词汇对象集的名称就取这个属性的名称,关系的名称取“has”。如果 OWL 没有直接指明属性的类型,就要从属性的值域来推断到底属于哪种类型。如果一个属性是另一个属性的子属性(`owl:subPropertyOf`),而它本身只定义了定义域或值域,那么该属性的值域或定义域就可以通过父属性的值域或定义域来推断;如果一个属性为另一个属性的逆属性(`owl:inverseOf`)那么这个属性的定义域就为那个属性的值域,值域为那个属性的定义域;如果一个属性被被声明为对称的(`owl:SymmetricProperty`),则该属性的定义域与值域是相同的。

此外还可以通过约束来推断一个属性的定义域,如果一个属性有类型约束 `owl:allValuesFrom`,那么这个属性的值域就为 `allValuesFrom` 指定的那个类;若有 `owl:someValuesFrom` 约束,属性的值域也可以“简单”认为 `someValuesFrom` 指定的那个类。

可以在关系集上创建参与约束来处理基数约束,在关系集的定义域一边创建参与约束,对 `owl:minCardinality` 约束就创建最小参与约束;对 `owl:maxCardinality` 创建最大参与约束。

4.2 OWL 类的转换:一般/特殊化过程

首先为每个类添加对象集,然后指定这些类之间的关系。如果一个类 A 通过 `rdfs:subClassOf` 和另一个类 B 关联,那么就定义 A 为 B 的一个特殊,即: `A isa B`;如果一个类 A 通过 `rdfs:unionOf` 和另外一个类集相关联,那么就定义 A 为那个类集的一个一般,类集的每个成员都为 A 的一个特殊,并加上 `unionOf` 约束;如果一个类 A 通过 `owl:intersectionOf` 和另外一个类集相关,那么就定义 A 为那个类集的一个一般,类集的每个成员都为 A 的一个特殊,并加上 `intersectionOf` 约束。

5 生成抽取规则,将抽取出的数据存入数据库

因为进行数据抽取的数据源一般都是某一领域的一组数据,所以首先将包含待抽取信息的页面过滤,转化成结构基本一致的各项记录^[6],各记录项中的同一数据项的相对路径基本一致。这里采用的是机器学习中的覆盖算法学习抽取规则^[7]。如果各记录项中的数据项的相对位置有所变动,抽取路径就不止一个。

5.1 规则生成算法

输入: n 个同类样本页,每个样本页中包含 m 个数据项。

输出: 抽取规则 ExtractRules。

```

ExtractRules =  $\emptyset$ 
for i = 1 to m do
  Path[i] =  $\emptyset$ ;
  ExtractRules[i] =  $\emptyset$ ;
for i = 1 to m do
  for j = 1 to n do
    Path[i] = Path[i] + Path[i][j]
  将第 j 个样本页中的第 i 项与对象集比较,如果能够
  匹配(这里的匹配规则比较简单,为同义词匹配。复杂的匹
  配会涉及到相似度的定义、计算问题[8]),则把该样本页中
  的该记录值的路径添加到 Path[i] 中;如果不能匹配,则
  对本体模型进行修改,即添加必要的类和该类可能与其他
  类之间的关系,同时修改关系数据库的模式。
  }
  for i = 1 to m do
    while Path[i] !=  $\emptyset$  do
      调用一个 Path[i][j] 与 Path[i] 中的其他 Path[i][j] 比较获
      得被 path 覆盖的集合 S;
      Path[i] = Path[i] - S;
      ExtractRules[i] = ExtractRules[i] + path;
    }
  Return ExtractRules = ExtractRules[1] + ExtractRules[2] + ...
  + ExtractRules[m]
  }
  
```

若一个数据项的抽取路径有多个,提取时使用标识符的方法来定位数据项的位置,标识符就是能唯一标识该数据项的字符串或字符串组合,比如可以使用引导符(“:”前面的字符)作为标识符;若当前标识符标识的数据项的路径与抽取规则中的路径比较是否有相同的,以决定使用哪个抽取规则。引导符后面的数据就是该抽取规则所要抽取的内容,提取后生成相应的 SQL 语句插入到关系表中,当一个记录项的各个数据项完成抽取后从而转到下一个记录项。

5.2 关系数据库的创建

首先创建一个主表,其每一行都代表一个资源实例,然后将与该资源相关的属性添加进来,如果是一个对象属性,那么就创建一个对象标识符作为该属性的属性值,然

后为每个对象属性再类似上面创建一个表,类似上面将与该对象属性相关的属性添加进来,如果是一个数据类型属性就直接将其值插入到关系表中。如图 3 所示。

Book	Name	Author	Price
1001	TCP/IP Illustrated	2001	3001
1002	Data on the Web	2002	3002

Author	LastName	FirstName
2001	Stevens	W.
2002	Abiteboul	Serge

Price	PriceUnit	PriceVal
3001	dollar	5.95
3002	dollar	3.98

图 3 数据库中的关系表

6 RDF 数据的生成

6.1 RDF 数据的生成算法

```

generateRDFInstances( )
{ ontoURI = "http://somewhere/someontology#";
  someURI = "http://somewhere/books.html";
  for 主表中的每一行
    { table = 主表;
      instance = someURI + "#" + table 中第一列的值(一个
      OID);
      Resource R = createResource(instance);
      for table 中的每一列 A
        { //B 为外部本体定义的且与 A 相等的属性;
          Property p = createProperty(ontoURI, B);
          if (p 为数据类型属性) then
            { range = A 的值;
              R.addProperty(p, range);
            }
          else //即为一对象类型属性;
            { range = someURI + "#" + A 的值(一个 OID);
              Resource Rsu = createResource(range);
              R.addProperty(p, Rsu);
              R = Rsu;
              table = 与 table 通过 OID 相关联的表;
            }
          }
        }
      }
  }
  
```

6.2 生成 RDF 数据

```

<rdf:RDF
  xmlns:rdf = "http://www.w3.org/1999/02/22 - rdf - syntax
  - ns#"
  xmlns:ontoURI = "http://somewhere/someontology#">
  ....
  <rdf:Description rdf:about = "http://somewhere/book.html#
  Price3001">
    <ontoURI:PriceUnit>dollar</ontoURI:PriceUnit>
    <ontoURI:PriceVal>25.95</ontoURI:PriceVal>
  </rdf:Description>
  
```

```

<rdf:Description rdf:about="http://somewhere/book.html#
Book1001">
  <ontoURI:Name> TCP/IP Illustrated </ontoURI:Name>
  <ontoURI:hasPrice rdf:resource="http://somewhere/book.
html#Price3001"/>
</rdf:Description>
.....
</rdf:RDF>

```

6.3 生成 RDF 数据图模型

在图 4 中,椭圆表示的是一个资源,有向边表示的是该资源的一个属性,方框表示该属性的属性值,可以看出任何一个能够理解引入本体的 Agent 都能够可以在这些 RDF 数据中查找到具有某个特定属性值的实例,如可以查找 Stevens W. 写的所有书。如果一个出版社的 Agent 和一个书店的 Agent 都能够理解这个本体,那么书店和出版社之间就可以进行通信,当书店的某种书少于一定的数量时就可以从出版社订购,出版社也可以根据各个书店的销售情况掌握某种书的受欢迎程度。当然作为一个实验,这个本体还有很多不完善的地方,因为本体的创建过程需要领域专家的参与,而且创建过程本身就是一项非常麻烦、耗时的工作,所以怎样在多个本体之间实现互操作就成为以后的工作重点,文中作为一个实验只引入一个本体,不过重要的是使用本体抽取语义数据这种方法。

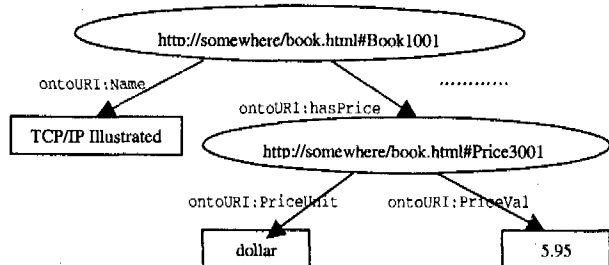


图 4 生成 RDF 数据图模型

7 结 论

传统的 Web 信息抽取是基于包装器的,包装器就是

一个网站网页的文法,包装器使用抽取规则描述数据在网页中的位置,只要数据在网页里面并且遵从包装器的格式,抽取器就可以抽取出数据来。基于包装器的数据抽取方法主要的缺陷是建立和维持包装器的工作量非常大,一旦用户要从别的网站上抽取数据,就要重新编写一个包装器。而基于本体的数据抽取可以跨不同网站,即使是领域发生变化也只需对本体进行修改。在数据抽取后为其添加语义标记,即转换成 RDF 格式,使其可以被语义 Web Agent 接受。

参考文献:

- [1] Berners-Lee T, Hendler J A, Lassila O. The semantic web [J]. Scientific American, 2001, 284(5): 34-43.
- [2] W3C Recommendation. RDF Primer[EB/OL]. <http://www.w3.org/TR/rdf-primer/>, 2004-02-10.
- [3] W3C Recommendation. Web Ontology Language Guide[EB/OL]. <http://www.w3.org/TR/owl-guide/>, 2004-02-10.
- [4] W3C Recommendation. Web Ontology Language Overview [EB/OL]. <http://www.w3.org/TR/owl-features/>, 2004-02-10.
- [5] Chartrand T. Ontology-based Extraction of RDF Data From the World Wide Web[EB/OL]. <http://www.deg.byu.edu/papers/>, 2003-03.
- [6] Embley D W, Jiang Y S, Ng Y. Record-boundary discovery in Web documents[A]. Proc. of the 1999 ACM SIGMOD Intl Conf. on Management of Data (SIGMOD'99) [D]. Philadelphia, Pennsylvania, USA: ACM Press, 1999. 467-478.
- [7] 李效东, 顾毓清. 基于 DOM 的 Web 信息提取[J]. 计算机学报, 2002, 25(5): 526-533.
- [8] Doan A, Madhavan J, Domingos P, et al. Learning to Match Ontologies on the Semantic Web[A]. In Proc. of the World-Wide Web Conf. (WWW2002)[C]. Honolulu, Hawaii, USA: ACM Press, 2002. 662-673.
- [3] Han J, Kamber M. Data Mining - Concepts and Techniques [M]. New York: Morgan Kaufmann, 2001.
- [4] 李雄飞, 李 军. 数据挖掘与知识发现[M]. 北京: 高等教育出版社, 2005.
- [5] 陈京民. 数据仓库与数据挖掘技术[M]. 北京: 电子工业出版社, 2002.
- [6] 万国华, 陈宇晓. 数据挖掘算法及其在股市技术分析中的应用[J]. 计算机应用, 2004, 24(11): 104-106.
- [7] 梁 循. 通过 Web 统计信息挖掘研究股市反应[J]. 微机发展, 2005, 15(8): 81-84.
- [8] Berson A, Smith S, Thearling K. 构建面向 CRM 的数据挖掘应用[M]. 郑 岩, 魏 黎译. 北京: 人民邮电出版社, 2003.

参考文献:

- [1] Dunham H. Data Mining - Introductory and Advanced Topics[M]. New Jersey: Prentice Hall, 2003.
- [2] Groth R. Data Mining - Building Competitive Advances[M]. New Jersey: Prentice Hall, 2000.

(上接第 4 页)

理利用消费者数据,也是当前实施数据挖掘面临的问题。

数据挖掘不会替代有经验的商业分析师或管理人员所起的作用,毕竟它只是提供一个强大的工具。数据挖掘工具要做的就是使这些模型得到的更容易、更方便,而且有根据。所以,数据挖掘和专业人员的关系是伙伴和朋友,而不是竞争对手。