rv001010 t50mn

001017400604010QI0g

0-01201i010 门 阳 国 4 0f01eá10Ov微10106600110再

**第9章** **语义网与知识图谱**

本章将介绍语义网的起源、语义网的结构、相关技术规范及其实例、语义网的应用与 挑战、知识图谱等内容。

**9.1** **语义网的基本概念**

传统的万维网是一个文档的网络 (Web of Documents)。

Web 已经成为人们获取信息的主要渠道，深刻地影响着人们的生活。人们在Web 上 搜索信息、浏览国内外新闻，甚至进行网上电子商务交易等。这个Web 是面向人的，也 就是检索信息需要用户的主动参与过程。面对海量的网页，准确、快速地获取有价值的信 息的难度是很大的。换句话说，目前的网页是设计给人看的，它们用自然语言加上简单的 标 记 (HTML 标记)表示，没有对所要表达的语义进行描述和标注，对于计算机来讲， 只能把网页看作一系列的字符。

1998年，互联网之父、 <HTTP> 协议和HTML 规范的发明人Tim Berners-Lee首次提 出了语义网的概念，拉开了语义网研究的序幕，他的文章发表在《科学美国人》杂 志上。

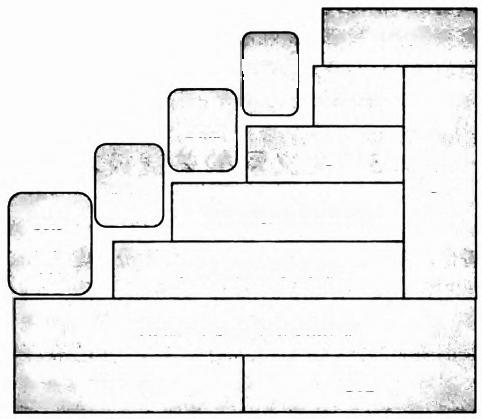
语义网 (Semantic Web) 是现有万维网的延伸与变革，它有一个重大的转变，即由 Web of Documents 向 Web of Data 转变。语义网的目标是让计算机理解数据的语义，从而 自动完成多种功能。

语义网将使用可以被计算机理解的方式描述事物，并且连接起来形成信息网络。在此 基础上，由于机器能够自动识别和理解万维网上内容的含义 (Meaning of Data), 它就能 够自动化收集、集成、处理来自不同数据源的数据(这里讲的处理包括统计分析、数据挖 掘、机器学习以及知识管理等),帮助用户准确、快速地获得他们所需的信息。语义网被 认为是下一代互联网即Web 3.0的发展方向。



**9.2** **语义网体系结构**

Tim Berners-Lee提出了语义网体系结构的最初设想，随着研究的深入，该体系结构 不断得到补充和发展。图9- 1给出了语义网的体系结构，这是 一种分层的架构，上层功 能依赖于下层功能，自下而上，每层的功能逐渐增强。下面我们了解各个层次所提供的 功能。

**Trust**

Rules

Proof

Data

Logic

Digital

Signat-

Data

Ontology Vocabulary

ure

Self- desc doc

RDF+RDF Schema

XML+NS+XML Schema

Unicode URI

**图** **9** **-** **1** **语义网体系结构**

( 1 ) 第 一 层 是 基 础 层 ， 主 要 包 括 Unicode 和 URI(Uniform Resource Identifier)。 Unicode 是一种流行的字符集，采用多字节的编码。任何自然语言的字符经过 Unicode 编 码都可以被机器容易地接受和处理。 URI 即通用资源标识符，是用于唯一标识抽象或物理 资源的简单字符串。网络上的任何资源包括 HTML 文档、程序、图片、音视频等，都可 以 用URI 进行编址，从而实现对资源的定位。

(2)第二层是语法层(句法层),核心内容是XML 以及相关规范。 XML 是 SGMI

(Standard Generalized Markup Language, 标准通用标记语言)的 一 个子集，通过自描述 的方式定义数据结构。它不仅描述数据的内容，而且体现数据之间的联系。

用户可以使用XML 自由定义元素名称 (Tag Name, 元素也称标记)以及元素的层次 关系，表达数据的结构和内容。为了方便其他用户以及计算机程序正确理解用户定义的内 容，人们使用XML 命名空间 (Name Space) 和 XML 模 式 (XML Schema) 约 束XML 文 档的结构。

(3)第三层是资源描述框架，包括RDF 及相关规范。 RDF 是用于描述 Web 上各种资 源的通用框架，包括网页的作者、创建日期、修改日期以及内容等。 RDF 本质上是 一 种 数据模型，它用主体 (Subject) 、 谓词或者属性 (Predicate/Property) 以及客体或者属性 值 (Object/Property Value) 构成的三元组来描述资源的元数据。由于 RDF 的 灵 活 性 ， 它成为知识表达的通用形式。如果我们把XML 看成一种标准的元数据语法规范， RDF 则



可以看作一种标准的元数据语义描述规范。

(4)第四层为本体 (Ontology) 层。它在RDF 的基础上定义了RDFS(RDF Schema) 和OWL(Ontology Web Language),使得用户可以创建具体应用领域的本体，也就是具 体领域的概念体系。RDFS 和 OWL 定义了语义信息，支持机器在使用RDFS 和 OWL 描 述的知识库和本体库中进行推理，获取新知识。

(5)第五层到第七层，分别是逻辑层 (Logic)、验证层 (Proof) 和信任层 (Trust)。

①逻辑层，在上述各层的基础上，进行逻辑推理。②验证层，对逻辑陈述进行验证，得出 结论。③信任层，是语义网安全机制的组成部分，它负责发布语义网所支持的信任评估。 目前验证层和信任层仍然处于设想阶段。

需要注意的是，随着研究的深入，上述语义Web 的体系结构将得到不断补充和发展。 为了把语义网的设想落到实处，科研人员和相关组织制定了相关的规范，研发了各种开发 工具和软件包。人们可以使用这些开发工具和软件，创建各个领域的语义网应用。

**9.3** **语义网的关键技术**

从图9-1可以看出，实现语义网的三大基础技术分别是 XML,RDF 和 Ontology。

其 中 ，XML 层作为语法层(句法层),RDF 层作为数据层，Ontology 层作为语义层。

**9.3.1 XML(Extensible Markup Language, 扩展标记语言)**

通用标记语言标准 (SGML) 是定义电子文件结构和进行内容描述的国际标准。 SGML 标准过于复杂，SGML 软件价格也非常昂贵，因此，在编写网页时，人们使用更为 简单的超文本标记语言 (Hypertext Markup Language,HTML) 。HTML比 SGML 简单 得多，但是缺少SGML 的灵活性和通用性，不能支持特定领域的标记语言，比如对数学、 化学、音乐等领域的数据表示支持较少。

XML 则结合了SGML 和 HTML 的优点，并且消除其缺点。 XML 是一种元标记语 言，元标记就是开发者可以根据自己的需要定义自己的标记。比如，为了对书籍进行管 理，用户可以定义<book><author><year><name> 等标记，并且用这些标记描述 具体的数据，比如：

|  |
| --- |
| <book>  <author>Qin Xiongpai,Chen Yueguo,Du Xiaoyong</author> <year>2017</year>  <name>Introduction to Data Science</name> </book> |

只要符合 XML 的命名规则，用户可以自行定义 XML 标记，为各种应用程序的数据 交换提供了方便。

XML 也是一种语义/结构化语言，它描述了文档的结构和语义，用户可以很方便地定 义自己领域的专用标记。使用XML 描述的数据，可以具有不同的展现形式，包括 WORD

格式、PDF 格式、HTML 网页格式等。

为了方便数据交换，文档的模式必须符合一定的规范，可以使用两种方式定义XML

文档的模式。(1)使用文档类型定义 (Document Type Definition,DTD),DTD定义了 XML 文档的基本结构，包括定义 XML 文档中出现的元素(标记)、这些元素出现的次 序/次数、它们如何嵌套以及XML 文档结构的其他详细约束信息。比如在上述实例中， <book> 标记下，必须嵌套出现各一次<auth or><year><name> 标记。 (2)使用

XML 模式 (XML Schema),XML模式可以定义DTD 能够定义的文档结构，还可以定义 数据类型，以及比DTD 更加复杂的规则。XML Schema使用XML 的语法定义XML 文档 的结构，易于理解。

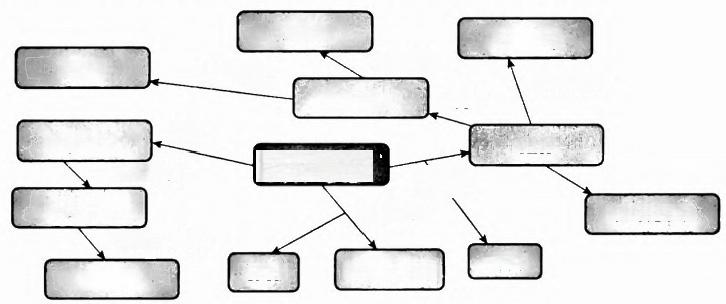
**9.3.2 RDF(Resource Description Framework, 资源描述框架)**

RDF 是由万维网联盟 (World Wide Web Consortium,W3C) 的资源描述框架工作组 于1999年提出的一个资源描述方案。该方案于2004年正式成为万维网联盟的推荐标准。 RDF 是一种语义资源描述语言，它是一种由数据结构、操作符、查询语言和完整性规则 组成的数据模型。

RDF 的基本数据模型包括资源 (Resource) 、 属性 (Property) 及陈述 (Statement)。

(1)资源： 一切能够使用RDF 表示的对象都称为资源，包括网络上的所有信息、虚拟概 念和现实事物等。资源用唯一的URI 来标识，不同的资源拥有不同的URI 。(2) 属性： 用来描述资源的特征或资源间的关系。每一个属性都有其意义，用于定义资源的属性值 (Property Value)、描述属性所属的资源形态、与其他属性或资源的关系。(3)陈述： 一 条陈述包含三个部分，通常称为RDF 三元组<主体，属性，客体>。其中主体是被描述 的资源，用URI 表示。客体表示主体在该属性上的取值，可以是另外一个资源(由 URI 表示)或者是文本。每个陈述是一个独立的事实。

RDF 三元组描述了实体具有什么属性以及实体间具有什么关系，于是我们可以用一 个图模型把这些关系都表达出来。比如图9-2,展示的是YAGO 知识库里，Max Planck 这个实体和其他实体及概念的关系。从图中我们了解到这样一些事实：Max Planck 于 1858年出生于Kiel,Kiel 是一个城市，这个城市位于德国境内等。



citta

**Stadt**

inGerman

**Pbysicist** type

subclassOf

**Scientist**

subclassOf

**Person**

Germany

inItalian

locatedln city type

**Kiel**

bormln hasWon

1858

**45°NO7E**

Nobel Prize

Yr

**1919**

**图** **9** **-** **2** **RDF 三元组与图模型**

**资料来源：**<http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/.>

**RDF 三元组是语义网数据表示的基础。要实现从目前的万维网到语义网的转变，构**

建海量的RDF 数据集是一项基础性工作。

**比如，DBpedia是一个通用知识库，它通过从维基百科** **(Wikipedia) 的词条里撷取结**

构化的信息而创建，描述这些信息的最自然的方式就是RDF。2014 版的DBpedia, 拥有超

**过458万的实体，包括144.5万个人、73.5万个地点、12.3万张唱片、8.7万部电影、** 1**.9万种电脑游戏、24.1个组织、25.1万种物种和6000种疾病。这些知识被BBC、路透** **社、《纽约时报》采用。Google,Yahoo 等搜寻引擎使用Dbpedia改善搜索结果。**

当用RDF 描述资源时，任何人可以定义用于描述的词汇，但是这些词汇的具体含义、 词汇之间的关系各不相同。显然，这不便于机器理解和处理数据。

为了解决这个问题，RDFS(RDF Schema) 定义了一组标准类及属性的层次关系词 汇，帮助用户构建轻量级的本体，即 RDF 是领域无关的，没有定义任何领域的语义，如 果用户要建立某个领域的本体和知识库，需要由用户借助RDFS 来完成。

RDFS 定义了某种模式，即定义了特定领域的词汇的含义。RDFS 描述类是通过资源 rdfs:Class 和 rdfs:Resource, 特性 rdfs:type 和 rdfs:subClass(f 来完成的。利用 rdfs:subClassOf 可以定义子类，形成层次结构。在RDFS 中，对类的属性的描述，是利 用RDFS 类 rdfs:Property 和 RDFS 特性 rdfs:domain (定义域)、rdfs:range (值域) 和 rdfs:subProperty(f 来进行声明和描述的。 rdfs:domain 表示哪些类有这个属性， rdfs:range限定属性的值域。比如，对于hadDriver属性来讲，Vehicle类拥有这个属性， 属性的取值只能是一个Person。

下面是用RDF 和 RDFS 来描述Web 资源的一个简单实例。

|  |
| --- |
| <rdf:RDF>  ≤rdf:Description about=’<http://iir.ruc.edu.cn/>～duyong/’> <nsruc:Name>Du Xiaoyong</nsruc:Name>  ≤nsruc:Position>Professor</nsruc:Position>  <nsruc:Works-for>Renmin University of China</nsruc:Works- for>  <nsruc:ChairOf>Database and Intelligent Information Re- trieval Lab  </nsruc:ChairOf>  </rdf:Description> <rdf:RDF> |

这是关于Du Xiaoyong教授的一些属性，包括Name,Position,Works-for,ChairOf

等属性，这些属性是由命名空间 nsruc指定的Schema 定义和解释的。其中，属性ChairOf 被约束为教授和实验室的一种关系，在一定的意义上明确了其语义。

|  |  |
| --- | --- |
| <rdf:Description  <rdfs:domain <rdfs:range  </rdf:Description> | ID='ChairOf’>  rdf:resource=’#Professor'/> rdf:resource=’#Lab'/> |

**9.3.3** **OWL** **与本体** **Ontology**

由 Tim Berners-Lee所提出的语义网架构的第4层为本体层，它是语义网技术里最关 键的部分。本体的概念最初起源于哲学领域，是“对世界上客观存在物的系统的描述”。

20世纪90年代初期，斯坦福大学计算机科学家 Tom Gruber 对于计算机科学术语 “Ontology”给出了自己的定义：本体是一种对于某一概念体系(概念化过程) (Concep- tualization) 的明确表述 (Specification)。对于特定的应用领域而言，本体表达的是其整 套术语、实体、对象、类、属性及其之间的关系，提供的是形式化的定义和公理，用来约 束对于这些术语的解释和运用。

一个本体描述了一个特定领域的一个形式化的、共享的概念化模型，其内容包括概 念、概念的同义关系、概念的上下位关系、概念的属性、属性的定义域 (Domain) 和值 域 (Range) 以及这些内容上的公理、约束等。

本体是领域相关的，因此难以制定一个标准的、通用的本体构建方法。在这个背景 下，本体工程学应运而生。本体工程学研究的内容包括面向领域的本体开发过程、本体生 命周期管理、本体构建方法及方法学、以及为这些任务提供支持的软件工具和语言。

本体一般需要采用本体语言来编制，本体语言是一种用于编制本体的形式化语言。目 前已经诞生了不少本体描述语言。

**1.OIL &DAML**

OlL(Ontology Interchange Language) 是由斯坦福大学、荷兰阿姆斯特丹大学等多 家机构从2000年开始联合开发的本体语言。 OIL 具有合并和表示本体以及进行系统间交 互的功能。用OIL 来描述本体，需要区分3个不同的层次。首先是对象级，这一级描述具 体的本体。其次是第1元级，这一级提供了确定的本体定义，用一种定义良好的语义来描 述结构化的词汇。最后是第2元级，这一级描述的是一个本体的元信息，如作者、名称、 主题等。

DAML(DARPA Agent Markup Language) 由 DARPA (美国国防部高级研究计划 署)主持开发。它与 OIL 一样建立在 RDF 之上，力图融入 RDF 和 (IL 等的优点。 DAML 以描述逻辑为基础。DAML 的主要目标是开发一个以机器可读的方式表示语义关 系.并与当前及未来技术相容的语言，尤其是要开发出一套工具与技术，使得代理 (A- gent) 程序可以识别与理解信息源，并在代理程序之间实现基于语义的互操作。DAML 扩 展了RDF, 增加了更多的更复杂的类、属性等定义。

DAML 的研究者和(IL 的研究者注意到对方技术的优点，他们开始合作，推出了 DAML+OIL 语言。该语言具备充分的表达能力(如唯一性、传递性、逆反性、等价等), 具有一定的推理能力.形成了语义网中知识表示语言的整体框架。目前支持DAML+(IL

的工具软件有 (ilEd,WebODE 等。

**2.0WL**

在众多的本体语言中，网络本体语言 (Web Ontology Language,OWL) 是广为接受 的一种语言，它提供了一种可用于描述网络文档应用中所涉及的类及其之间关系的语言。

OWL网络本体语言于2004年成为W3C 的推荐标准，属于编纂本体的知识表达语言 家族。为了适应不同的表达能力和计算效率的需要， OWL 提供了3种表达能力递增、计 算效率递减的子语言：(WL Lite,OWL DL,OWL Full 。(1)OWL Lite是 OWL DL的

一个子集，仅支持部分的OWL 和简单约束的用户。(2)OWL



语言要素。OWL Lite用于提供给那些只需一个分类层次 DL包括(WL Lite语言的所有成分，但有一定的约束，如

一个类不能同时是一个个体或者属性，一个属性不能同时是一个个体或者类等。OWL DL 适合于那些在拥有计算保证的前提下追求强大表达能力的用户使用。(3)OWL Full是 OWL 语言的全集，包含所有的(WL 语言要素，并拥有与RDF 一样的句法自由，它面向 那些需要RDF 的最大限度表达能力的用户。 OWL Full 允许引人本体来扩展预定义的 RDF/OWL 词汇的含义。同OWL DL 相 比 ，OWL Full对推理的支持难以预测。

在表达概念的语义灵活性、Web 内容的机器可理解性等方面，OWL 比早前的XML, RDF.RDF-S 等语言都要强。OWL 弥补了RDFS 的不足，运用人工智能 (Artificial In- telligence,AI) 中的逻辑来赋予语义，支持多种形式的推理。在OWL 之上，W3C 还定 义了规则互换格式 (Rule Interchange Format,RIF) 和语义网规则语言 (Semantic Web Rule Language,SWRL) 来辅助推理，其中RIF 支持在不同的规则 (Rule) 格式进行互操作。

**3.** **实际应用中的本体库**

有了本体描述语言，人们就可以开发各种各样的本体。本体库可以分为通用的本体库 和特定领域的本体库两种。

通用的本体库描述的是通用的概念及其关系，可以运用到各个领域。比如 YAGO (Yet Another Great(ntology) 就是一个通用的本体库(当然它也是一个知识库)。 YA- GO 从 Wikipedia,WordNet,GeoNames 等数据源抽取概念和实体来创建。目前 YAGO 的知识库包含超过1000万个实体(包括人物 (Person) 、 组 织 (Organization) 、 城 市 (City) 等),以及关于这些实体的1.2亿个事实 (Fact)。

特定领域的本体库，则面向某个特定领域的专用的概念和术语的描述，它的各个层级 的概念可以挂接到通用本体库的某个概念之下。比如，国际卫生术语标准组织 (Interna- tional Health Terminology Standards Organization) 负责维护 SNOMED(Systematized

Nomenclature of Medicine) 医学术语系统，目前SNOMED 包含30万个医学概念以及这 些概念之间的关系组成的层次语义网络 (Hierarchical Semantic Network)。

**9.4** **知识库与知识图谱**

**9.4.1 知** **识** **库** **与Linked Open Data**

本体是对概念、概念之间关系的形式化描述。比如，我们对“全国”“省/直辖市” “区/县/市”等概念以及它们的层次关系做了形式化说明，就构成了本体(库)。在此基础 上，我们把全国所有的省/直辖市、区/县/市等具体的行政区划，按照上述本体库的关系 约束，组织成实体(包含属性)以及实体之间关系的网络，这就是知识库。

可以这么认为，本体(库)是对知识库的结构进行描述，知识库则描述和保存具体的 知识，它保存了各种实体、实体的属性以及实体之间的关系。因此，知识库就是实体(实 体包含属性)以及实体之间关系的总和。需要注意的是，知识库不仅要保存具体的知识， 而且需要保存关于这些知识的描述即本体，也就是知识库包含本体和知识本身。



主要的知识库包括上文提到的 YAGO,DBpedia 等，还包括WordNet.Freebase 等。 YAG() 和 DBpedia已经在上文中介绍. WordNet 则在第7章“文本分析”中做过介绍。 这里我们详细介绍Freebase 知识库。

Freebase 是个类似于Wikipedia 的创作共享类网站，所有内容都由用户添加。两者之 间最大的不同之处在于Freebase中的条目都采用结构化数据的形式，Wikipedia 则是半结 构化或者无结构的网页。Freebase 的结构分为三层，分别是 Domain,Type 和 Topic 。可 以简单地认为，Domain 对应实际生活中的各个领域，比如艺术与娱乐领域等。Type 对应 某类实体，某类实体具有一套固定的属性 (Property) 。Topic 则对应某个具体的实体，比 如某个人物、某部电影。截至2007年5月30日. Freebase 中已经包含61个 Domain、 765个Type 、2312676 个 Topic。

到2015年， Freebase 规模则比2007年要大得多，它包含4000万个实体，超过6亿 条事实。所有内容都由用户添加.采用创意共用许可证，可以自由引用。2010年拥有 Freebase 的 Metaweb 被 Google 公司收购。此后.(oogle 基于 Freebase 研发了Knowl- edge Graph, 并宣称扩展到了5亿个实体和25亿条事实。2015年，Google 发布了Knowl- edge Graph API. 代替Freebase API 。2016年 5 月 .Google 关闭了Freebase.com。

虽然知识库的数量越来越多.但是每个知识库都难以包含所有人需要的知识，知识库 之间还包含冗余的信息，比如一些著名的历史人物，几乎所有的知识库都包含其信息。人 们希望这些知识库开放出来，供人们共享。开放共享的知识库(数据集)称为开放数据集 ((pen Data)。由于知识库之间的冗余性，需要解决一个问题，即把不同知识库的同一个 实体标识为一个唯一的实体。

可以通过Web 把这些开放的数据集(知识库)连接起来，形成一个巨大的知识库， 这就是Linked(Open Data项目 (<http://linkeddata.org/>) 的初衷。为了支持开放数据集 的连接，需要一些相关的技术，包括：(1) URI. 通过 URI 把现实世界中的实体和概念标 识出来，每个资源需要一个全局唯一的 URI 来进行标识。(2) <HTTP>, 对开放数据集里 的资源 (Resource) 以及关于资源的描述 (Description of Resource) 的提取，在 <HTTP> 协议之上实现。(3) RDF, 一般来讲，开放出来的数据集需要以 RDF 数据模型对实体、 实体的属性、实体的关系进行描述和存储。当用户查询某个资源时，以 RDF 格式返回相 关信息。比如，如果用户查询的URI 指向数据库里的一条记录，则需要以RDF 格式返回 该记录的所有属性值。

在这里，我们通过一个实例来描述数据集间的同一个实体如何建立联系。比如在 GeoNames 项目中的Auburn 的 URI, 等同于 DBpedia项目的Auburn 资源。此外，Free- base 项目中的Auburn 的 URI, 也等同于DBpedia项目的 Auburn 资源。我们使用 OWL 建立如下连接关系，表示它们指的是同一个实体。

#使用OWL 连接标识符

#Connecting the DBpedia resource for Auburn.CA to two other #resources using owl:sameAs

@prefix owl:<<http://www.w3.org/2002/07/owl#>.> <<http://sws.geonames.org/5325223/>>

owl:sameAs<<http://dbpedia.org/resource/Auburn,_California>.> <<http://rdf.freebase.com/ns/m.Or2rz>>

owl:sameAs<<http://dbpedia.org/resource/Auburn._California>.>

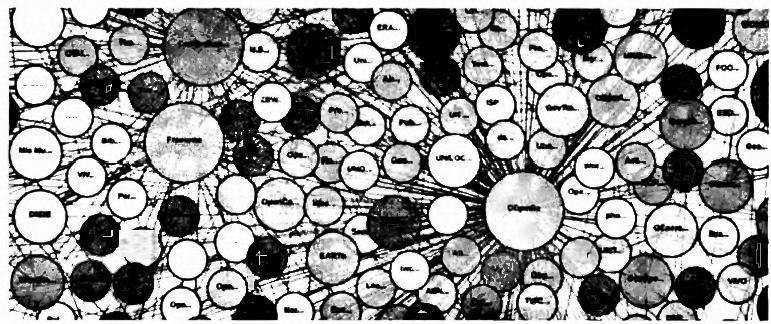


语义网是一个愿景，该怎么建设呢? Linked Open Data提供了一个可能的途径，即语 义网是由Linked Open Data 构成的。通过上述原则，把各个Open Data数据源连接起来以 后，就可以使用SPARQL(RDF 数据库的查询语言)进行跨数据集的查询。

在一个 SPARQL 查询语句中，可以把多个数据源包含进来，并且包含相关的连接集 合 (Link Set)。我们保持原有的数据集原封不动，但是将标识符连接关系保存到另外一个 文件中，这就是Link Set。于是，各个数据源以及各个数据源的实体间的连接关系成为查 询的基础。查询系统可以基于这些数据集和Link Set 完成查询。

|  |
| --- |
| #包含数据集和链接集的SPARQL 查询  **Select** variable-list  **From** dataset1,  dataset2,  linkset  **Where**  {  graph pattern #查询条件  『 |

目前，Linked Open Data的数据集越来越多，涵盖学术研究、生命科学、政府、演 员、导演、影片、饭店等众多的信息。包含超过149423660620个RDF 三元组的9960 个数据集(至2017年)①建立了连接关系。图9-3展示的是截至2014年4月的LOD 云 图 (Link Open Data Cloud)。

e.

Mon.

ML..

第L

Ord

Vy.

VTY.

**图9-3** **Linked** **Open** **Data** **Cloud(2017年2月版本，局部)**

注：完整版请参见<http://lod-cloud.net/versions/2017-02-20/lod.png>

**9.4.2** **知识图谱**

**1.** **什么是知识图谱**

随着开放链接数据 (Linked Open Data) 等大量结构化 RDF 数据源的发布，互联网



①<http://stats.lod2.eu/.>



已经逐步从仅仅包含网页以及网页间超链接的文档万维网 (Web of Document), 发展出包 含大量实体描述以及实体间丰富的语义关系的数据万维网 (Web of Data)。

在这个背景下，知识图谱应运而生。Google 公司于2012年首次提出知识图谱的概念， 它的目的是使用现实世界中存在的各种概念和实体以及这些概念、实体之间的关系，改善 搜索的结果。知识图谱可以认为是超大规模的知识库。

Amit Singhal博士认为“The world is not made of strings,but is made of things”, 这句话很好地总结了知识图谱的特点。传统的互联网上的网页，虽然有各种各样的媒体类 型，包括文本、语音、视频等，但绝大部分内容是文本，信息检索的主要方式是对用户的 关键字查询和网页内容做字符串 (String) 的匹配。 “thing” 指的是各种概念和实体以及 这些概念和实体的关系。它是信息的精华，是人们在检索信息时真正需要的内容。

**2.** **知识图谱和本体的区别和联系**

知识图谱在本体的基础上进行了丰富和扩充，这种扩充主要体现在实体 (Entity) 层 面。本体主要描述了概念以及概念之间的关系，可以认为是知识图谱的数据模式 (Sche- ma) 。 为知识图谱构建模式，即为其构建本体。知识图谱依据本体的模式增加丰富的实体 信息。

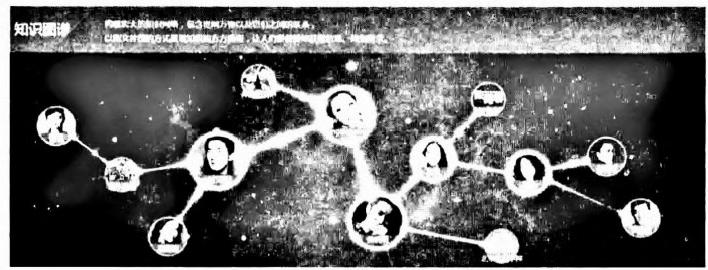
比如本体里描述了两个概念——老师和学生，并且规定了老师和学生之间的指导关 系。建立知识图谱则是增加一系列的老师和学生的实例，并且具体地建立这些老师和学生 之间的指导关系。由此看来，知识图谱不是本体的替代品。

知识图谱中的概念与概念之间存在各种类型的关系，同时，知识图谱中的实体之间存 在这些类型的关系。知识图谱可以看成一张巨大的图，图中的节点表示实体或概念，图中 的边则构成关系。在知识图谱中，每个实体或者概念用一个全局唯一的ID 来标识，称为 它们的标识符 (Identifier) 。 实体可以拥有属性，用于刻画实体的内在特性。每个属性都 是以“<属性，属性值>对”(Attribute Value Pair,AVP) 的方式来表示的。关系 (Re- lation) 用来连接两个实体，刻画它们之间的关联。这个图模型可用W3C 提出的资源描述 框架RDF 来表示。

**3.** **知识图谱实例**

在Google 推出知识图谱 (Knowledge Graph) 之后，国内外的其他互联网搜索引擎公 司也纷纷构建了自己的知识图谱，包括微软的 Satori、搜狗的知立方、百度的知心(见 图9 - 4)等。

Baid **知识图** 2种' n FNG 下 临n



**图9-4** **百度知识图谱**

知识图谱，按照其概念、实体的覆盖面，可以分为通用知识图谱和专用知识图谱两 类。(1)通用知识图谱。目前已经发布的知识图谱很多都是通用知识图谱，它强调的是广 度，也就是包含更多的实体。其主要应用领域是搜索等业务，对准确度的要求不高。同 时，由于实体众多，类别千变万化，所以很难为其生成一个全局的完整的本体层，进行统 一管理。(2)专用(领域)知识图谱。领域知识图谱通常为各个领域或者行业服务，用于 辅助进行各种复杂的分析应用以及决策支持，它的概念、实体、属性和关系具有具体行业 的意义。领域知识图谱的准确度要求高，具有严格的数据模式。

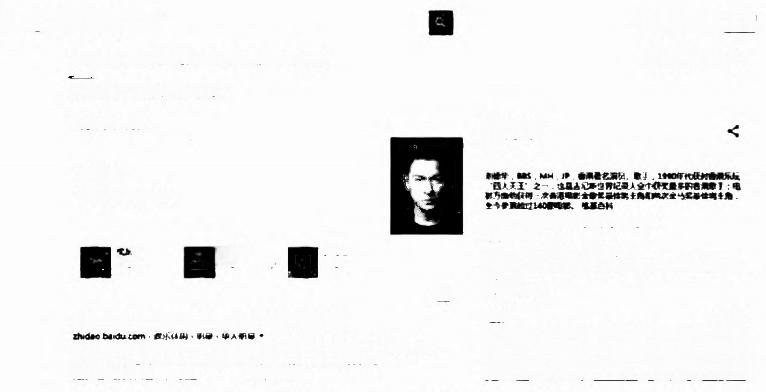
**4.** **知识图谱的应用**

知识图谱在语义搜索、智能问答系统、推荐系统、数据挖掘、知识工程、数字图书馆 等领域有着广泛的应用。下面以语义搜索为实例做一个简要的介绍。

知识图谱的出现开启了语义搜索的大门。搜索引擎利用知识图谱的帮助，能够提供的 不仅仅是网页链接，甚至可以提供答案本身。

图9-5展示了当用户提交“刘德华的年龄”查询时， Google 提供的搜索结果，这个 结果包括：(1)相关的网页链接以及摘要列表，这是传统的搜索引擎就能够提供的。

(2)在结果网页的上半部给出搜索的直接精确的答案.为“55岁”(2016年查询).并且 列出了相关人物的年龄，包括成龙、李连杰、郭富城等。(3)在网页的右侧，以知识卡 片 (Knowledge Card) 的形式列出了刘德华的相关信息，包括出生年月、身高、配偶、 专辑、子女等。知识卡片为用户查询中所涉及的实体提供详细的结构化信息，让用户参 考。这些结构化信息，可以看作从知识图谱里提取的针对特定查询 (Specific Query) 的 片段。



Google 郊踏毕的年驰

约万7.000集忙律(用a58聘)

刘锦埠/年

**55岁**

**1961年9月27日**

鸟 唯 50岁

野

**成龙、刘德华，张学友还有李连杰年龄多少?\_百度知道**

20071624H 单华，怅学友让白争垂杰4多少1前：Ⅲ47195444月78.山 人， 青名功头薪尺F星1964、1992年任时评出华高离年、1909年伙母高语.

中资多大年验?n 度 0 201年1]L4

ca 时 田 :

P8 安 +

**刘德华**

满

**十：1961** **月27H(55岁** **)** **地** **大**

**骗：175×**

**配：肠牌(忙网2008年)**

**◆帽：5够本、卑牛章，警不兴，** **,如导作是L** **佛** **☆** **,** **啊**

以 ，

**7位：汉映控马**

电影 **3**

郸 · 宝 L

*t45+* *森*

62岁

**图9-5** **利用知识图谱增强的Google搜索引擎的搜索结果(2016年)**

搜索引擎能够直接返回正确答案以及实体的结构化信息，显示了知识图谱的强大威 力。同时，这类应用也遇到了一个挑战，搜索引擎必须了解用户真正的搜索意图，比如用 户查询中包含“李娜”这个实体，那么这个“李娜”到底是网球运动员、歌手、舞蹈家还 是其他人，需要确定。 一般可以根据用户以前以及近期的搜索记录，理解用户的搜索意 图，优先返回某些结果。在一些难以准确判定实体的情况下，可以考虑返回多样性的查询 结果。

**9.4.3** **知识图谱的创建**

知识图谱特别是通用知识图谱，规模要足够大才能发挥其应有的作用。表9-1列出 了主要的知识库及其规模。

**表9-1** **主要知识库的规模(2015年)**

|  |  |  |  |
| --- | --- | --- | --- |
| 知识库(知识图谱) | 数量 | | |
| 实体(Entities) | 关系类型(Relation Types) | 事实(Facts) |
| Freebase | 40M | 35000 | 637M |
| Wikidata | 13M | 1643 | 50M |
| DBpedia | 4.6M | 1367 | 60M |
| YAGO2 | 10M | 72 | 120M |
| Google Knowledge Graph | 570M | 35000 | 18000M |

我们希望知识图谱不仅包含大量高质量的已有知识，而且能够及时发现和添加新的知 识，完善不那么常见的知识。首先需要创建初始的知识图谱，然后对其不断丰富。文 献①②对如何创建知识图谱、如何丰富知识图谱做了详细介绍，这里简述如下。

构建知识图谱的数据来源包括百科类站点、各种垂直站点的结构化数据以及各种半结 构化数据(比如HTML 网页的表格)等。(1)维基百科 (Wikipedia) 。 通过大量用户的 协同编辑，维基百科的内容不断丰富和完善，已经成为最大的在线百科全书，它的质量可 与《不列颠百科全书》媲美。从维基百科的网页可以抽取实体、实体的同义词 (Syno- nym)、 实体的<属性，值>对 (Attribute Value Pair,AVP)、实体的关系、实体的类 别、概念的上下位关系等信息。(2) Linked(pen Data(LOD) 。LOD项目在发布各种语 义数据的同时，通过owl:sameAs 将新发布的语义数据中涉及的实体，和 LOD 中已有数 据源所包含的潜在同一实体进行关联，从而实现了手工的实体对齐 (Entity Alignment)。 LOD 不仅包括如DBpedia 和 YAGO 等通用语义数据集，而且包括如MusicBrainz 和Drug- Bank 等特定领域的知识库。通过整合 LOD 中的部分语义数据，特别是垂直领域的各种知 识，可以大幅度提高知识图谱的覆盖率。(3)垂直领域站点。 Web 上存在大量高质量的垂 直领域站点(包括各类电商网站、点评网站等),这些站点包含很多存放在数据库中的数 据，称为 Deep Web。这些站点通过动态网页技术，将保存在数据库中的各种领域相关的 结构化数据，以HTML 表格的形式展现给用户。可以通过收购这些网站获得其数据，或 者通过表单填充技术 (Form Filling), 爬取网站的Deep Web内容，并且对结果网页中的 结构化信息进行自动化抽取，获得这些站点包含的数据(知识)。(4)长尾 (Long Tail) 站点。Web 上存在大量长尾的站点，这些站点提供的数据与主流的垂直领域站点所提供的 内容具有很强的互补性。我们可以构建面向站点的包装器 (Site-specific Wrapper, 面向特 定站点的网页结构分析软件),实现AVP 自动抽取。具体原理是，从当前待抽取站点采样 并标注几个典型详细页面 (Detailed Page)。利用这些页面，通过模式 (Pattern) 学习算 法，自动构建出一个或多个以XPath 表示的模式，然后将这些模式应用在该站点的其他详 细页面上，从而实现自动化的AVP 抽取。(5)搜索日志。通过搜索日志，进行实体和实



①<http://www.sohu.com/a/53506297_116235.>

②<http://blog.csdn.net/starzhou/article/details/51434548.>

体属性的挖掘。搜索引擎公司积累了大量的搜索日志，每条搜索日志包含<查询，点击的 页面链接，时间戳>等信息。这些搜索日志是宝贵的财富，通过挖掘搜索日志，可以发现 最新出现的各种实体及其属性，从而保证知识图谱的实时性。我们可以从查询的关键字和 点击的页面所对应的标题中，抽取实体及其属性。为了完成抽取任务， 一种常用的做法是 针对每个实体类别，挑选出若干属于该类别的实体(及相关属性)作为种子(Seed), 找 到包含这些种子的查询和页面标题，构造正则表达式或文法模式。这些模式将用于抽取查 询和页面标题中出现的其他实体及其属性。如果当前抽取到的实体尚未包含在知识图谱 中，则该实体成为一个新的候选实体。如果当前抽取到的属性尚未出现在知识图谱中，则 该属性成为一个新的候选属性。为了保证知识的质量，我们仅保留置信度较高的实体及其 属性。新增的实体和属性将作为新的种子，用于发现新的模式，此过程不断迭代。这里举 一个简单的例子。比如，我们知道《非诚勿扰》是一档相亲类综艺电视节目。我们搜索 “非诚勿扰”,获得一系列网页，这些网页的标题为“……非诚勿扰综艺节目…… ”“…… 非诚勿扰综艺全集……”“……非诚勿扰最新一期……”“……非诚勿扰……相亲 …… ” 等。通过分析，我们提取如下模式，“……{}综艺节目……”“……{}综艺全集 …… ” “ …… {}最新一期 …… ”“ …… {} ……相亲……”等({}表示实体出现的位置),用于 提取新的类似的综艺节目(新实体)。过了一段时间，有一个电视台开播了另外一档相亲 类节目，称为《我们相亲吧》。人们对其很感兴趣，纷纷通过互联网进行搜索。我们结合 搜索关键字和搜索结果网页的标题，使用上述模式，就能够把“我们相亲吧”这个新的实 体抽取出来。

简单总结，知识图谱的一般构建过程如下：首先，通过收集来自百科类站点和各种垂 直站点的结构化数据来覆盖大部分常识性知识。其次，通过从各种半结构化数据(比如 HTML 表格),抽取相关实体的<属性，值>对来丰富实体的描述。此外，通过搜索引擎 的日志发现新的实体或新的属性，不断扩展知识图谱的覆盖率。

抽取出来的知识称为抽取图谱，在加入统一的知识图谱之前需要经过必要的处理。

(1)实体对齐 (Object Alignment)。实体对齐的目的是发现具有不同ID 但代表真实世界 中同一对象的那些实体，然后将这些实体归并为一个具有全局唯一标识的实体，添加到知 识图谱中。各大搜索引擎公司普遍采用的方法是实体聚类，聚类的关键在于定义合适的实 体相似度。这些相似度度量，一般遵循如下原则，即具有相同描述的实体可能代表相同对 象(字符相似);具有相同<属性，值>的实体可能代表相同对象(属性相似);具有相同 邻居的实体可能指向同一个对象(结构相似)。此外，利用来自LOD 中已有的对齐标注数 **据(使用owl:sameAs 关联两个实体)作为训练数据，然后结合相似度计算，可以使用** **基于图的算法(如标签传递** **(Label Propagation) 等算法),发现更多相同的实体对。上** **述方法无法保证100%的准确率，因此其结果作为候选知识，需要人工进一步审核和过滤。**

**(2)不一致性的解决。当融合来自不同数据源的信息构建知识图谱时，有一些实体会同时** **属于两个互斥的类别(比如某个实体，从数据源1来看属于类别1,从数据源2来看属于** **类别2,类别1和类别2互不相容),或某个实体的一个属性(比如某人的性别，从数据源** **1来看是男的，从数据源2来看是女的)对应多个值，出现不一致性问题。解决该问题的** **一个简单有效的方法是充分考虑数据源的可靠性，以及不同信息在各个数据源中出现的频** **率等因素，然后决定最终选用哪个类别或者哪个属性值。(3)知识图谱模式** **(Schema** ) **构建。为知识图谱构建模式，相当于为其建立本体** **(Ontology) 。** **最基本的本体包括概念、** **概念层次、关系、属性、属性值类型、属性的定义域** **(Domain) 与值域** **(Range) 等。在**

此基础上，可以额外添加规则 (Rule) 和公理 (Axiom) 来表示模式层更加复杂的约束关 系。Google 等公司普遍采用的方法是自顶向下 (Top-Down) 和自底向上 (Bottom-Up)

相结合的方法。自顶向下的方法是指通过本体编辑器((ntology Editor) 预先构建本体， 然后用定义好的模式，抽取属于某个类别 (Type) 或满足某个属性 (Property) 的新实体 (或实体对)。自底向上的方法则通过各种抽取技术，特别是通过搜索日志和 Web Table, 抽取新的实体类别、属性和关系，并将置信度高者合并到知识图谱中。自顶向下的方法有 利于抽取新的实例，保证抽取质量，自底向上的方法则能发现新的实体类别、属性和关 系，两者是互补的，应该结合使用。

**9.4.4** **知识图谱的挖掘**

为了进一步提高知识图谱的知识覆盖率，还需要在知识图谱上进行挖掘。换句话说， 在知识图谱上进行挖掘，目的是进一步丰富知识图谱。

知识图谱上的挖掘包括实体重要性排序、相关实体挖掘以及推理等。(1)实体重要性 排序。搜索引擎识别用户查询中提到的实体，并通过知识卡片展现该实体的结构化摘要。 当查询涉及多个实体时，搜索引擎应该选择与查询更相关且更重要的实体来展示。实体的 相关性度量是和查询相关的，需在查询时在线计算；实体重要性与查询无关，可离线计 算。知识图谱中的节点是各种类型的实体，图谱中的边代表各种丰富的语义关系，也就是 说，知识图谱可以看作包含异构节点和异构的边的一张图。考虑这些因素，给予实体和语 义关系一个初始重要性，然后使用带偏的 (Biased)PageRank 算法①,可以对实体的重要 性进行排序。(2)相关实体挖掘。在相同查询的结果中共现的 (Co-Ocurrcnce) 实体，或 者在同一个查询会话 (Session) 中被提到的其他实体，称为相关实体。挖掘相关实体的一 种做法是将这些查询结果或者会话看作虚拟文档，将其中出现的实体看作文档中的词条。 然后，使用主题模型(比如 LDA 等②)分析虚拟文档集中的主题分布，每个主题包含1 个或者多个实体。在同一个主题中的实体，即互为相关实体。(3)推理。通过推理 (Rea- soning/Inference), 我们可以发现隐含知识。推理功能一般通过可扩展的规则引擎来完 成。知识图谱上的规则包括两大类。一类是针对属性的，即通过数值计算来获取实体的属 性值。比如.知识图谱中包含某人的出生年月.可以通过当前日期减去其出生年月获取其 年龄。另一类规则是针对关系的，即通过链式法则发现实体间的隐含关系。比如，我们可 以定义规则，某个人的孩子的母亲，是某个人的妻子。利用这条规则，当已知王二的孩子 是王小二，王小二的母亲是王大花，那么可以推导出，王二的妻子是王大花。

**9.5** **思考题**

(1)语义网概念。

(2)语义网体系结构。



①关于带偏的PageRank 算法的详细信息.请读者参考相关资料。

② 请参考第7章“文本分析”。

(3)XML 标记语言及其实例。

(4)RDF 及其实例。

(5)Ontology 及其实例。

(6)知识库及其实例。

(7)知识图谱与知识图谱实例。

(8)知识图谱的创建与知识图谱的挖掘。

(9)知识图谱的应用。