Steffen Staab, Rudi Studer. Handbook on Ontologies. Springer, 2004
Part II Ontology Engineering
9 Ontology Learning

9 本体学习

Alexander Maedche and Steffen Staab

曾新红译
2005.8

文摘：本体学习可以极大地方便本体工程师进行本体构造。我们这里所说的本体学习概念包括许多互补的规范，它们以不同类型的非结构化和半结构化数据为原料，以支持一种半自动化的本体工程处理。我们的本体学习框架依次进行本体导入、提取、修剪和精化，给本体工程师提供了丰富的本体建模协调工具。除了总的体系结构，我们还在这篇论文里介绍了一些本体学习领域可仿效的技术，这些技术已在我们的本体学习环境 KAON Text-To-Onto 中实现。

9.1 介绍

本体构成了一个特定兴趣领域（为一组人共享）的形式化概念模型。要让本体成为信息系统的组成部分，可能要从那些与数据处理（例如查询）和可视化（如设计）主要相关的软件视角（aspect）模块化出许多主要与领域（如分类结构 taxonomic structures）相关的软件视角。

有人可能会争辩道，人们在此时会遇到的障碍是，这样的信息系统软件不可能用一种隐式的领域认识来建立，而有必要让领域的概念模型显式化——这可能是一项困难的任务，会导致众所周知的知识工程瓶颈。当人们回答这种争论时，也发现在软件工程中的确如此：为了能够方便地改写和扩展结构（这是更快和更实惠地保持本体工程的追求），你必须使它们显示化。虽然在过去十年中本体工程工具已经成熟，但手工建立本体仍然是一项冗长乏味的麻烦工作。

因此，在将本体用作信息系统的基础时，人们必须面对开发时间、困难、信心和本体维护等问题。最终的结果类似于知识工程师在过去的二十多年为知识获取努力寻求方法或为定义知识库努力制作工作台所遇到的情形。一种已被证明对知识获取极为有益的方法是，知识获取与机器学习技术的集成。

本体学习[22]的目标是集成大量的方法（discipline）以方便本体的构建，尤其是本体工程[49，11]和机器学习。因为由机器进行完全自动的知识获取还在遥远的未来，所以总的过程是有人工干预的半自动化过程。它依靠的是“平衡合作建模”范式[33]，描绘了一种构造本体的人工建模者和学习算法之间的协调交互。我们在这里要描述的就是这种头脑中的目标，一种将本体工程和机器学习结合在一起的方法。

组织

本章组织如下：9.2 节介绍本体学习机器相关组件的总体系结构。9.3 节介绍各种互相补充的基本本体学习算法，它们可以作为本体学习的基础。9.4 节描述了我们如何在一个名为 KAON Text-To-Onto 的具体系统中实现我们的本体学习构思。9.5 节调查了相关的工作。

9.2 本体学习的体系结构和处理模式

本节的目的是介绍本体学习的总体系结构及其四个主要组件。以后再继续详细介绍我们为 KAON Text-To-Onto 系统开发的概念模型。

因此，我们的过程模型建立在数据挖掘的主要思想上，这个过程（例如[6]）有事务和数据理解、数据准备、建模、评价、配置等几个阶段。这意味着我们的本体学习概念使所有的过程步骤都透明了——与某些备受关注的（focused）本体学习应用形成对照，人们可以为其配置具体的过程途径。具体的过程途径必须通过 KAON Text-To-Onto 提供的通用模型来进行配置。

- **本体管理组件**：本体工程师使用本体管理组件来手工处理本体。特别是它允许包含现有的本体，允许对它们进行浏览、确认[48]、修改、改编（versioning）和演化[29]。
- **资源处理组件**：这个组件包含发现、导入、分析和转换相关输入数据的广泛技术。其中一个重要的子组件式自然语言处理系统。资源处理组件的总任务是生成一组预处理数据，作为算法库组件的输入。

Fig.9.1. 本体学习概念体系结构

- **算法库组件**：这个组件是整个框架的算法中枢。它为包含在本体模型中的本体片断的提取和维护提供了许多算法。为了能将不同学习算法的提取结果结合在一起，有必要将输出标准化为一种共同的方式。因此我们为所有学习方法提供了一种共同的结果结构。如果若干提取算法获得了重叠或互补的结果，它们会被结合起来，只为用户展示一次。
- **协调组件**：本体工程师使用协调组件与资源处理和算法库本体学习组件进行交互。它为本体工程师提供了详尽的用户界面帮助选择相关数据、应用处理和转换技术或启动一个特定的提取机制。数据处理也可以通过选择一个需要特定表示的本体学习算法来激发。处理结果可以使用结果集结构来合并，并以本体结构的不同角度展示给本体工程师。

下面我们将给出这四个组件的具体概貌。

9.2.1 本体管理

作为我们这个方法的核心，我们已经建立了本体管理和应用基础设施（称为 Karlsruhe Ontology）和语义 Web 基础设施（Semantic Web Infrastructure） (KAON; 以及参见[39])，使人们可以方便地进行本体管理和应用。

KAON 基于一种[37]介绍的本体模型。简单说来，本体语言是基于 RDF(S)的，但是将本体原语与本体本身完全分开（从而避免自我描述的 RDFS 原语（例如 subClassOf()）缺陷），为元类（meta-class）建模和合并若干共同使用的建模原语（如传递、对称、反转属性或基数）提供方法。所有信息都被组织在所谓的 OI 模型（ontology-instance models, 本体-实例模型）中，既包含本体实体（概念和属性），也包含它们的实例。这允许概念和它们众所周知的实例被组入自我包含的单位。例如，一个地理信息 OI 模型可能包含概念 Continent 及其 7 个众所周知的实例。一个 OI 模型可以包含另一个 OI 模型，从而使被包含 OI 模型的所有定义都可以被自动获得（参见[23]）。

Fig.9.2. OI 模型示例

一个词汇 OI 模型通过能反映本体实体的不同词汇属性的特定条目扩展了 IO 模型。例如，像 PLANET-VENUS 这样的概念（以及关系）有一个标签（如“Venus”）、同义词（例

如“evening star”、“morning star”)、一个词干和正文文件（textual documentation）。

在词汇条目和实例之间存在一种 n：m 的关系，它由属性 REFERENCES 建立。因此，相同的词汇条目（例如"his jaguar"）可以和若干元素相联系（例如，一个表示 Jaguar 车的实例或一个表示美洲豹（jaguar cat）的实例）。词汇条目的值，也就是它的字面存在（literal occurrence），由属性 VALUE 给出，而值的语言则通过 INLANGUAGE 属性指定，该属性涉及 ISO standard 639 所定义的所有语言的集合。

9.2.2 协调组件

本体工程师使用协调组件来选择输入数据，例如 Web 上的 HTML 文档这样的相关资源，它们将在下一步的发现过程中被使用。其次，利用协调组件，本体工程师也可以在一组资源处理方法（可从资源处理组件钟获得）和一组算法（可从算法库中获得）中进行选择。

9.2.3 资源处理

正如之前所提到的，该组件包含*发现*、*导入*、*分析*和*转换*相关输入数据的广泛技术。其中一个重要的子组件就是自然语言处理系统。该组件的总任务是生成一组预处理数据，作为算法库组件的输入。

资源处理策略因输入数据的类型而不同。

• 半结构化文档，如字典，可以转换成一个预先定义的关系型结构，如[16]所述。HTML 文档可以索引和简化为自由文本。

• 为了处理自由文本，我们访问语言依赖型自然语言处理系统。例如，对于德语我们使用了 SMES(Saarbrucken Message Extraction system)，一个简单的文本处理器[35]。SMES 含有一个基于习惯表达的*表示器*（tokenizer）,一个包括不同词典的*词汇分析*组件，一个*语型学分析*模块，一个*指定实体识别器*，一个*局部语言（part of speech）附加*以及一个*知识块解析器*。对于英语，可以采用一个建立在多种可利用语言处理资源之上的系统，如[7]中所述的 GATE 系统。在目前支持的实现中（见 9.4 节），我们只使用了著名的 Porter 填塞算法[42]，作为一个很简单的规范自然语言术语的方法。

按照这些策略（或类似的策略）进行预处理之后，资源处理模块将给定的数据转换成特定算法的关系表示。

9.2.4 算法库

本体可以被描述为概念、关系、词汇条目和这些实体之间链接的集合。一个本体可以通过遵循这个规范（specification），使用不同的基于预处理输入数据的算法来建立。因此，不同的算法可能产生交叠或互补的本体片断的定义。例如，一个专门处理同格（apposition）的信息提取模块可能发现子类关系，类似于用 kNN(参见,如[40])统计技术发现的分类。然而，有相关规则的机器学习，可能被进一步用来检测普通的二元关系[25]。因此，我们可以重用库中的算法来获得不同的本体定义片断。

随后，我们会介绍一些在我们的实现中可获得的算法。通常，我们使用一种结果合并方法，即，库中提供的每个算法都按照上面拟定的本体结构产生规范化的结果，共同促成一个聚合的本体定义。对于交叠结果如何以一种真正的多策略学习方式来共同促成一个通用的结果，将来还需要做更多的研究。

9.3 本体学习算法

这里我们会给出一些学习算法例子。它们涵盖了本体定义的不同部分——这些部分也可以被彼此隔离地评价。

9.3.1 词汇条目和概念提取

提取可以表明概念的相关词汇条目的一种简单技术，就是计算给定（语言预处理过）文档集合（corpus D）中术语的频率。通常这种方法是基于一种假设：一个特定领域文本集合中的高频术语预示着相关概念的存在。信息检索研究已经表明，有比简单频率计算更有效的术语加权方法。人们在为术语加权追求一种标准的信息检索方法，基于以下度量：

- **词汇条目频率 $\mathbf{lef}_{l,d}$** 是词汇条目 $l \in L$ 在一文档 $d \in D$ 中的出现频率。
- **文档频率 $\mathbf{df}_l$** 是 l 所处的文档集 D 中的文档数量。
- **文档集频率 $\mathbf{cf}_l$** 是 l 在整个文档集 D 中的总出现数量。

读者可能注意到 $df_l \leqslant cf_l$ 并且 $\sum_d lef_{l,d} = cf_l$。术语的相关性是基于信息检索度量 **tfidf（术语频率反转文档频率）**来衡量的。

定义 1. *设 $\mathbf{lef}_{l,d}$ 是词汇条目 l 在文档 d 中的术语频率，df_l 是词汇条目 l 的总文档频率，那么词汇条目 l 对于文档 d 的 tfidf 为：*

$$tfidf_{l,d} = lef_{l,d} * \log\left(\frac{|D|}{df_l}\right) \tag{9.1}$$

Tfidf 为一个词汇条目在一个文档中的频率加权，带有一个因数，当它出现在几乎所有的文档中时这个因数冲减了其重要性。因此出现太少和太频繁的术语比那些保持平衡的术语排列次序要更低。不仅为一个文档，而且为整个文档库排列一个术语的重要程度，词汇条目 l 的 tfidf 值如下式计算：[4]

定义 2.

$$Tfidf_l := \sum_{d \in D} tfidf_{l,d}, \qquad tfidf_l \in R \tag{9.2}$$

用户可能会定义和改变一个**阈值** $k \in R^+$, $tfidf_l$ 必须超过它。然后这个阈值被用来从文档集探测术语，让它们包含进词汇条目集合，也可能进入概念等级结构。

9.3.2 分类关系的提取

分类关系的提取可以用不同的方法来做。在我们的框架中主要包括以下三种方法：

- 使用聚类（clustering）的基于统计的提取
- 使用分类（classification）的基于统计的提取
- 词典句法模式（lexico-syntactic pattern）提取

下面我们将简短地描述这三种方法。

聚类（clustering）

使用聚类机制，有关单词的分布式的数据可以用来建立概念，并从零开始将它们嵌入一个等级结构。

一个分布表示（distributional representation）用术语出现在一个描写性上下文中的（加权）频率来描述一个术语。这种描写（delineation）可以用术语的顺序邻接（“多少术语出现在已表示和正在表示的术语之间？”——这就是我们所用的），一个句法描写（“哪些术语可能对已表示的术语有句法依赖？”），或一些文本结构标准（“什么术语出现在由 HTML 标记勾画出的同一段落中“）来定义。

聚类可以被定义为，基于分布表示将对象组织成组（其成员在某一方面是类似的）的过程（见[15]）。通常有三种主要的聚类类型：

1. 凝聚型（agglomerative）：在初始阶段，每个术语都定义为构成它自己的一个簇。在成长阶段，通过合并最相似/最少不同的簇反复生成更大的簇，直到达到某个

停止标准。

2. 分割型（partitional）：在初始阶段，所有术语的集合是一个簇。在细化阶段，通过把最大的簇或最不相似的簇劈成若干子簇，（反复）生成更小的簇。

凝聚型和分割型的聚类技术都可以用来产生术语的等级描述。它们都依赖于相似性（不相似性）的概念，对此已有了相当多的方法（例如，Jacquard, Kullback-Leibler 发散，L1 标准，余弦；参见[21]）。

实际上，分割型聚类（如 K-Means 聚类技术）更快，它可以在运行时 $O(n)$中执行，比较之下，凝聚型聚类则需要 $O(n^2)$，这里 n 是被表示术语的数量。

3. 概念型（conceptual）：通过调查处于两个已表示术语之间的正在表示术语的精确交迭来建立一个术语格。在最坏的情况下，结果概念格的复杂度是 n 的指数次方。因而，人们要不只计算一个子格[47]，要不使用试探法。

任一方法都可以建立术语簇的等级结构，以便于本体工程师更细致地检验。

分类（classification）

当已给定了一个丰富的等级结构时，例如从一个通用的资源（如 WordNet）[32]作了基础层次分类，人们可能又决定通过将新的相关术语分入给定的概念层次来细化分类系统（taxonomy）。于是上面描述的分布表示被用来从训练文档集（training corpus）和具有他们的词汇条目的预定义概念集合中学习一个分类符（classifier）。

然后，人们可以建立相关的未分类术语的分布表示，并让已学习到的分类符建议一个节点，在这个节点上将新的术语作为子类（参见，如[40]）。k nearest neighbor(kNN)和支持向量机是典型的为此目的开发的学习算法。

词典句法模式（Lexico-syntactic Patterns）

为了提取语义关系，尤其是分类关系而使用词典句法模式（以规范表达式的形式出现）的思想，已由[16]提出。基于模式的方法通常是使用规范表达式的试探性方法，最初已成功地应用于信息提取领域。在这种词典句法本体学习方法中，文本被扫描作为不同的词典句法模式的实例，代表着一种兴趣关系，例如分类。因而，基础思想非常简单：定义一个规范表达式，它捕获重复出现的表达式，并将匹配表达式的结果映射到一个语义结构，例如概念间的分类关系。

例子

在[16]中考虑了以下词典句法模式：

$$\ldots NP\{, NP\} * \{,\} \text{ or other } NP \ldots$$

当我们将这个模式应用到一个句子时，可以推断出 NP 对 or other 左边概念的引证（referring）是 NP 对其右边概念引证的子概念。例如从句子

Bruises, wounds, broken bones or other injuries are common.

中可以提取出分类关系（Bruise, Injury）, (Wound, Injury)以及（Broken-bone, Injury）。

在我们的本体学习系统中，我们已将不同的技术应用到保险和电信领域上下文中的字典定义，如[19]中所描述的。在这个系统和方法中的一个重要方面是，现存的概念被包含在整个过程中。与[16]形成对比，提取操作在概念层次执行，因而模式直接在概念上匹配。所以，除了从零开始提取分类关系，系统还可以细化现有的关系和参考现有的概念。

9.3.3 通用二元关系的提取

关联规则已在数据挖掘领域建立，于是可以在一个大数据项集合中发现有趣的关联关

系。许多公司对从他们的数据库中挖掘关联关系有兴趣（例如，可有助于许多商业决策，如消费者关系管理、跨市场和亏本销售（loss-leader）分析）。关联关系挖掘的典型例子是市场篮分析，这个过程通过找出客户置于购物篮中的不同商品之间的关联来分析客户的购买习惯。由关联规则揭示的信息有助于开发市场策略，例如超市中的陈列优化（将牛奶和面包放在邻近的地方可能会进一步鼓励一次进店同时购买这两件商品）。[1]给出了提取给定商品之间关联的具体例子。这些例子基于超市产品，这些产品包含在一组从客户购买收集到的事务中。一个经典的趣闻就是“尿布和啤酒一起买”。

对于本体学习来说，这种数据挖掘算法可以应用到文本中的句法结构和统计共现上。为了描述其工作原理，我们这里给出了一个例子。这个例子基于[25]中描述的真正的本体学习试验。一个由 WWW 旅游信息提供商提供的文本文档集已经处理好了。这个文档集描绘了真实的有关地点、住宿、住宿家具、管理信息或文化项目的对象，例如以下例句：

（1）a.”Mechkenburg’s” most beautiful “hotel”is located in Warnemuende.

b. A “hairdresser” in our “hotel” is a special service for our guests.

c. The hotel Mercure offers “balconies” with direct “access” to the beach.

d. All “rooms” have “TV”, telephone, modem and minibar.

处理例句（1a）和（1b），术语之间的依赖关系被提取（以及更多）。在句子（1c）和（1d）中，前置词的短语附加（最小附加：附加一个前置短语到其最近的名词短语）试探法和句子试探法分别把成对的词汇条目联系起来。因此，四个概念对——除了许多别的以外——从词典中获得知识。

表 9.1 语言学上相关概念对的实例

图 9.3 一个作为提取通用二元关系背景知识库的实例概念分类

用于学习通用关联规则的算法使用分类法（taxonomy）。这种分类法可以用 9.3.2 节中介绍的方法来建立。包含上述概念对的摘录在图 9.3 中描述。在我们的实际试验中，该算法发现了大量有趣而重要的通用二元概念关系。其中一些在表 9.2 中列出。注意在这个表中我们也列出了两个概念对，即（AREA,HOTEL）和（ROOM,TELEVISION），但是它们是修剪过的。原因是有些祖传的相关规则，即（AREA, ACCOMODATION）和（ROOM, FURNISHING），分别有更高的可信度和支持度数量。

表 9.2 被发现的通用二元关系例子

其他算法使用动词作为通用二元关系的潜在指示符，并从文档集数据中获得选择的限制[5]。

9.3.4 本体修剪

如果人们借用一个（通用的）本体到一个给定的领域，修剪是必要的。我们假设在 Web 文档中特定概念和概念性关系的出现对于一个给定概念或关系是否应该保留在一个本体中是至关重要的。我们开发了一种基于频率的方法来确定一个文档集中的概念频率。在一个给定文档集中的高频实体被认为是一个给定领域的要素，但是——和本体提取形成对照——纯粹的本体实体频率是不充分的。

要决定领域相关性，从一个领域文档集检索出来的本体实体要和从一个通用文档集中获

得的频率相比较。用户可以为频率计算选择若干种相关度计量方法。本体修剪算法使用计算好的频率来决定包含在本体中的每个概念的相对相关度。所有现有的概念和关系，在特定领域文档集中更高频的保留在本体中。用户也可以控制那些既不在特定领域文档集也不在通用文档集中的概念的修剪。

9.4 KAON Text-To-Onto

本节介绍已实现的本体学习系统 KAON Text-To-Onto，该系统嵌于 KAON(Karlsruhe Ontology 和语义 Web 的基础)中（参见[39]）。KAON（http://kaon.semanticweb.org）是一个开放源代码本体管理和应用基础设施，其目标是语义驱动的商业应用。它包含一个全面的工具套件，可以很容易地进行本体管理和应用。

图 9.4 KAON OI-Modeler

在 KAON 中我们开发了 KAON OI-modeler，这是一个最终用户应用程序，它为 OI 模型的产生和演化实现了一个基于图形的界面[29]。图 9.4 显示了在一个 OI 模型之上的基于图形视图。

KAON OI-modeler 的一个特有的方面是，它在用户层支持本体演化（见这张屏幕拷贝的右侧）。这张图展示了一个建模对话，用户试图删除概念 Person。在将这个变化应用到本体之前，系统计算了必须被应用的额外变化集合。从属变化树展示给用户，从而允许用户在变化被真正应用之前了解该变化的效果。只有当用户同意了，变化后的东西才会应用到本体。

KAON Text-To-Onto 建立在 KAON OI-modeler 之上。KAON Text-To-Onto 提供了方法将一个文档集定义为本体学习的基础（见图 9.5 的右下方）。在该文档集的上方可以有不同的算法被应用。在图 9.5 所描述的例子中，用户已经选择了不同的参数，并并行执行了一个基于模式的提取和一个关联规则的提取。KAON Text-To-Onto 实现了前述的算法，虽然还没有实现分割和概念聚类。

最后，系统将结果展示给用户（见图 9.5），并为增加词汇条目、概念和概念关系提供图形方法。

图 9.5 KAON Text-To-Onto

9.5 相关工作

直到最近，本体学习*本身*，即本体的全面构建还不存在。这里我们给了读者一个现有工作的全面概述，这些工作是真正研究和实践过的技术，用来解决本体学习的部分问题。

只有少许方法描述了从数据提取本体的框架和工作平台开发：Fature & Nedellec [9]介绍了一种合作型机器学习系统 ASIUM，该系统基于句法输入获得分类关系和动词的次范畴化结构。ASIUM 系统基于句法上相关的动词层次化地聚类名词，反之亦然。从而，他们协作扩展词典、概念集和概念等级结构。还有一个更近期的方法由 Missikoff 等[36]提出，他们将一个焦点集中在合意建造（consensus building）的本体工程环境，与用于提取具有多词标签（例如“swimming pool”）的概念和用于提取关系（不是 taxonomy, 例如 pertainance）的复杂工具相结合。

Hahn 和 Schnattinger[14]介绍了一种维护特定领域 taxonomy 的方法学。随着新概念从真实世界的文本中获得，一个本体是增量式更新的。获取的过程集中在处于概念假定的产生和精化之下的不同形式的证据的语言和概念“质量”上。他们的本体学习方法嵌入在一个名为 Syndicate[13]的自然语言理解框架中，并且最近他们已将其框架扩展为一个语法和本体的二

元学习器[12]。

Mikheev & Finch [31]介绍了他们的用于“从自然语言获取领域知识”的 KAWB 工作平台。这个工作平台包含了一组工具，用于揭示自然语言文本的内部结构。该工作平台的主要思想是文本表示和文本分析阶段的独立。在表示阶段文本由标注工具方法从一串字符转换为兴趣特征部件（feature）。在分析阶段这些部件为统计收集和推理工具所使用，以发现文本中的重要相关性。分析工具独立于有关部件集合本性的特定假设，并工作在部件元素（表示为 SGML 项目）的抽象层次上。

在许多学科——计算语言学、信息检索、机器学习、数据库、软件工程——中，有大量工作实际已研究和实践了解决部分问题的技术。因此，与本体学习相关的技术和方法可以在诸如选择限制的获取（参见 Resnik[43]和 Basili 等[4]），词感知（sense）解疑和词感知学习(参见 Hastings[51])，形式化上下文中概念格的计算（参见 Ganter & Wille[10]）以及软件工程中的逆向工程（参见 Mueller 等[38]）等术语下找到。

本体学习把大量的研究活动推向一个前景，这些研究活动致力于不同类型的输入，但共享了共有的领域概念化目标。人们可以认识到这些活动在非常不同的团体之间展开。尤其在三个本体学习研究小组[45，28，3]中可以找到进一步的参考。

9.6 结论

我们已将本体学习介绍为一种可以极大地方便本体工程师进行本体构造的方法。在本文中介绍的本体学习概念，其目标为为了方便本体构造而进行大量学科的整合。整个过程被认为是带有人工干预的半自动化过程。它依靠的是“平衡合作建模”范式，描绘了一种为语义 Web 构造本体的人工建模者和学习算法之间的协调交互。