基于语义 Web 的本体映射方法综述*)

袁 洋 李善平
(浙江大学计算机学院 杭州 310027)

摘 要 本体之间的映射是语义 Web 发展中的一个重要问题。事实上,语义 Web 是由多种信息源组成的,每个信息源都以某个本体的形式表示。为了实现这些信息源的信息共享,就要用到本体映射方法。本文首先讨论了本体映射的三种体系结构。然后按照 E. Rahm 提出的分类标准,对现有的一些主要的本体映射方法进行归纳。最后,给出了 4 种方法的比较分析。从中可以看出各种独立匹配方法的组合将是一个极有希望的发展方向。

关键词 语义 Web,本体,映射,分类,方法

The Overview of Ontology Integration Approaches Based on Semantic Web

YUAN Yang LI Shan-Ping
(The Computer College of Zhejiang University, Hangzhou 310027)

Abstract One of the important problems in the development of techniques for the semantic Web is the mapping of ontologies. Indeed, the Web is constituted by a variety of information sources, each expressed over a certain ontology. In order to share the information among such sources, their semantic mapping is required. In this paper, first we discuss the three integration architectures. Then we refer to the taxonomy which is presented by E. Ramh, and apply to summarize various major ontology mapping approaches. At last, we compare four typical mapping approaches following the criteria, and conclude that the combination of the individual matchers is a promising direction.

Keywords Semantic Web, Ontology, Mapping, Taxonomy, Approach

1 引言

现在,互联网已成为人们获取信息最重要的途径,其规模也在以惊人的速度增长着。然而,当前互联网上的绝大多数信息是以人类能理解的格式(例如,HTML)来表示的,而作为智能程序的软件代理(software agent)并不能理解和处理这些信息,互联网的潜力还远远没有挖掘出来。

为了解决这个问题,研究者们提出了下一代互联网的概念——语义 Web[1]。在语义 Web 上,信息是以结构化的形式表示的,而本体则描述了其中的语义。本体是对概念世界的显式说明,它允许人们把领域内的知识表示成概念的分类体系(taxonomies of concepts),概念有自己的属性,概念之间关系则存在各种关系。当信息用本体来标记后,软件代理就能理解其意义,也就可以自动地完成互联网上的信息收集和集成。

由于本体的多样性,要想完成信息交流的任务就必须在本体之间架起语义映射的桥梁。最初,这些映射过程都是由人手工完成的。但随着语义 Web 的发展,在 Web 上用本体表示的信息越来越多,仅仅由人来完成这些工作已经力不从心,因而迫切需要发展一些方法,来自动地或半自动地完成这种映射过程。

使用概念模型的信息集成领域的研究可以分成发现、表达和执行三类[2],本文研究的是如何发现本体之间的语义联系。

2 问题描述及映射框架

2.1 问题描述

由于本体的创建者不同,使用的建模方法不同,因而即使对同一个领域内的问题建模,不同的领域专家开发出来的本体必然存在着差别。本体映射的目的就是找到这些本体之间的语义联系。其中最简单的映射关系是一对一(1:1)的映射,如图 1 所示。

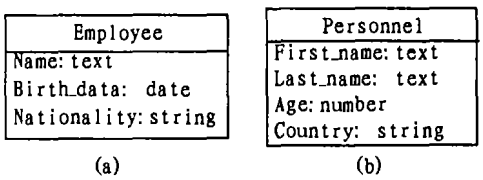

图 1 两个简单本体之间的映射关系

本体 Employee 中的属性 Nationality 和本体 Personnel 中的属性 Country 是 1:1 的映射关系。而属性 Birth_date 和 Age 之间也是 1:1 的关系,但它们并不是简单的等价关系,其转换规则是:Age=This year−Bith_date。

其它的映射关系包括 1:n,n:1,n:m,如属性 Name 与 First_name 和 Last_name 之间就是 1:n 的关系。

2.2 映射框架

处理本体映射问题的基本体系结构有三种[3]:单本体结构、多本体结构及混合结构。如图 2 所示。

在单本体结构中,一个全局的本体为具体的语义说明提供了一个共享的词汇表。所有的信息源都联系到这个全局本体上,因而它们在语义上是一致的。全局本体可以是许多模块化的子本体的组合。

在多本体结构中,每一个信息源都有自己的本地本体,它

*)基金项目:国家自然科学基金资助项目(60174053)。袁 洋 硕士生,从事语义 Web,本体论研究。李善平 教授,从事嵌入式系统,人工智能研究。

们并不一定使用同样的词汇表。每个本体都是独立发展的，它们之间有松散的联系。要完成本体之间的互操作，必须建立映射的规则(链接)。

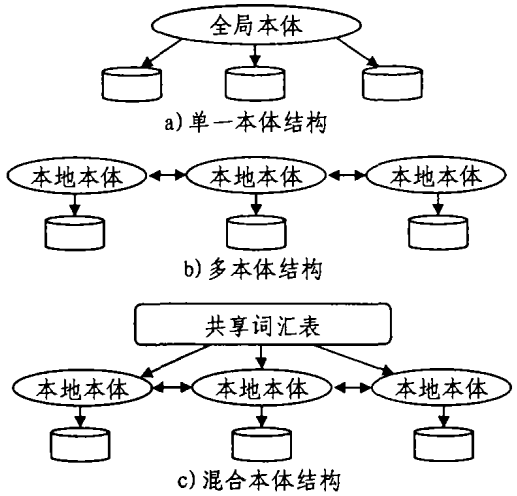

图 2　三种体系结构

在混合结构中，它综合了前两种方法的基本特征以克服它们的不足之处。像多本体方法一样，每个信息源都有自己的本地本体。但本地本体是在一个全局共享的词汇表下发展起来的。共享词汇表定义了领域内的基本术语，在本地本体中这些术语可以组合起来表达复杂的语义。

总的来说，单本体方法建立在紧密联系的基础上，缺乏足够的灵活性，不能适应大的开放式的应用环境。一旦加入了新的信息源，常常会导致全局本体的变化，因而不太适合于大多数本体映射应用环境。而多本体方法和混合方法更适合于完成本体映射的任务。在这两种情况下，都需要发展一些协助映射的方法。

3　本体映射方法

3.1　方法分类

一般来说，完全自动地实现本体之间的映射，而不需要人的干预，这几乎是不可能的事情。因为本体内的一些潜在的语义关系并没有以形式化的方式显式地表示出来，必须由人借助已有的知识和经验才能识别出这些信息。因而，本体映射所用到的匹配方法应该提供一个可能匹配的候选结果列表，然后由人来决定是接受，拒绝，还是需要改变后再接受。在此过程中，人还能加入一些系统没有发现的映射关系。

本体映射中用到的方法主要可以按照 E. Rahm 提出的分类体系进行如图 3 所示的划分[8]。

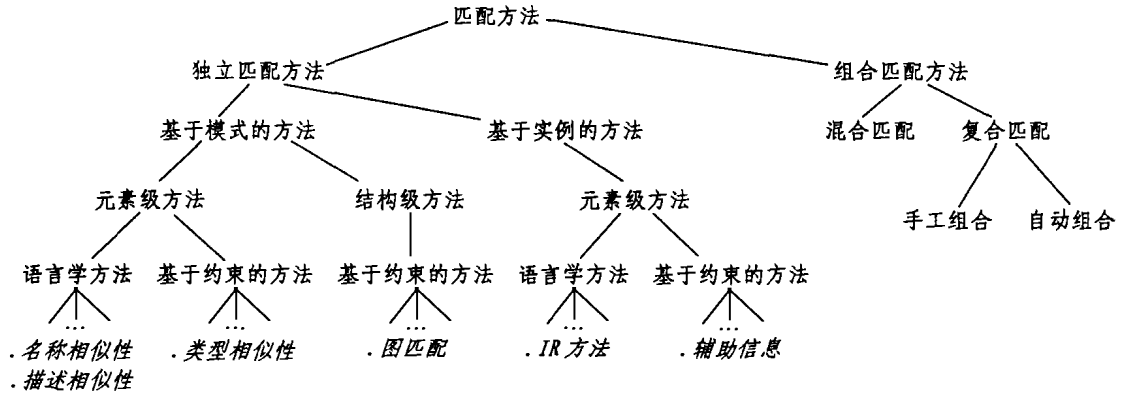

图 3　方法分类

本体映射时可能会用到多种匹配算法(匹配器)。我们可以根据具体的应用要求灵活地选用不同的方法及其组合。在具体实施时有一个如何组合使用它们的问题。我们可以在匹配过程中先后使用多个匹配标准，这是混合匹配法。我们也可以分别执行各个匹配算法，然后再将结果合并，这是复合匹配法。

对于单独的匹配算法，我们可以考虑以下互不相关的分类标准。

模式级与实例级　前者只考虑模式信息，而不考虑实例数据。模式信息包括名称，描述，关系，约束，等等。后者利用了这两方面的信息。

模式级匹配方法值考虑模式信息，不考虑实例数据。可用的信息包括本体模式元素的一般属性，如名称，描述，数据类型，关系类型(part-of，is-a 等)，约束和模式结构等等。一般地，一个匹配算法会找到多个候选结果，每个候选结果都有一个介于 0 到 1 的数值表示它的相似程度。

实例级方法由于利用了数据实例的信息，因而和模式级方法互为补充。它既可以和模式级方法一起使用，互相验证，也可以单独使用。从实例数据中提取出模式元素特征的方法多，如规则，神经网络，机器学习等。一般的实例级方法寻找的是模式元素之间的匹配关系，要找到模式元素组合或结构的匹配就需要比较这些元素组合的数据实例。显然，这样做要遇到的主要问题是模式元素的可能组合空间是极其巨大的。如果不加限制，这样的任务是根本不可能完成的。

元素粒度与结构粒度　前者只考虑本体中独立的概念元素，后者还要考虑这些概念元素的组合。

元素级匹配考虑的是本体中的单个概念、属性或关系，而不考虑这些逻辑概念之间的联系。也就是说，它在匹配父概念时，并不会去考虑其子概念或与其它概念之间的关系。

与之相反，结构级匹配不光要考察单独的对象，还要考虑它们之间的联系。结构匹配可能是完全匹配，也可以是部分匹配，这取决于匹配所要求的完整性和准确性。理想情况下，两个本体中相应结构的所有组成元素都能一一对应，即完全匹配。但实际上，一个本体中的某些元素在另一个本体中找不到对应部分，这时就只能达到部分匹配。针对复杂的情形，为了提高效率，我们可以在数据库中存储已知的等价模式，然后在匹配过程中直接参考这些模式。

基于语言与基于约束　前者基于语言(如名称和描述文本)，后者基于基本的约束信息。基于语言的方法中最常用的信息是元素名称。度量名称相似的标准有很多，如等价关系，同义关系，上义关系(hypernym，若 Y 属于 X，则 X 是 Y 的上义词，如，“出版物”是“论文”的上义词)，以及编辑距离(edit

distance)，甚至词语的发音等。为了发现这些关系，通常都要用到词典。在这方面，自然语言词典是很有帮助的。在具体应用领域中，领域相关的词典由于含有领域知识(常用的专业词汇，简写等)，因而具有特别重要的价值。但是，当前可利用的领域词典较少，需要研究者付出更多的努力。

自然语言中的一词多义现象会极大地干扰名称匹配的过程。为了减少由此产生的误配情况，需要由人或词典提供失配信息。引入上下文内容，有助于在算法中自动利用失配信息。这样的方法很类似于基于结构的方法，这也使得两种方法之间的分别变得模糊。

本体模式中包含的约束信息，有数据类型、取值范围、唯一性、可选性、关系类型和可选值等。如果要比较的双方都有这样的约束信息，就可以它为根据来决定模式元素的相似性。

如果仅使用约束信息进行匹配，得到的往往是n:m的匹配结果。具有相同约束条件的元素可能有好几个，例如，有好几个元素都是string类型。为了进一步区分这些元素，可以和其它的匹配方法(如名称匹配)结合起来使用。

一些结构信息也可以认为是约束信息，如整体与部分的关系(part-of)。这些信息告诉我们哪些元素属于同一个更高级别的元素，这个过程可以在多级结构上传递地进行。当然，这些约束信息也可以看作是结构信息，用结构匹配方法来判定相似性。这样的匹配既考虑了拓扑结构，也考虑了不同的元素类型和可能的不同类型的结构连接。

本体模式的结构是基于一些包容关系的分级结构。在执行基于结构的匹配时，我们既可以从上到下，也可以从下到上地遍历整个模式结构。比较起来，从上到下的算法花费的代价较小，因为一开始所要比较的对象比较少，以后的比较也只要用到前面的比较结果。然而，从实际来看，高层元素的差别是很大的，而底层元素则比较相似。这样的话，从上到下的遍历更有可能得到错误的结果。相反，如果从下到上的遍历，即使中间层和高层结构差别很大，仍然能得到较好的匹配结果。

匹配基数　一些方法只产生1:1的映射关系，另一些方法会产生1:n或n:1的映射关系。匹配基数又分为局部的和全局的。如果只在一条映射规则中考虑，则是局部的。如果在不同的映射规则中考虑，那就是全局的。例如，在两条映射规则中，概念C1分别和T1,T2相似，则其局部基数是1:1，全局基数是1:n。

现有的匹配方法大多是把一个本体模式中的每一个元素与另一个本体模式中具有最高相似性的元素匹配。这样产生的结果在局部是1:1的匹配，在全局则是1:1或1:n的映射关系。现有的大多数方法都不能产生局部和全局1:1和n:m的映射关系，要产生这些映射关系需要在匹配算法中采用更复杂的标准。

辅助信息　大多数匹配方法不仅仅依赖所输入的本体信息，还会用到一些辅助信息，例如字典，以前的匹配结果，还有用户的反馈。

3.2　不同方法的结合

每种匹配方法利用了不同的信息，对于一个给定的匹配任务，各有不同的适应性和价值。因而，组合使用几种方法比单单采用一种方法会产生更好的结果。组合的方式有两种：混合方式集成了多种标准，复合方式则合并各个独立执行的匹配方法的结果。组合多种匹配方法也为同时进行评估提供了可能。

混合匹配方法在整个过程中采用了多个标准。和多个匹配方法的单独执行比较起来，它可以提供更好的候选结果和更好的性能。由于仅符合一种标准的候选结果可以在早期被排除，以及在匹配过程中要综合考虑多种标准，混合匹配方法效率更高。结构级匹配也能从与其它方法如名称匹配联系使用中得到好处。一种组合结构级和元素级匹配的方法是先用一种方法产生部分映射，然后再用另一种方法完成映射。

混合匹配方法可以提供更好的性能，因为它可以减少遍历整个模式结构的次数。例如，元素级匹配的混合方法可以在每个S2元素上同时测试多个标准，然后再测试下一个S2的元素。

另一方面，复合匹配方法则把几个独立执行的匹配方法的结果合并起来，这些方法中也可以包括混合方法。这种合并多个匹配方法的能力使它比混合方法具有更大的灵活性。混合方法通常用硬连接的方法组合同时执行或以固定次序执行多个匹配方法。与之对比，复合方法允许我们以模块化的方法选择所需的方法。例如，我们可以用机器学习的方法组合独立的匹配方法。而且，复合方法在执行顺序上没有特别的要求，我们可以让它们同时执行，也可以让它们顺序执行。在后一种情况下，前面执行的匹配方法的结果可以被后面执行的方法利用，以取得更好的结果。

匹配方法的选择、执行次序的决定和独立运行结果的合并，这些既可以由匹配方法本身自动决定，也可以由人来决定。自动化的方法可以减少人的参与，但是很难获得一个适合于不同应用领域的通用的解决办法(虽然可以通过调整参数来进行控制)。作为可选的方案，可以由人来直接选择匹配方法，决定执行次序，和如何合并结果。这样更容易实施，也给了用户更多控制的余地。在任何情况下，用户的参与都必不可少，因为匹配方法本身只是提供一些候选结果，最终需要用户来决定是接收，拒绝，还是改变结果。

为了处理复杂的匹配任务，还需要在匹配过程中支持多个用户的迭代开发。在复合方法中，各个匹配算法可以按照一定的顺序执行，用户提供的匹配结果也可以作为其中一种独立的匹配算法。对于用户提供的匹配输入，复合匹配方法必须意识到它的权威性，不会去改动它，而把精力放在解决不匹配部分上。

下面，我们按照上面的分类标准讨论一些主要的匹配算法。

4　原型方法介绍及其比较

Cupid[5]

Cupid是一种基于元素级匹配和结构级匹配的混合方法。它可用于数据库、本体论等多种领域的匹配任务。其思想是，如果两个概念的子概念是相似的，那么这两个概念就趋向于相似；如果两个概念具有相似的祖先，那么它们也趋于相似。为了处理同义词、缩略语、首字母缩写，它用到了辅助的信息源，如词典。为了解决共享元素的问题，它在概念树中加入辅助节点以反映共享节点和父节点之间的多重关系。

整个算法分成三步。第一步作语言学上的元素级匹配，并通过名称、数据类型和领域进行分类。这个过程中，复合名词被分解成单个词(如，Company_Name变成{Company，Name})，按照数据类型，语义内容归入不同的类别，然后在每个类别内计算概念元素对之间的语言相似系数，计算中用到了子串匹配和辅助信息源。第二步，把原来的模式转化成一棵概念树，作自底向上的结构匹配。两元素之间的相似性取决于它们的语言相似性以及它们的叶子集的相似性。如果算出的相似系数超过了阈值，那么就增加其叶子集的相似系数。之所以关注叶子集是基于这样的假设，叶节点包含了更多的信息，而且，与中间节点相比，在不同的本体模式中的变化较少。这

一步计算出匹配概念对之间的语言相似系数和结构相似系数的加权平均值。第三步，用这些加权平均值来选出匹配结果。这一步是和具体的应用领域相关的，在 Cupid 算法中没有详细研究。

Similarity Flooding(SF)[6]

SF 的思想是基于相邻概念节点之间的相似传递性，也就是说如果两个概念节点的邻近节点是相似的，那么它们趋向于相似。SF 也是一种综合使用了名称匹配和结构匹配的混合方法。首先，它把模式信息转化成有向图(labeled graph)，然后通过简单的名字匹配得出各个节点之间的初始化相似系数。这时的结果是相当粗略的，不能准确地反映节点之间的语义关系。接着，它用 SF 方法对初始系数进行迭代计算，直到得到收敛值，也就是各个节点对之间最终的相似系数。最后，它用一些过滤方法从数值最高的几个候选节点中找出最合适的一个。与其它模式级匹配方法不同的是，它并没有使用词典，没有利用术语之间语言学上的语法关系。

GLUE[7]

GLUE 系统用机器学习的方法来完成不同本体之间的匹配任务，其思想是多策略学习。它代表了一种自动合并不同匹配器(learner)匹配结果的组合方法，产生的是原子级的 1:1 的映射关系。除了名称匹配器之外，它还用到了几个在预处理阶段经过训练的实例级匹配器。在预处理阶段，用户先给出一些映射实例，然后用这些实例训练 learner，发现其中特有的实例模式(pattern)和匹配规则。用这些模式和规则去匹配整个本体模式，得到候选值的列表。

一个全局的匹配器用同样的机器学习方法融合这些由不同 learner 得出的匹配候选值列表，得到一个综合的列表。在预处理过程中，它也经过了训练，以决定每个 learner 的权值。由于是组合式的匹配方法，加入新的 learner 也很方便。

虽然此方法主要是面向实例的，但它也能利用模式信息。此外，它还能加以扩展，利用用户提供的领域约束信息以提高匹配准确性。

COMA[8]

COMA 系统采用的是复合方法，可以灵活地组合不同的匹配算法及其结果。它所应用的匹配器主要利用模式信息，如元素和结构属性。与其他系统不同的是它可以重用以前的匹配结果，这可显著地提高匹配效率。在匹配过程的不同阶段 COMA 应用了不同的组合策略，如匹配结果的聚合以及匹配候选值的选择。在匹配过程中，它把模式转化成带有根节点的有向无环图，所有算法都基于这个内部表示结构来工作。算法产生的相似值矩阵保存在基于 DBMS 的知识库中。每个模式元素都以从根节点出发的完整的路径名称来唯一标识。

COMA 中应用的匹配算法包括两种元素级的混合匹配算法，Name 和 TypeName，以及三种结构级的混合匹配算法，NamePath，Children 和 Leaves。其中，Children 和 Leaves 在比较元素相似性时都用到了 TypeName 算法。

表 1 不同方法的比较

		Cupid	SF	GLUE	COMA
匹配粒度		元素和结构级	元素和结构级	元素和结构级	元素和结构级
匹配基数(局部/全部)		1:1/n:1	1:1/m:n	1:1/n:1	1:1/m:n
方法分类	基于名称	名称，同义关系，上义关系，同形异义关系	名称，简单的子串匹配	名称，同义关系	名称，同义关系
	基于约束	数据类型，应用约束			数据类型
	结构匹配	用叶节点衡量的子树匹配	SF 方法		叶节点
	实例级方法			Whirl，Baywsian learner	
	重用信息	词典，词汇表		训练实例的比较，查询有效领域值	词典，以前的匹配结果
匹配器组合方式		混合	混合	组合，用机器学习方法自动组合各个匹配器的结果	组合
用户输入		用户可以调整加权系数		匹配和失配规则，人机交互精化结果	可以交互方式影响执行过程

结论 本体是对领域知识概念的抽象和描述，其目的是为了信息的共享和软件的重用。语义 Web 的发展为基于 Internet 实现一个巨大的、虚拟的、分布式的知识仓库提供了可能。而以本体形式表达的信息(知识)则是这个系统的基础。为了实现在本体之间的信息交流，研究者提出了许多方法来发展本体之间的语义联系。

在这些方法中，大多数是采用了多种匹配标准的组合方法。这说明单一的标准并不能提供足够精确的结果。而通过匹配算法的组合，采用多个匹配标准，可以挖掘多方面的本体信息，从而有效地提高了匹配质量。现有的方法大多数是基于模式信息的，而对于大量的实例数据却没有考虑。未来的匹配算法除了考虑更有效地利用模式信息外，还应该更多地挖掘实例数据提供的信息，或者研究一些更有效地组合这些算法的方法。

参考文献

1 Berners-Lee T, Hendler J, Lassila O. The Semantic Web. Scientific American, 279, 2001

2 Maedche A, Motik B, Silva N, Volz R. MAFRA-An Ontology Mapping Framework in the Semantic Web. In: Proc. of the ECAI Workshop on Knowledge Transformation, Lyon, France, 2002

3 Wache H, Vögele T, Visser U, et al. Ontology-Based Integration of Information--A Survey of Existing Approaches. In: Proc. of the IJCAI-01 Workshop: Ontologies and Information Sharing, Seattle, WA, 2001. 108～117

4 Rahm E, Bernstein P. A survey of approaches to automatic schema matching. VLDB Journal, 2001, 10(4): 334～350

5 Madhavan J, Bernstein P A, Rahm E. Generic schema matching with cupid. In: Proc. of the 27th Intl. Conf. on Very Large Databases, 2001. 49～58

6 Melnik S, Garcia-Molina H, Rahm E. Similarity Flooding: A Versatile Graph Matching Algorithm. In: Proc. of the 18th Intl. Conf. on Data Engineering (ICDE). San Jose. CA. 2002

7 Doan A, Madhavan J, Domingos P, Halevy A. Learning to map between ontologies on the semantic web. In: Proc. of the World-Wide Web Conf. (WWW-2002), 2002

8 Do H H. Erhard Rahm: COMA-A System for Flexible Combination of Schema Matching Approaches. In: Proc. of the 28th Intl. Conf. ov Very Large Database, 2002. 610～621