

基于领域学习的本体映射方法

庞雄文 鲍苏苏

(华南师范大学计算机学院 广州 510300)

摘要 在语义数据集成中本体映射是关键,手工设置映射关系不仅消耗时间而且不准确,需要使用本体映射工具自动发现这种映射关系。在现有本体映射方法的基础上提出了基于领域学习的映射发现策略,可以从领域知识中发现复杂映射的规则,增加映射时的实例数据,从而提高映射发现的查全率和准确率;同时利用本体中丰富的语义知识去筛选候选数据,从而降低映射方法的算法复杂度。实验证明该方法是有效的。

关键词 本体,本体映射,领域学习

Domain Learning-based Ontology Mapping Method

PANG Xiong-wen BAO Su-su

(Department of Computer Science, South China Normal University, Guangzhou 510640, China)

Abstract Ontology mapping is the key problem of semantic data integration, discovering these mapping manually is inaccurate and time-consuming, using tools to automatically discover these mapping between source ontology and target ontology is necessary. We proposed a domain learning-based ontology mapping method which can use domain knowledge and historical mappings to augment the training data and to find the complex mapping rules about the ontology being matched, so they can be matched better. We also used the semantic information of ontology to optimize the mapping process in order to decrease the complexity of mapping method. The experiments show the correctness and efficiency of our solution.

Keywords Ontology, Ontology mapping, Domain learning

在信息集成领域,数据源之间的语义异构是数据集成的一个重要方面,解决这个问题的主要方案是使用(领域)本体来表达数据语义;由于数据源和本体是独立构建的,导致不同的本体、数据源表示相同的或者相互覆盖的业务领域,因此本体映射成为近年来的研究热点(同时本体映射也是语义 Web 研究的关键问题之一)。目前,国内外研究本体映射的有很多,所使用的具体方法也不同,但遵循同样的过程:以两个分别存在的独立本体(或者一个本体、一个数据源)作为输入,通过应用多种不同的映射策略计算两个元素之间(元素对之间)的相似度,然后选择相似度最大的元素对作为有映射关系的元素对;大多数研究主要集中在如何改进各种映射策略的效率、如何有效组合多种映射策略的计算结果成为元素对之间的最终结果,没有充分利用本体中丰富的语义、业务领域的业务知识和业务规则去加速、优化映射发现过程,导致映射发现过程不仅算法复杂度高,而且查全率和准确率不高(特别是针对具体的业务领域更是如此)。因此在映射发现过程中,如果能充分利用业务领域的业务知识和业务规则、利用本体中包含的丰富语义,不仅可以发现更多的映射关系的,提高映射发现的查全率和查准率;而且在映射发现过程中,可以利用这些知识在计算元素对之间的相似性的时候预先筛选掉那些不满足的元素对,这样可以减少需要计算的元素对数量,从而降低映射发现算法的复杂度。

1 本体映射研究现状

1.1 本体映射方法研究现状

本体映射主要是计算两个本体元素对之间的相似度,元素之间的相似度和本体定义、本体的实例以及本体的规则或约束相关,所以本体映射方法也分为与本体结构(定义)相关的方法、与实例相关的方法以及与领域限制相关的方法^[1,2]。

基于本体结构的映射方法从本体结构、本体元素两个层次上进行本体映射。在本体元素层次上,主要是比较本体元素的名称相似度来计算本体元素之间的相似度;在本体结构层次上,主要使用图相似方法和基于模型(model-set)的映射方法来进行映射发现。在图相似方法中,将源本体和目标本体看作是有向图,计算两个本体元素之间的相似性就是分析本体元素在图中的位置;这种方法的思想是如果两个本体元素相似,则它们的邻居节点也可能相似。基于模型的映射方法主要是利用本体所表示的语义来发现和验证映射结果。

基于实例的映射方法一般采用学习策略进行本体映射,使用本体中的实例数据作为训练数据和待分类数据,所以也可称为基于机器学习的映射方法。该方法的主要思想是将映射问题转换成分类问题,为某个概念选择最优映射的问题就转换成对其进行分类的问题。分类学习的方法通常利用一个本体中的信息学习分类模型,然后利用另一个本体中的信息

预测其每个元素可能的映射对象。基于机器学习的本体映射通常利用已有的机器学习方法,如使用支持向量机(Support Vector Machine SVM)、形式概念分析(Formal Concept Analyse, FCA)、贝叶斯学习(Bayes Learning)以及神经网络(Neural Networ, NN)等。

基于领域限制的映射方法一般不能作为单独的映射方法使用,主要作用是在映射发现过程中作为选择候选匹配对的依据之一。如领域限制中有“A与B不可能相等”,则在选择后项匹配对时,如果发现存在(A, B),该匹配对因为违反领域限制条件而被丢弃,不再进行相似度计算。

1.2 本体映射工具研究现状

目前的本体映射工具有很多,国内常见的工具包括清华大学的 RiMOM^[3],东南大学的 MASON^[4],浙江大学的 Dart-Mapping^[5]等,国外常见的工具包括 LSD^[6], GLUE^[7], MAFRA^[8], QOM^[9], PROMPT^[10], FCA-Merge^[11]等。

RiMOMR 基于贝叶斯决策理论提出最小风险的本体映射模型,将映射发现问题转换成风险最小化问题并提供了一个多策略的本体映射方法。MASON 是一种数据源-本体映射工具,它首先利用预先定义的转换规则,把关系数据库模式和本体分别转换到有向无环图,然后采用基于字符串比较的映射算法和基于结构的算法以寻找元素之间的简单对应关系。DartMapping 也是一种数据源-本体映射工具,它采用本地视图的方法,人工地把任意多个关系数据库映射到一个特定的通用的中医药本体上,最终以类似 datalog 语法的形式表达对应关系。

LSD 使用多策略学习方法发现本体映射,在 LSD 中学习器分为基本学习器(base Learned)和元学习器(meta learner);基本学习器(包括名称学习器、内容学习器、Bayes 学习器和 XML 学习器)使用多种方法计算本体之间的相似性,元学习器合并基本学习器的输出计算本体元素之间的相似度。

GLUE 通过多策略机器学习的方法半自动产生本体映射,通过计算本体元素实例之间的联合概率分布(Joint Probability)来计算本体元素之间的相似度;GLUE 使用朴素 Bayes 分类法计算概念之间的联合概述分布,然后采用松弛操作(relation labeling)产生满足条件的本体映射。

MAFRA 使用语义桥(semantic bridge)来表示本体之间的映射,MAFRA 的主要过程包括:(1)转换(标准化),将源本体和目标本体转换为相同格式;(2)相似度计算,计算概念和属性之间的相似度,计算过程采用多策略学习方法进行相似度计算;(3)产生语义桥,根据相似度计算的结果产生语义桥,语义桥采用语义桥本体(SBO)表示;(4)执行阶段,根据构建的语义桥将源本体中的实例转换为目标本体中的实例;(5)后处理阶段,对执行阶段返回的结果进行反馈。PROMPT 是一个半自动的本体映射和合并工具,使用基于元素名称和基于句法的映射方法产生建议的本体映射,并提供图形用户界面供用户修改和确认这些本体映射对结果进行解释;不同的是在 QOM 中,需要计算的元素对的搜索空间是可以迭代的,它使用动态编程方法进行搜索空间的迭代。

PROMPT 是一个半自动的本体映射和合并工具,使用基于元素名称和基于句法的映射方法产生建议的本体映射,并提供图形用户界面供用户修改和确认这些本体映射。

FCA-Merge 使用形式化概念分析技术产生每个本体的

形式化表示,通过词汇分析,当与领域相关的特定词汇存在于某个本体项目中就将该词汇或词汇的组合与特定的概念相联系。例如可以将 Holiday Hotel 与 Hotel 的概念联系起来。这一过程是半自动的,需要领域知识的介入。这一方法通过词汇分析建立一对一联系,但多对一、一对多联系还不能满足。

1.3 存在的主要问题

从以上对目前本体映射工具的分析可以看出,虽然存在很多本体映射工具(系统),但是本体映射还存在以下主要的问题:

(a)大多数工具只能处理属性之间的映射,对于如何根据属性之间的映射关系计算概念之间的映射关系却很少有工具涉及。

(b)大多数工具没有利用本体中丰富的语义。本体和模式数据最明显的差别就在于本体具有丰富的语义,很少有工具利用本体中的语义来加速、优化本体映射过程。

(c)大多数工具没有充分利用领域知识来加速、优化本体映射过程。

2 基于领域学习的本体映射方法

基于领域学习的本体映射过程如图 1 所示,包括以下几个主要步骤。

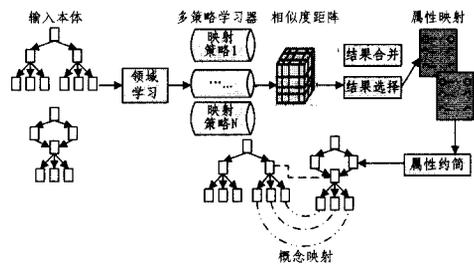


图 1 基于领域学习的本体映射过程

(1)领域学习。学习器首先从领域知识库中学习相关的知识,学习过程是通过源本体、目标本体和领域知识库之间的相关映射来完成的。这里的领域知识库不仅包括领域限制、业务规则,还包括用户以前映射过的相关本体、本体实例以及本体之间的映射。

(2)多策略学习器。多策略学习器是本体映射发现的基础;每一个映射策略独立预测元素对之间的相似性。假设源本体有 M 个元素,目标本体有 N 个元素,采用的映射发现策略有 K 个,则多策略学习器返回的结果是一个大小 $M * N * K$ 的相似度矩阵。

在本文中采用的多策略学习器包括基于名称的映射发现策略、基于结构的映射发现策略和基于实例的映射发现策略。

基于名称的映射发现策略主要是根据本体元素名称之间的词形相似度和词义相似度(使用 wordnet 作为词汇,通过计算两个单词在词典中的距离来计算词义相似度);基于结构的映射发现策略主要思想是两个概念结点的邻近结点是相似的,那么它们也趋向于相似,本文采用改进的相似度泛洪算法(SF)来计算本体元素在结构上的相似性;基于实例的映射发现策略主要是从实例的角度计算本体元素之间的相似性,文本类型的属性使用朴素贝叶斯分类器来计算相似度,数值类型的属性采用 KL 距离来计算相似度,分类属性(具有两个或

者两个以上属性值的属性)采用互信息来计算相似度。

在使用朴素贝叶斯分类器计算文本类型的属性相似度时,分类器的训练数据主要是本体自身,如本体中的某属性 s ,将 s 的所有实例数据作为“正”样本,在 s 的兄弟节点中(父类为概念)数据类型为文本的所有元素的实例作为“负”样本。从分类器的训练数据可以看出,分类器训练数据比较单一,不能完全刻画类别的内在特性,在本文中,通过领域学习可以大大扩展分类器的训练数据,提高分类器的准确性。

(3)根据多策略学习器的输出计算元素对之间的最终相似度。合并的原理是给每一个映射策略一个信任度,最终相似度是所有映射策略的输出该策略信任度之和。如果源本体的一个元素在目标本体中存在几个相似的元素,则使用结果选择算法选择满足条件的匹配对。该步骤的输出是直接属性映射。

(4)根据直接属性映射发现一对多映射和多对一映射。

(5)计算概念之间的映射,由于一个概念具有多个属性,因此要求概念的所有属性之间完全匹配是不可能的,所以只要概念的主要属性(进行属性约简后的属性)匹配,则可以认为概念之间是匹配的。

3 基于领域学习的映射优化

3.1 基于领域学习的映射优化

在基于贝叶斯分类器的文本分类中,相似度计算的准确程序依赖于训练数据,在没有领域知识的情况下,只能使用源本体中元素的实例作为训练数据,由于训练数据比较单一,影响了相似度计算的准确程度。在本文中,使用领域知识进行映射优化,领域知识对映射的优化主要表现在两个方面^[12]: (1)使用领域知识进行领域学习以增强分类器的性能。(2)使用领域知识的业务规则来优化映射选择。

定义 1(领域知识) 领域知识 DK 是一个四元组, $DK = \{O, H, M, R\}$, 其中 O 表示领域本体及其实例, H 表示和领域本体已经进行映射的所有本体以及实例(称为历史本体), M 表示领域本体及历史本体之间的映射关系, R 表示领域中的规则(可以是业务规则也可以是映射发现相关的规则)。

从领域知识的定义可以看出,我们在常规的领域知识的概念中增加了历史本体和映射,目的是使用这些已有映射关系和实例数据通过领域学习更加有效地进行映射发现。

定义 2(领域附加知识) 设 S 表示源本体, DK 表示领域知识,对 S 中任一元素 s , 领域附加知识(记为 AK) 满足: $\forall s \in S, AK = \{e | e \in DK \wedge e \cong s\}$ 。从定义可以看出,对 s 来说,领域附加知识是在领域知识中和 s 语义相等的元素及其实例。

定义 3(领域学习) 设 S 为源本体, T 为目标本体,对 S 中任一元素 s , ak 为 s 的领域附加知识, t 为 T 中的任一元素, $\{s, t\}$ 为需要计算相似度的元素对,在 DK 中寻找 ak 用于 $\{s, t\}$ 相似度计算的过程称为领域学习,即 $\{s, t\} \rightarrow \{(s, ak), t\}$ 。

领域学习过程如图 2 所示,其主要过程如下:

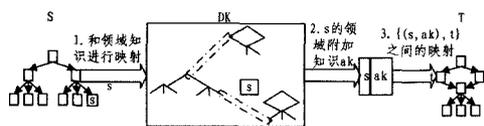


图 2 领域学习过程

对 S 中的任一元素 s ,通过前面介绍的映射发现机制在 DK 中发现和 s 相似的本体元素及其实例,组成 s 的附加领域知识。由于领域知识中包含了历史本体和领域本体之间以及历史本体之间的映射关系,因此只要在 S 和领域本体之间进行映射即可, s 和其他历史本体之间的映射关系通过映射的传递性自动计算。

s 的附加领域知识 ak 作为 s 的训练数据一起参与学习器的训练过程。如对文本分类的贝叶斯分类器来说,在没有附加知识的情况下,训练的正例数据是元素 s 的实例数据,训练的反例数据是概念中除 s 外其他元素的实例数据;在有领域附加知识的情况下,训练的正例数据包含了 s 和 ak ,反例数据可以包含领域知识中的反例数据。可见,在有领域附加知识的情况下,分类器的训练数据成倍增加,大大提高了分类器对 s 的刻画能力。

使用训练后的分类器计算元素对 $\{s, t\}$ 之间的相似度。由于领域知识中包含历史本体以及历史映射关系,因此当每一次本体映射结束后,可以将源本体和目标本体作为历史本体增加到领域知识中,这样领域知识不断增加,刻画分类器的能力也不断增强。

3.2 基于领域规则的映射优化

本文利用启发式规则来缩小元素对的搜索范围和进行映射选择,每个启发式规则由规则描述、信任系数以及违反规则时的惩罚系数组成。信任系数和惩罚系数将在映射选择时用到,它们可以手工设置,也可以由系统自动计算。

在本文中使用的启发式规则包括:

(1)元素只能在相容的数据类型之间进行映射。本文将常用数据类型分为文字型、数值类型以及日期型三类(数值类型包括整数型、浮点型等)。该规则用于缩小元素对的搜索范围;如源本体 S 中元素 s 数据类型为文字类,则目标本体中只有数据类型为文字型的元素 t 才能和 s 组成元素对。

(2)元素的 URI 相同,则元素相同。

(3)概念的属性全部相似,则概念相似。

(4)属性的定义域与值域相似,则属性相似。

(5)概念的父类以及直接子类相似,则概念相似。

(6)概念的兄弟节点全部相似,则概念相似。

(7)属性的父属性与直接子属性相似,则概念相似。

(8)概念(属性)的实例全部相似,则概念(属性)相似。

(9)如果实例对应的概念(属性)相似,则实例相似。

(10)两个实例通过相同的属性连接到另外一个实例,则这两个实例相同。

(11)本体语义中明确指定等价(不等价的),如 *equivalentclass*, *equivalentproperty*, *sames*, *alldifferent*, *disjointwith*, 该规则用于缩小元素对的搜索范围。

(12)使用 *allvaluefrom* 约束的属性,如果该属性值域约束相似,则属性相似。

(13)如果属性的势(cardinality)不同,则属性对的可信度降低。注意这条规则和前面规则不同,前面的规则是违反时进行处罚,这条规则是满足时进行处罚。

3.3 基于最小成本的映射选择

对源本体 S 中的任意一个元素 s ,在目标本体中存在多个元素(记为 t_i),如果 $\text{sim}(s, t_i) > \eta$ (其中 η 为最小相似度阈值,本文取 0.5)且没有满足条件的一对多映射,则需要从中

选择一个作为和 s 最匹配的元素,一般方法是选择相似度最大的元素作为匹配元素。本文采用基于最小成本的选择方法,映射的成本定义如以下公式表示:

$$Cost(M) = -\log \text{sim}(s, t) + \alpha_i \sum_{i=1}^K \beta_i K(i)$$

其中 M 表示 s 到 t 之间的直接映射,记为 $s \rightarrow t$, $\text{sim}(s, t)$ 为 s 和 t 之间的相似度, K 表示领域知识中启发式规则的个数, α_i 表示第 i 个规则的信任系数, β_i 表示第 i 个规则的惩罚系数, $K(j)$ 表示映射 M 违反第 i 个规则时的罚分。

定义了每个映射的成本后,映射的选择就变成在满足条件的映射中,选择成本最小的映射作为最后映射。在本文中为简单处理,每个规则的信任系数和惩罚系数手工确定,默认信任系数为 1,惩罚系数为 0.5。对映射违反规则的罚分也简单处理,当映射违反某一规则时,则 $K(i)=1$,否则 $K(i)=0$ 。

3.4 一对多映射的发现

对源本体 S 中的任一元素 s ,在目标本体 T 中存在多个元素(记为 t_i),如果满足以下条件:

(1) $\text{sim}(s, t_i) > \eta$ (其中 η 为最小相似度阈值,本文取 0.5)

且

(2) t_i 属于同一类数据类型(参见启发式规则 1)

(3) 元素 s 和元素 t_i 均有实例存在

则需要检查元素 s 是否在目标本体中存在一对多映射,即计算 $\text{sim}(s, F(t_1, t_2, \dots, t_n))$ 的相似度。确定一对多映射的关键是确定函数 f (即多个元素之间是如何组合在一起的)。本文从以下角度确定函数 F 的表达式:

(1) 基于表达同一领域时本体之间的相似性,首先从领域知识库的历史映射关系中包含的一对多映射作为函数 f 的首选表达式(如假设在历史映射关系中存在 $A = (B+C)/2$,则记函数 $F(x) = (y+z)/2$),根据该函数产生新的元素 t_{new} ,然后计算 $\text{sim}(s, t_{new})$ 。

(2) 使用固定函数计算新元素。在本文中提供的固定函数包括 `concat`(连接多个字符串,其中的字符串允许进行大小写转换、长度截取等)、`Add`(多个数字相加)、`subset`(多个数字相减)、`multiply`(多个数字相乘)。

(3) 自己定义函数格式。在本文中用户自定义的函数格式在匹配完成后会增加到领域知识库中,这样相同格式的一对多映射,用户只需要定义一次。

4 实验分析

在本文中,我们使用 `java` 和 `wordnet` 实现了基于领域学习的本体映射工具 `DLMMapping`。试验数据主要有两个来源:自己构建的采购订单本体、销售订单本体以及从 `EON2004` (2004 年本体映射国际竞赛)中抽取的 3 个本体(103, 201, 301),试验数据包含的类数量、属性数量、映射数量以及实例数量如表 1 所列。在分析试验数据时,主要是分析发现映射的查全率和准确率,查全率表示发现的映射数量和总映射数量的比例;准确率表示正确的映射数量和发现映射数量的比例。

表 1 实验数据

组号	本体名称	类个数	属性个数	映射数	实例数
1	采购订单 1	15	108	55	3000
	采购订单 2	20	163		3000
	销售订单 1	29	126		3000
2	销售订单 2	18	97	43	3000

3	参考本体	33	59	91	76
	103.00	34	61		111
4	参考本体	33	59	91	76
	201.00	34	62		111
5	参考本体	33	59	61	76
	301.00	15	40		0

使用基于领域学习的本体映射方法对 5 组本体数据进行本体映射,结果如表 2,表 3,图 3 和图 4 所示。

表 2 发现的映射数量

组号	映射数	无领域学习		有领域学习	
		查全率	准确数	查全率	准确数
1	55	46	43	53	51
2	43	35	33	38	37
3	91	84	82	88	86
4	91	78	73	81	76
5	61	46	45	49	46

表 3 算法的查全率和准确率

组号	映射数	无领域学习		有领域学习	
		查全率	准确率	查全率	准确率
1	55	83.64%	93.48%	96.36%	96.23%
2	43	81.40%	94.29%	88.37%	97.37%
3	91	92.31%	97.62%	96.70%	97.73%
4	91	85.71%	93.59%	89.01%	93.83%
5	61	75.41%	97.83%	80.33%	93.88%

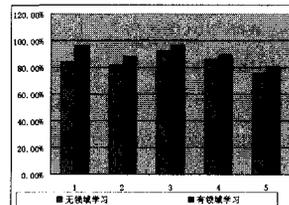


图 3 算法的查全率

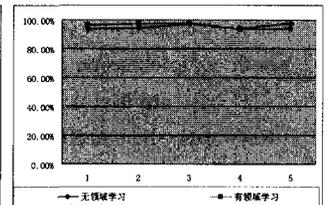


图 4 算法的准确率

从实验数据可以看出,使用基于领域学习的发现策略后,平均查全率提高了 6.46% (从 83.69% 到 90.16%),平均准确率也提高了 0.45%。经过分析发现,使用该策略主要是提高了文本类型的元素对之间的准确性,通过增加训练数据的数量从而提高了贝叶斯分类器的准确性;领域学习还根据领域知识中以前的映射知识发现了其他策略中有效发现的多对一和一对多映射。

结束语 本文提出了基于领域学习的本体映射策略,该方法可以从领域知识中发现复杂映射的规则,增加映射时的实例数据,从而提高映射发现的查全率和准确率;同时将本体中丰富的语义知识转化为启发式规则,通过这些规则来优化映射选择过程,从而降低算法的复杂度。根据本文提出的本体映射方法,实现了本体映射工具 `DLMMapping`。实验表明,本文提出的本体映射方法是有效的。

本文算法中使用的启发式规则是所有领域通用的规则,和特定领域相关的业务规则还需要手工输入,因此如何从映射历史数据中自动发现特定领域相关的业务规则,是本文下一步的工作重点。

参考文献

- [1] 瞿裕忠, 胡伟, 等. 关系数据库模式和本体间映射的研究综述 [J]. 计算机研究与发展, 2008, 45(2): 300-309

一步将尝试使用北京大学计算语言学研究所的中文概念词典(Chinese Concept Dictionary, CCD)进行义项影射。同 Word-Net 一样, CCD 使用同义词集表示义项, 不存在同义词词林中相关词集的问题; 或调整理解算法, 用最显著特征取代义项交集, 因为前者准确率更高。

表 6 隐喻理解结果

	特征交集	义项交集	最显著特征	总体
正确样本数	106	21	95	222
样本数	116	33	113	262
正确率	91.38%	63.64%	84.07%	84.73%

隐喻生成的准确率是 $205/262 = 78.24\%$ 。错误来源主要是返回的源域和要描述的目标域来自同一概念域, 或是形成的隐喻理解起来不够直观, 或语言表达上不够优美。还需要更多地考虑目标域的信息, 来提高隐喻生成的准确率。当然, 知识库的规模和质量也是影响隐喻处理结果的因素。因此, 知识库扩展和改进将是下一步主要工作。

目前汉语隐喻处理的研究还处于起步阶段, 尚无公共评测语料, 因此还无法与其他研究进行横向比较。但是本文实验结果表明, 文中提出的基于实例的方法是有效的。构建公共评测语料, 以评测推动方法的改进, 是隐喻研究的课题之一。

结束语 本文使用搜索引擎, 大规模获取明喻实例, 自动构建明喻知识库。在知识库的基础上, 考察了汉语源喻的使用情况, 提出了隐喻的自动理解和生成方法, 取得了较好的实验结果。

知识库还可以用于隐喻的识别。例如, “A+N”型隐喻(“红色的海洋”)的识别。从知识库中可以看到, “海洋”的显著特征之一是“蓝”, 而“红色”违反了组合限制, 因此是隐喻用法(可能是“红色的花的海洋”)。知识库中的明喻实例, 对于汉语学习、比喻学习等都是有帮助的。知识库中概念之间的组合关系, 还可以用于词义消歧、未知词语意思推断等自然语言处理任务。

下一步将进行知识库扩展。一是模式扩展, 采用模式“如 N 般 A”、“A 如 N”、“比 N 还 A”等, 获取更多实例。二是词类扩展, 通过模式“像 N 一样 B”、“像 N 一样 Z”、“像 N 一样 V”(像水一样流淌、像水一样蒸发)等, 引入区别词(B)、状态词

(Z)及动词(V)等。隐喻的表现形式多种多样, 最根本的形式是“X 是 Y”, 其他形式都由这一根本形式衍生而成。随着知识库的扩展, 将考虑处理其他形式的隐喻(如动词隐喻等)。

参考文献

- [1] Lakoff G. The contemporary theory of metaphor // Ortony A. Metaphor and thought. 2nd ed. Cambridge: Cambridge University Press, 1993: 202-251
- [2] Carbonell J G. Metaphor: an inescapable phenomenon in natural-language comprehension // Lehnert W, Ringle M. Strategies for natural language processing. Hillsdale, New Jersey: Lawrence Erlbaum Associates, 1982: 415-434
- [3] 俞士汶. 自然语言理解研究与文学表现手法 // 第二届文学与信息技术国际研讨会. 2005: 2-13
- [4] 王治敏. 汉语名词短语隐喻识别研究. 博士学位论文. 北京大学, 2006
- [5] Fass D. Met*: A method for discriminating metonymy and metaphor by computer. Computational Linguistics, 1991, 17(1), 49-90
- [6] Martin J H. A computational model of metaphor interpretation. Boston: Academic Press, 1990
- [7] Barnden J, et al. Reasoning in metaphor understanding: the ATT-Meta approach and system // Proceedings of COLING' 2002. 2002: 1-5
- [8] Kintsch W. Metaphor comprehension: A computational theory. Psychonomic Bulletin and Review, 2000, 7(2): 257-266
- [9] Abe K. A computational model for metaphor generation process // Proceedings of CogSci' 2006. 2006: 937-942
- [10] 苏畅, 周昌乐. 基于合作机制的汉语名词性隐喻理解方法. 计算机应用研究, 2007, 24(9): 67-69
- [11] 游维, 周昌乐. 基于统计的汉语隐喻生成模型及其系统实现. 心智与计算, 2007, 1: 133-141
- [12] Bowdle B, Gentner D. The career of metaphor. Psychological Review, 2005, 112(1): 193-216
- [13] Veale T, Hao Y F. Comprehending and generating apt metaphors: A web-driven, case-based approach to figurative language // Proceedings of AAAI' 2007. 2007: 1471-1476
- [14] 俞士汶, 朱学锋, 等. 现代汉语语法信息词典详解. 第二版. 北京: 清华大学出版社, 2003

(上接第 137 页)

- [2] Choi N, Song I-Y, et al. A Survey on Ontology Mapping[J]. SIGMOD Record, 2006, 35(3): 34-41
- [3] 唐杰, 梁邦勇, 李涓子. 语义 WEB 中的本体自动映射[J]. 计算机学报, 2006, 29(11): 1956-1976
- [4] Chen Huajun, Wu Zhaohui, Wang Heng, et al. RDF / RDFS-based Relational Database Integration[C] // Proc. of the 22nd Conf. on Data Engineering. Los Alamitos, CA: IEEE Computer Society Press, 2006
- [5] 郑东栋, 胡伟, 瞿裕忠. 一种关系数据库模式和本体间的匹配方法[C] // 第二届江苏计算机大会论文集. 南京: 东南大学出版社, 2006: 209-213
- [6] Doan A H, Domingos P, Halevy A. Learning to Match the Schemas of Data Sources: A Multistrategy Approach[J]. Machine Learning, 2003, 50 (3): 279-301
- [7] Doan A H, Madhavan J. Learning to map between ontologies on the semantic web [A] // Proceedings of the 11th International Conference on World Wide Web [C]. Hawaii, USA, 2002: 662-

- 673
- [8] Maedche A, Motik B, Silva N, et al. Mafra - A Mapping framework for distributed ontologies[A] // 13th European Conference on Knowledge Engineering and Knowledge Management [C]. Lyon, France, 2002: 235-250
- [9] Ehrig M, Staab S. QOM - Quick Ontology Mapping[A] // International Semantic Web Conference 2004[C]. 2004: 683-697
- [10] Noy N F, Musen M A. PROMPT: Algorithm and tool for automated ontology merging and alignment[A] // Proceedings of the 2000 National Conference on Artificial Intelligence[C]. Austin, Texas, 2000: 450-455
- [11] Stumme G, Madche A. FCA-Merge: bottom-up merging of ontologies[A] // Proc. of the Seventeenth Intl. Conf. on Artificial Intelligence (IJCAI '01)(C). Seattle, USA, 2001: 225-230
- [12] Madhavan J, Philip A, Halevy A. Corpus-based schema matching [A] // Proceedings of the International Conference on Data Engineering (ICDE)(C). 2005: 57-68