第37卷 第15期 计 算 机 工 程 2011年8月
Vol.37 No.15 Computer Engineering August 2011

·软件技术与数据库· 文章编号：1000—3428(2011)15—0043—03 文献标识码：A 中图分类号：TP391

基于扩展标记的改进本体概念分类算法

吕素刚，郑洪源

(南京航空航天大学信息科学与技术学院，南京 210016)

摘 要：研究 Pellet 系统本体概念分类算法及其优化技术，在此基础上给出一种基于扩展标记的改进算法。该算法通过概念间已知的包含关系，控制分类过程中遍历时概念加入的顺序，并最大程度地双向传播这些关系，从而有效地降低概念包含测试的次数。验证结果表明，该算法的概念分类性能平均提高约 22%。

关键词：本体；描述逻辑；概念包含测试；本体概念分类；扩展标记

Improved Ontology Concept Classification Algorithm Based on Extend Tag

LV Su-gang, ZHENG Hong-yuan

(College of Information Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)

【Abstract】Based on research of Pellet concept of ontology classification algorithm and some optimization techniques, this paper presents a method based on extend tag improvement. The method is mainly through the use of the inclusion relations concept between the known, taking control of the classification process of inclusion of the traversal order, and the maximum two-way dissemination of these relationships, thus effectively reducing the number of concept subsumption test. Algorithm verification results show that the refined algorithm improves the performance on average by about 22% when carrying out the concept classification.

【Key words】ontology; Description Logic(DL); concept subsumption test; ontology concept classification; extend tag

DOI: 10.3969/j.issn.1000-3428.2011.15.012

1 概述

本体(Ontology)，即以明确的、形式化的方法规定领域概念术语体系及其相互关系的可共享的描述[1]。本体对领域知识明确地定义为一种表述，统一了领域内的术语和概念，从而增强了知识共享、知识重用的程度。近年来本体受到信息科学领域的广泛关注，成为计算机科学中的一种重要方法，并且是语义 Web[2]、智能信息集成等领域的关键技术。

目前，定义本体主要采用基于描述逻辑(Description Logic, DL)的描述语言，特别是 W3C 推荐的 OWL 系列，其中又以 OWL DL 语言应用最多。如此一来，不但能够利用本体对领域知识进行合理的组织表达，构成领域本体，并且能够利用 DL 对领域本体进行有效的推理，推导出领域本体中的隐含知识。通常，在术语集 TBox 上的主要推理任务有一致性检测(Consistency Check)[3]、概念包含测试(Subsumption Test)等[4]，而后者主要用于本体概念分类。当前一些主流的本体推理机如 Pellet、Fact++等都实现了本体概念分类功能，但分类效率仍不尽人意，其中一个很重要的原因就是未能充分利用在推理过程中产生的概念节点间的关系，导致执行了很多无意义的概念包含测试。

针对上述不足，为了有效利用推理过程中产生的信息，以减少概念包含关系的测试次数，本文描述了一种称为基于扩展标记的本体概念分类优化方法，并以基于 Java 的推理机 Pellet 为基础，使用优化后的方法替代原有算法，然后利用一些标准本体进行测试，以验证本算法的有效性。

2 Pellet 概念分类算法分析

领域本体知识库(Knowledge Base, KB)中的术语(概念)集合一般具有分层结构，但分类信息并不完整，且含有大量的隐藏知识。因此，领域本体需要利用 DL 的推理服务重新计算知识库中的概念层次。

KB 中的概念分类过程就是计算 KB 中每一个概念的完整确切的包含关系，具体分类算法就是对 TBox 中的概念进行上位搜索和下位搜索，直到遍历所有概念后终止，从而建立起完整的概念层次关系。在 Pellet 推理机中，计算概念之间的包含关系采用基于 Tableau 算法[5]的包含测试推理，即利用以下方式将概念间的包含转化为可满足性问题：概念 C 包含概念 D 成立，即有 $D \subseteq C$，当且仅当 $D \cap \neg C$ 是不可满足的。用此方法测试一个概念相对于 TBox 是否满足复杂度很高，而概念之间的相互包含关系的计算，需要进行大量的可满足性判断，因此，概念分类的优化主要在于减少不必要的概念包含关系测试[6]。

为了提高 TBox 包含测试推理的效率，Pellet 系统采用了一些优化技术，如预处理优化、显示包含计算等[7]。

预处理优化技术就是利用一些等价规则将知识库中复杂的公理转化为一系列简单的公理，简化知识库的复杂程度，从而提高包含测试的效率。目前基于预处理优化的方法主要有 Lazy Unfolding、Backjumping 等技术。

显示包含计算是为了进一步减少概念包含测试，从概念的定义出发，找出那些明显的概念包含关系。显然，如果 $C = X \cap \cdots$，那么 $C \subseteq X$，X 称为 C 的 told subsumer。根据概念的定义，分析其结构，得到它的 told subsumer，要比通过概念可满足性的测试高效得多。

作者简介：吕素刚(1985－)，男，硕士研究生，主研方向：知识工程；郑洪源，副教授、博士

收稿日期：2011-01-12 **E-mail：**nuaalsg@gmail.com

通过上述介绍可以看出，Pellet 系统采用的是目前主流的概念分类算法，效率并不高。同时，对这些优化技术分析可知，虽然各种方法采用的手段不同，但却有着本质的共同点，那就是尽早地对推理树上的无效分枝进行剪枝，从而避免了对这些分支上的概念节点进行包含测试。

利用这些优化技术，可以有效降低概念包含测试次数。但是，由于概念包含测试的复杂性，本体概念分类仍然有着效率较低的问题。同时从对显示包含计算的分析中可以得出，概念间的已知关系对后续的分类有很大的帮助作用，受此启发，提出一种基于扩展标记的优化算法，充分利用分类过程中得出的概念间的包含关系，进一步提高本体概念分类效率。

3 Pellet 概念分类算法优化

基于扩展标记的概念分类优化方法主要利用概念间已知的包含关系，控制遍历时概念加入的顺序，最大程度地向上和向下传播这些关系，在后续的包含测试过程中以此为参考，从而有效地避免部分概念间的包含测试。

3.1 优化原理与方法

该方法的设计初衷是：在概念分类的过程中，为每一个概念节点创建一个标记散列表来存储其与知识库中其他概念之间的包含关系，并且动态更新所有相关概念的标记值。一旦找到已知的概念包含或执行了包含测试，力求把已确定的概念包含传播得最远。这就可能确定其他概念节点(不同于执行包含测试的概念)之间的包含关系，从而最大程度地减少进行包含测试的次数。该方法的主要改进依据是：对于知识库中的 2 个原子概念 C、D，在测试过程中如果得出结论 $C \supseteq D$，则有 C 的所有等价概念和父概念是 D 的所有等价概念和子概念的父概念；同样，如果有结论 $D \not\subset C$，C 的所有等价概念不是 D 的所有等价概念和子概念的父概念。

该方法充分利用了包含关系的传递性，将概念分类过程中包含测试的结果最大限度地传播，使后续的某些包含测试得以避免。具体来说，就是在构建知识库的概念分类层次过程中，创建每个概念节点时利用当前已知的概念包含信息初始化标记散列表，并在分类过程中结合了已知的概念包含和包含测试结果，通过更新所有相关概念标记值，将已经确定的包含关系传播到所有相关概念，从而避免一些不必要的包含测试，最终达到提高分类效率的目的。

3.2 Pellet 概念分类算法的改进

本文依据 Pellet 1.41，对其进行相关的扩展与改进。算法的主要步骤有：(1)扩展相关的数据结构。(2)加载本体到知识库。(3)计算概念预分类层次和拓扑排序。(4)计算概念节点的包含关系。

3.2.1 相关数据结构的扩展

在 Pellet 系统中，有 2 个和概念分类密切相关的类：TaxonomyNode 类和 Taxonomy 类，前者主要存储概念节点的一些包含关系信息，后者用来保存概念分类树的全局关系信息。为了存储概念分类过程中的一些有用信息，为这 2 个类分别增加一个散列表(Map)对象，对 TaxonomyNode 添加 Map 对象 relationMark，保存该节点与其他节点关系的标记信息；同样，对 Taxonomy 类进行扩展，添加的散列表用于存储概念分类树中的概念节点。

3.2.2 本体到知识库的加载

目前大多数领域本体都采用 OWL 语言描述，但 Pellet 系统不能直接对其进行处理，利用 HP 实验室开发的 Jena 工具包对领域本体进行解析，然后加载到 Pellet 系统的知识库中。此时知识库主要由 TBox、ABox 和 RBox 3 个部分组成。其中，TBox 存储概念和概念公理信息；ABox 存储个体和个体断言信息；RBox 则存储角色和角色公理信息。

概念分类的目的是计算知识库中概念的层次关系，所以计算过程中主要利用 TBox 中的信息。在此，给出概念分类的主体算法：

```
Begin
  Variable: T, preTaxonomy∈Taxonomy, List sortedNodes;
  preTaxonomy =preClassify(TBox);
  sortedNodes=topoSort(preTaxonomy);
  foreach t in sortedNodes do
    classify(T,t);
  end for;
  return T;
End
```

在上述算法中，首先利用知识库中的已知信息计算概念层次进行预分类，然后以此为依据对概念进行拓扑排序，最后依次计算各个概念节点的包含关系。

3.2.3 概念预分类层次的计算和拓扑排序

预分类主要是利用 TBox 中显示公理计算出一些有用信息。主要处理是将 TBox 中的公理迭代地进行演绎，最终细化为若干条 2 个原子概念之间的关系，然后将它们之间的关系在相应节点中显示标记，如 isSub、isSuper、Unknown 等。例如有公理 $D \subseteq C$，则在预分类层次中，概念 C 处于 D 的上层，并且将 D 及其等价概念添加到 C 及其等价节点的子概念列表中，同时将 C 及其等价概念添加到 D 及其等价节点的父概念列表中，如果 D 有子概念节点同样对其标记，C 若有父概念节点则标记之。将所有的公理都处理完之后，预分类层次中各节点间一些基本关系信息已经被标记了，从而可以避免已经具有明确包含关系的节点间的包含测试，在一定层度上提高分类算法的效率。

上述操作完成后，对结果集进行拓扑排序，得到一个关于概念节点的有序列表。这里的拓扑排序方法是：计算每一个概念节点的父概念列表的长度，长度越长，则该概念节点在列表中的位置就越靠后。一般来说，如果一个概念的父概念个数越多，那么该概念通常位于层次关系的底层；反之，则通常位于较高的层次。如此控制概念加入分类计算的顺序，通常能够有效地降低概念分类的包含测试次数。

3.2.4 概念节点包含关系的计算

依次对加入的概念节点进行包含关系测试，具体关系包括直接父概念、直接子概念以及等价概念等信息。分类算法主体如下：

```
Function classify (T, t)
Begin
  T.getNode(t).supers = getSuper(T, t);
  T.getNode(t).subs = getSub(T, t);
  T.getNode(t).instances = getInstance(T, t);
  T.getNode(t).equivalents = getEquivalent(T, t);
End
```

上述算法分别计算一个概念节点的父概念列表、子概念列表、概念的个体列表和等价概念列表，从而构成一个概念节点的完整包含关系信息。综合这些信息将得到整个知识库中概念节点的完整层次关系。

由于上面几种操作过程类似，下面仅以获取一个概念节点的父概念列表算法为例，在保留原有的优化技术基础上，给出一种基于扩展标记的优化方法。

算法的主题思想是，从概念分类树的根节点出发，在原有遍历搜索的过程中，参考概念节点间的已知关系信息并不断地更新这些信息，递归遍历直至结束。相应的算法流程如图 1 所示。

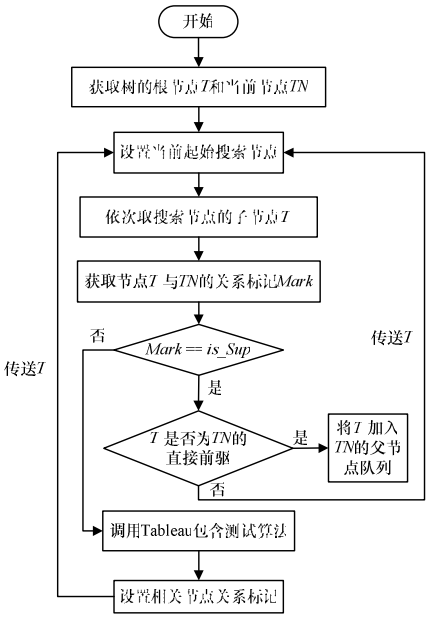

图 1　改进 Pellet 概念分类算法流程

从上述算法中可以看出，充分利用概念间的已知关系标记信息，可有效降低包含测试次数，并且这些信息在遍历过程中不断地得到更新，从而有效提高算法的工作效率。

4　算法验证与分析

和原有算法相比，改进后的算法由于加入了一些新的数据结构，会额外带来一定的开销，在空间复杂度上有所增加。但总体而言，额外开销并不大，现有的系统配置完全可以接受。为了验证改进后的算法在概念分类时的效率，从 W3C、DAML 等组织的网址上下载了一些常用测试本体进行实验，主要比较 2 种算法各自在进行概念分类时所消耗的时间。实验平台参数为：CPU P4 2.4 GHz，内存 512 MB，操作系统：Windows XP。2 种算法的测试结果如图 2 所示。

从图 2 可以看出，改进后的算法和此前相比效率普遍有所提高，且对较大的本体进行概念分类时提升效果更为显著，平均效率提高了约 22%。通过进一步分析可以得出，本体的复杂度以及本体的原始层次结构等因素对优化结果有很大影响。一般的，本体的复杂度越大，包含的概念越多，优化效果越明显；同样，本体的原始层次结构越平衡，优化后分类效率提升的幅度也越大。

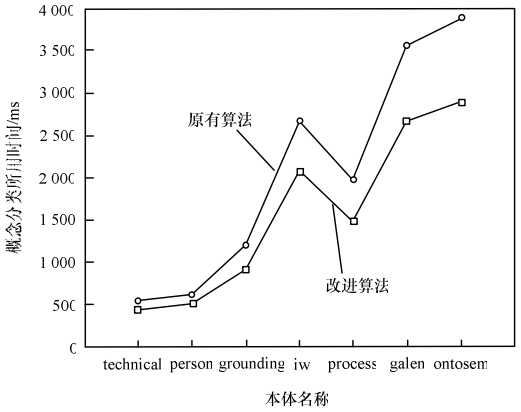

图 2　算法改进前后分类用时对比

5　结束语

本文在分析当前本体概念分类算法和一些优化技术的基础上，给出一种基于扩展标记的改进方法，经过验证，该方法能够较好地提升概念分类的效率。

目前对本体的研究已不再局限于语义网络等理论领域，特别是在基于描述逻辑的本体表达语言提出以后，可以充分利用描述逻辑的推理服务，基于领域本体构建相关领域的智能应用系统。这将是今后本体应用研究的一个重要方向。

参考文献

[1] 邓志鸿, 唐世渭, 张　铭, 等. Ontology 研究综述[J]. 北京大学学报: 自然科学版, 2002, 38(5): 730-738.

[2] 张志平, 杨建伟. 语义网技术及应用研究综述[J]. 情报学报, 2008, 27(5): 721-726.

[3] 许　勇, 王智学, 李宗勇. 领域本体的一致性检查[J]. 计算机工程, 2009, 35(1): 55-57.

[4] Baader F, Calvanese D, McGuinness D, et al. The Description Logic Handbook: Theory, Implementation and Applications[M]. Cambridge, UK: Cambridge University Press, 2003: 47-100.

[5] 刘　全, 孙吉贵, 于万筠. 基于 Tableau 的自动推理技术综述[J]. 计算机科学, 2005, 32(11): 1-4.

[6] 方　流. 描述逻辑推理优化技术研究[D]. 杭州: 浙江大学, 2008: 23-26.

[7] Tsarkov D, Horrocks I. Optimised Classification for Taxonomic Knowledge Bases[C]//Proc. of International Description Logic Workshop. Manchester, UK: [s. n.], 2005.

编辑　顾逸斐

(上接第 42 页)

参考文献

[1] 邓　貌, 鲁华祥, 金小贤. 基于特征分析的粒子群优化聚类算法[J]. 计算机工程, 2010, 36(8): 185-187.

[2] Chen Liang-Hwa, Chang Shyang. An Adaptive Conscientious Competitive Learning Algorithm and Its Applications[J]. Pattern Recognition, 1994, 27(12): 1787-1813.

[3] Guha S, Rastogi R, Shim K. CURE: An Efficient Clustering Algorithm for Large Databases[C]//Proc. of 1998 ACM-SIGMOD Int'l Conf. on Management of Data. Seattle, USA: ACM Press, 1998: 73-84.

[4] 雷小锋, 谢昆青, 林　帆, 等. 一种基于 K-Means 局部最优性的高效聚类算法[J]. 软件学报, 2008, 19(7): 1683-1692.

[5] Likhovidov V. Variational Approach to Unsupervised Learning Algorithms of Neural Networks[J]. Neural Networks, 1997, 10(2): 273-289.

[6] Blake C, Keogh E, Merz C J. UCI Repository of Machine Learning Databases[D]. Irvine, USA: University of California, 1998.

[7] Hall M, Frank E, Holmes G, et al. The WEKA Data Mining Software: An Update[J]. SIGKDD Explorations, 2009, 11(1): 10-18.

编辑　张正兴