OWL 中的概念相似度计算和应用*)

智东杰[1] 智慧来[2] 刘宗田[2]

(河南理工大学计算机科学与技术学院 焦作 454150)[1] (上海大学计算机工程与科学学院 上海 200072)[2]

摘 要 为了提高信息检索的查全率和查准率,经常要处理相似的概念,因此计算概念间的相似度是必要的。OWL 中的概念有如下两个特点:概念按照继承关系形成了一个层次结构的分类树,子概念继承父概念的属性;概念的属性由断言条件定义。概念相似度计算分为两步:首先,递归计算相同属性的数量;然后,根据相同属性的数量计算概念相似度。计算模型利用了 OWL 的较为完整的信息,计算结果与人为判断基本吻合。

关键词 本体,概念相似度,属性,深度

Concept Similarity Calculation and Application in OWL

ZHI Dong-jie[1] ZHI Hui-lai[2] LIU Zong-tian[2]

(School of Computer Science and Technology, He'nan Polytechnic University, Jiaozuo 454150, China)[1]

(School of Computer Engineering and Science, Shanghai University, Shanghai 200072, China)[2]

Abstract In order to improve the recall and precision, similar concepts are often processed during the information retrieval. With this regard, the possibility of evaluating concept similarity is acquiring an increasing relevance, since it allows the identification of different concepts that are semantically close. Concepts in OWL have two features. Firstly, as son concepts inherit their father concepts' attributes, so all concepts form a hierarchical taxonomy tree. Secondly, concept's attributes are defined by asserted conditions including the ones inherit their father concept's. The computing model also has two parts. First part is using recursive algorithm to compare concepts' attribute similarity. The second part is to calculate concept similarity by using the amount of same attributes.

Keywords Ontology, Concept similarity, Attribute, Depth

目前,领域本体的应用越来越引起人们的重视,成为一个具有美好前景的研究领域。文献[1]论述了基于 OWL-S 的语义检索。文献[2]论述了采用 OWL 语言工具实现基于本体的语义匹配。在基于本体的应用中,都涉及相似概念的处理,概念相似性的研究越来越引起人们的关注。

近一两年内,国内外有一些学者对概念相似性的度量进行了较为深入的研究。文献[3]利用概念间的距离和最近根概念深度来计算概念相似度,并设计了基于语义相似度的信息检索方法。文献[4]中具体给出两种词汇语义相似度计算方法,其中第一种方法计算词汇语义相似度基于词语间距离度量,第二种方法计算词汇语义相似度则建立在义原相似度基础上。文献[5]利用形式概念分析来计算概念的相似度,但忽略了概念的深度对相似度的影响。

1 OWL 和 Protégé

本体是源自哲学上的一个概念,用于描述事物的本质。1991 年,Neches 等人指出:"一个本体定义了组成主体领域的词汇的基本术语和关系,以及用于组合术语和关系一起定义词汇的外延的规则"[6]。几经锤炼之后,Studer 等提出的"共享概念模型的明确的形式化规范说明"[7]成为了人工智能领域公认的对本体的定义。

语义网中,本体发挥着重要的作用。Web 本体语言 OWL 可以用来定义和实例化 Web 本体。具体地,OWL 不仅可以清晰表达概念的含义,而且可以描述概念间的关系。和 XML,XML-Schema,RDF,RDF-Schema 相比,OWL 有更加丰富的语言来描述类和属性,包括类之间的关系的定义(e. g. disjoint)、基数(cardinality)的定义(e. g. "exactly one")、相等(equality)的定义、更加丰富的属性类型、丰富的属性特征(e. g. symmetry)、枚举类(enumerated classes)的定义[8]。

一个基于 OWL 的本体可以用一个附带 OWL plugin 的 Protégé 开发。Protégé 是一个开放的本体建模和知识获取平台,其 OWL 表达式中包括的语法符号如表 1 所示。

表 1 Protégé 中的逻辑符号

OWL element	symbol
Owl:allValuesFrom	∀
Owl:someValuesFrom	∃
Owl:hasValue	∋
Owl:minCardinality	≥
Owl:maxCardinality	≤
Owl:cadinality	=
Owl:intersectionOf	∩
Owl:unionOf	∩
Owl:oneOf	{}

2 OWL 中的概念相似度计算

文献[2]研究了基于本体的语义匹配,介绍了概念差异性

*)基金项目:国家自然科学基金(项目批准号 60575035)。**智东杰** 研究员,主要研究方向为人工智能、符号计算;**智慧来** 博士,主要研究领域为信息处理、概念格、本体;**刘宗田** 硕士,教授,博士生导师,主要研究领域为人工智能、软件工程和形式概念分析。

的计算方法。如果两个概念存在差异，则判定两个概念不匹配。本文研究的是如何在 OWL 中计算两个概念的相似性，应用概念的相似性来优化查询。

2.1 概念相似度计算的基本观点

概念相似度可以由概念在 OWL 本体层次结构中的距离来度量。直观上可以看出：距离越大，其相似度越低；反之，两个概念距离越小，其相似程度越大。距离为 0 时，其相似度为 1；词语距离为无穷大时，其相似度为 0；相似度为词语距离的单调递减函数。

概念所处的节点的深度对相似度计算也有影响，同样距离的两个概念，概念相似度随着它们所处层次的总和的增加而增加，随着它们之间层次差的增加而减小。因为，层次总和的增加意味着分类趋向细致，与同样概念距离的层次总和较小的概念对比较，其相似程度就越高。

2.2 概念相似度计算

OWL 本体中的概念由断言条件来定义，断言条件分为继承的断言条件和非继承的断言条件两种。继承的断言条件继承其父概念的所有属性。计算概念的相似性就要比较概念定义的断言条件，如果断言条件中包括其它概念，那么还要计算这些概念的相似性。

计算概念相似度的算法有 4 个步骤：

(1)依据概念间的继承关系，对两个概念分别收集从这两个概念到根结点概念的断言条件；

(2)比较两个概念的所有断言条件：断言条件不包含概念的比较，转 3；否则，转 4；

(3)如果两个断言条件相同，相同增加概念的相似度；否则，相似度不变；

(4)如果断言中包含的概念的关系为 disjiont，则置相似度为 0，返回；否则，递归计算所包含概念的相似度，并按照这条断言的比重处理后累计相似度。

计算概念相似度的算法进一步求精如下：

```
Global m;/*定义全局变量记录 c1 和 c2 的最大断言条件数目*/
int Calculate-SameProperty(concept c1;concept c2)
{
  int sameProperty=0;/*记录相同的断言条件数目*/
  p=c1;q=c2;
    while (p. father! =SUP) /*SUP 为根结点概念*/
  {
    p. restriction union p. father. assertCondition;
    p piont at p. father;
    }/*收集 c1 的所有断言条件*/
    while (q. father! =SUP)/*SUP 为根结点概念*/
    {
      q. restriction union q. father. assertCondition;
      q piont at q. father;
    }/*收集 c2 的所有断言条件*/
    m=max(|p. restriction|,|q. restriction|);/*取 c1、c2 的断言条件数目的最大值*/
      for every property p in p. restriction
    { /*对每一条断言进行处理*/
    if restriction is (∋ p d1)
      then if there exist (∋ p d1) in q. restriction
    sameProperty=sameProperty+1;/*累计相同的断言条件数目，第 1 条判据*/
    if restriction is (≥p x)
  then if there exist (≥p y) in q. restriction and (y≥x)
      or there exist (= p x) in q. restriction
      or number of all filter in q. restriction more than x
      sameProperty=sameProperty+1;/* 第 2 条判据*/
    if restriction is (≤p x)
  then if there exist (≤p x) in q. restriction and (y≤x)
      or there exist (=p x) in q. restriction
      or number of all filter in q. restriction less than x
      sameProperty=sameProperty+1;/* 第 3 条判据*/
    if restriction is (= p x)
      then if there exit (=p x) in q. restriction
      or number of all filter in q. restriction equal x
      sameProperty=sameProperty+1;/*第 4 条判据*/
    if restriction is (∀p c) or (∃p c)
      then if there exist (∀p d)or (∃p d)
      if c disjoint d
      then return 0;
      /*若 c 和 d 关系为 disjoint，则可以判断 c1 和 c2 一点也不相关，返回 0*/
        else then
          sameProperty=sameProperty+ Calculate-SameProperty(c,d)/m;
    /*递归计算 c,d 的相同断言数目，并按照这条断言的比重处理后累加到 sameProperty 上，第 5 条判据*/
    }
  }
  float Calculate-Similarity(concept c1;concept c2)/*调用 Calculate-SameProperty 计算相似度*/
  {
    return Calculate-SameProperty(concept c1;concept c2)/m;}
```

2.3 计算实例

为了与文献[2]进行对比研究，这里采用文献[2]中的关于“数据”的本体，如图 1 表示。基本 DatatypeSet 中的各个数据类型是相互不关联的，即具有“disjoint”的属性定义。

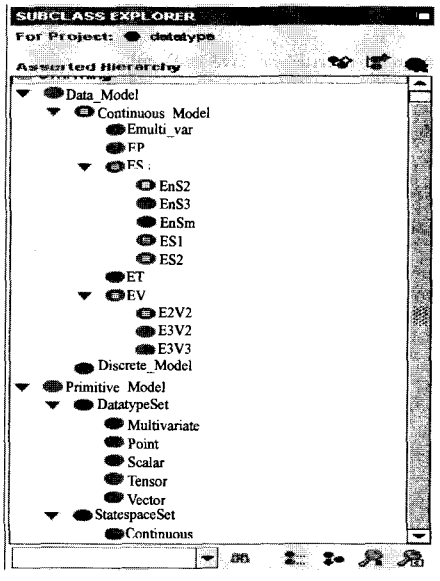

图 1 “数据”本体

①E2V2 的断言条件为 {has-Dimension = 2, has-Vector =2},

②ES2 的断言条件为{has-Dimension = 2, has-Scalar = 1},

③ES1 的断言条件为{has-Dimension = 1,has-Scalar = 1},

④ES 的断言条件为{has-DatatypeSet Scalar};

⑤EV 的断言条件为{has-DatatypeSet Vector};

⑥EnS2 的断言条件为{has-Dimension = 2,has-ScalarP 1};

⑦Continuous_Model 的断言条件为{ has-StatespaceSet Continuous}。

(1)计算 E2V2 和 ES2 相似度

E2V2 继承了父概念 EV 的断言条件 has-DatatypeSet Vector,ES2 继承了父概念 EV 的断言条件 has-DatatypeSet Vector,has-DatatypeSet Scalar,根据第 5 条判据,递归计算 Vector 和 Scalar 的相似度,因为 Vector"disjoint"Scalar,所以 Calculate-Similarity(E2V2,ES2)=0。

(2)计算 ES1 和 Continuous_Model 相似度

ES1 通过继承,加之本身的断言条件,具有的断言条件为{has-Dimension = 1, has-Scalar = 1} ∪ {has-DatatypeSet Scalar} ∪ { has-StatespaceSet Continuous}

ES1 和 Continuous_Model 相同的断言条件只有{ has-StatespaceSet Continuous},所以 Calculate-Similarity(E2V2, ES2)=1/4=0.25。

(3)计算 ES2 和 EnS2 相似度

Calculate-Similarity(E2V2,ES2)=1

在本体的层次结构中,可以用函数 Calculate-Similarity 计算任意两个概念的相似度,这里不再举例。

3 应用与结论

在信息检索的实现过程中,为了提高查全率,如果按照查询关键字进行检索得不到相关信息,那么就对查询关键字进行泛化和平级扩展操作[9],用其相似概念进行查询。

检索算法描述如下:

```
Result Retrieve(Key)
{
  Find relative data and save to Result;
  if Result! =Null
  then return Result;
  else
  { New_Key=Find_Parent_Concept(Key);
  if Calculate-Similarity (New_Key,Key)>ε/ * 计算相似度 * /
  then
  {Key=New_Key;
   Retrieve(Key);} / * 递归调用 * /
  };/ * 泛化操作 * /
  {New_Key=Find_Sibling_Concept(Key);
  if Calculate-Similarity (New_Key,Key)>ε / * 计算相似度 * /
  then
  {Key=New_Key;
   Retrieve(Key);} / * 递归调用 * /
  };/ * 平级扩展操作 * /
}
```

结合笔者的研究领域,建立了一个关于"本体"的领域本体,包括本体、事件、主题、基于事件的本体、领域本体等多个概念形成的形式背景。结合上述算法并结合设计了一个文献检索的工具,并在从"中国知网"下载的关于本体的 2463 篇文献中进行实验。实验结果表明提高了检索的查全率。例如:没有应用算法 Retrieve,输入"事件本体"进行文献检索,没有文献命中,应用算法后,返回了 14 篇文献,其中有 3 篇与我们所要研究的"事件本体"不相关。

实验表明,本文的计算模型充分考虑到了距离对概念相似度的影响,并利用 OWL 中本体结构的较为完整的信息,计算模型简单明了,计算结果能够满足实际应用的要求。

参 考 文 献

[1] Naveen S, Massimo P, Sycara K P. An efficient algorithm for OWL-S based semantic search in UDDI // SWSWPC. 2004:96-110

[2] Shu Gao, Rana O F, Avis N J, et al. Ontology-based semantic matchmaking approach. Advances in Engineering Software[J], 2007,38:59-67

[3] 王进,陈恩红,施德明,等. 一种基于语义相似度的信息检索方法. 模式识别与人工智能[J],2006,12(6):696-701

[4] 吴健,吴朝晖,李莹,等. 基于本体论和词汇语义相似度的 Web 服务发现. 计算机学报[J],2005,4(4):595-602

[5] Formica A. Ontology-based Concept Similarity in Formal Concept Analysis. Information Science[J],2006(176):2624-2641

[6] Neches R, Fikes R, Finin T, et al. Enabling Technology for Knowledge Sharing[J]. AI Magazine,1991,12(3):36-56

[7] Gruber T R. A translation approach to portable ontologies[J]. Knowledge Acquisition,1993,5(2):199-220

[8] OWL Web Ontology Language Overview. http://www.w3.org/TR/2004/REC-owl-features-20040210/

[9] 曹锐,陈刚,蔡铭. 基于本体的网络化制造资源检索. 计算机工程[J],2004,2(3):143-146