基于多个领域本体的文本层次被定义聚类方法

张爱琦 左万利 王 英 梁 浩
(吉林大学计算机科学与技术学院 长春 130012)

摘 要 传统的聚类方法常常将文本中关键词的相似度作为聚类的依据,丢失了很多重要的语义信息,导致聚类结果不够准确且计算量大。提出了一种基于多个领域本体的文本层次聚类方法,利用多个领域本体将用关键词表示的文本特征向量表示为与之匹配的概念向量集,定义文本相似度的计算公式,设计并实现基于多个领域本体的文本凝聚聚类算法。实验结果表明,该方法从概念层次上表示和处理文本,降低了聚类对象空间的维度,减少了计算量,提高了文本聚类的精确度和聚类效率。

关键词 领域本体,相似度计算,凝聚层次聚类
中图法分类号 TP391 **文献标识码** A

Text Hierarchical Clustering Based on Several Domain Ontologies

ZHANG Ai-qi ZUO Wan-li WANG Ying LIANG Hao
(College of Computer Science and Technology, Jilin University, Changchun 130012, China)

Abstract Traditional clustering methods are usually based on the similarity of keywords appearing in documents. Since these methods may lead to the loss of lots of semantic information, their clustering results are not accurate enough and often need large amount of computation. A new method for hierarchically clustering documents based on several domain ontologies was proposed. This method first transformed keyword-based vectors into corresponding concept-based vectors making use of several domain ontologies. Then, a formula was given for calculating similarities between different documents. An algorithm for document clustering based on several domain ontologies was proposed and its corresponding prime system was also designed and implemented. The experimental results show that our method can express and process documents from the perspective of concept semantics. It can decrease the amount of computation by reducing the dimension of the space of clustered objects and improve both the accuracy and the efficiency of document clustering.

Keywords Domain ontology, Similarity computing, Agglomerate hierarchical clustering

随着Internet和Intranet的飞速发展,各种电子文本数据急剧增加,对其快速、有效地获取、管理和使用已经成为信息科学和技术领域的一大挑战。文本聚类技术能自动对文本信息进行处理,将大量信息按照特征无监督地进行分类,使用户从繁琐的文档处理工作中解放出来,极大地提高了信息处理的准确率。文本聚类技术作为文本信息处理的重要技术之一,得到了相关研究者的广泛关注。

文本聚类方法按数据的积聚规则通常分为层次法[1]、划分方法和基于模型的方法等。层次法将数据对象在不同阶段组成不同粒度的簇,在簇的凝聚和分裂过程中不断改善聚类的效果,达到逐步求精的目的。根据层次的分解如何形成,可以分为凝聚和分裂两大类。文本聚类方法通常以词作为基本特征来表示文本,然而由于词空间的高维性、稀疏性和相关性等特点,聚类质量和效率并不能令人满意[2],因此人们希望能够使用词的语义能力来表示文本。Agirre等人和Dario等人试图用英文词典WordNet[3]来解决词的歧义性[4,5],Sopan等人讨论了通过自动扩充词的语义来表示文本以提高Web资源的语义分类[6],Sedding等人和Hotho等人研究了使用WordNet进行文本聚类[7,8]。这些研究都是基于WordNet的。WordNet也可看作是通用本体,能表示出词语之间的近义和反义等语义,但通常会忽略概念间的关系,而领域本体却可以弥补这一缺陷。

本体是共享概念模型的形式化规范说明,具有良好的语义结构,能清晰地表明概念间的语义关系。领域本体是用于描述特定领域知识的一种专业性本体,可清晰地刻画出专业领域内部特征概念间的语义关系。将领域本体用于文本聚类方法中,从概念层次上表示和处理文本,将使文本聚类更加准确和高效。目前,德国卡尔斯鲁厄大学的Hotho等人将特定

到稿日期:2009-04-03 返修日期:2009-07-30 本文受国家自然科学基金(No. 60373099, 60603031),国家教育部高等学校博士学科点专项科研基金(No. 20060183044, 200801830021),吉林省自然科学基金(No. 20070533),吉林大学基本科研业务费交叉学科与创新项目(No. 200810025)资助。

张爱琦(1986—),女,硕士生,主要研究领域为Web挖掘、本体工程、网络搜索引擎,E-mail:zhangaq163@163.com;左万利(1957—),男,教授,博士生导师,主要研究领域为数据库、数据挖掘、网络搜索引擎;王 英(1981—),女,博士生,主要研究领域为Web挖掘、网络搜索引擎、本体;梁 浩(1980—),男,博士生,主要研究领域为数据挖掘、Web智能、本体工程和信息融合。

领域的本体用于 K-Means 聚类方法中，较好地实现了文本聚类[9]，但该方法局限于单一领域内，无法实现多领域文本的聚类。谢红薇、余雪丽等人研究了将大学学校领域本体用于 K-Means 聚类方法，成功地实现了 Web 页面聚类[10]，但在进行文本特征转换时将无法转换为本体概念的特征直接丢弃，聚类结果不够准确。

1 本体概念间的相似度

本体的主要构成元素有类、属性和类的实例。类，即概念，代表一些对象的集合。属性既可以是对类所具有的特性的描述，也可以表示两个类的实例之间的关系。实例代表了我们在某个领域中感兴趣的对象。

下面介绍计算本体概念相似度时需要考虑到的因素。

本体中概念间的相似度计算的基础是：两个概念具有一定的语义相关性，且它们在概念间的结构层次网络图中存在一条通路。一般情况下根据概念之间的上下位关系、语义重合度、语义距离和层次深度等多种因素来计算领域内部概念之间的语义相似度。这些因素的定义如下：

定义 1(语义重合度)[11] 两个概念之间的语义重合度是指本体内部概念之间包含相同上位概念的个数。若用 $p(x)$ 代表从 x 节点上溯到根节点的节点集合，则用 $|p(x)\cap p(y)|$ 表示 x 和 y 两个概念之间的语义重合度。

语义重合度表明了两个概念间的相同程度，在实际计算中可以转化为公共节点的个数。

定义 2(语义距离)[11] 两个概念之间的语义距离是指在本体图中连接这两个概念节点的通路中最短路径的边数。用 $\mathrm{Dis}(x,y)$ 表示概念 x 与 y 之间的语义距离。一个词语与其本身的距离为 0。

语义距离是决定概念间相似度的一个基本因素。一般而言，两个词的距离越大，其相似度越低；反之，两个词的距离越小，其相似度越高。二者之间可以建立一种简单的对应关系。这种对应关系需要满足以下 3 个条件：两个词语距离为 0 时，其相似度为 1；两个词语距离为无穷大时，其相似度为 0；两个词语的距离越大，其相似度越小(单调下降)。

定义 3(层次深度)[11] 概念的层次深度是指概念在本体概念图中的深度，用 $h(x)$ 表示概念 x 的层次深度。

同样距离的两个词，词的相似度随着它们之间层次深度的差的增加而减小。计算不在同一深度上的两个概念间的相似度时，概念间的层次深度差可以基本上反映相似度。随着深度差的增加，概念间的相似度降低。

2 文本凝聚层次聚类框架

本节提出了基于多个领域本体的文本凝聚层次聚类框架，如图 1 所示。

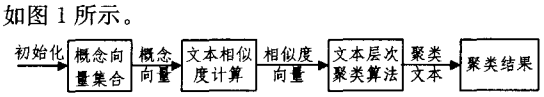

图 1 文本层次聚类框架

首先初始化时需要输入关键词向量集合、词频集合和本体存储路径集合；概念向量集合的获取是根据多个领域本体，将关键词向量集合转换成与之相匹配的概念向量集合，该概念向量集合将包括两部分：本体概念和未转换的关键词，本文将未转换的关键词称为未登录词；文本相似度由两部分组成：概念向量集合中的概念文本相似度和概念集合中未登录词文本相似度，这两部分分别乘以权重调节因子再求和，就获得了最终的文本相似度；最后将文本相似度作为凝聚层次聚类的依据，实现文本聚类。

下面详细描述该框架的每个步骤的实现技术。

3 概念向量集合的获取

本文假设已经对文本集合进行了预处理，获得了能代表文本的关键词特征向量集合 $VK=\{K_1,K_2,\cdots,K_m\}$，其中关键词向量表示为 $K_d=\{k_1,k_2,\cdots,k_n\}(d=1,2,\cdots,m)$；与关键词特征向量集合对应的词频向量集合 $WQ=\{WQ_1,WQ_2,\cdots,WQ_m\}$，其中词频向量表示为 $WQ_d=\{wq_1,wq_2,\cdots,wq_n\}(d=1,2,\cdots,m)$。

本节提出将关键词向量集合转换成与之相匹配的概念向量集合的转换方法。转换过程算法描述如下。

算法 1 关键词向量到概念向量的转换

输入：某文本的关键词向量 K，对应词频集合 WQ，本体存储路径集合 Op，词频阈值 fq

输出：转换后的概念向量 C，与转换后向量对应的词频集合 W

```
1. Initialize() //初始化各变量及内存空间
2. for each k in K
     cn=Match(k,C,Op) //关键词与概念进行匹配
     if(cn! =null)
        C.add(cn)
        else if(k.wq >fq) //保留出现次数较频繁的未登录词
        C.add(k)
3. for (int i=0;i<C.length-1;i++)
     for (int j= i;j<C.length;j++)
       if(C.get(i)==C.get(j) ) //合并转换后集合中的相同概念
           TidyCW(C,WQ,i,j) //整理概念集合和词频集合
```

该转换过程应用领域本体，采用相应的匹配算法 Match() 使关键词与本体中概念进行匹配，然后将匹配的关键词用相应的概念进行替代：当不存在与关键词匹配的概念时，考虑该关键词在文本中出现的次数，如果大于规定的阈值，就保留该关键词，否则舍弃；最后合并转换后集合中的相同概念：如果在转换后的集合中存在两个相同的概念，则将概念集合中的一个概念去掉，将两个概念的词频相加，其中存入词频集合中保留概念的相应位置。使用上述方法实现文本的关键词集到概念集合的转换，可最大限度地保留文本信息。

其中关键词与概念的匹配函数 Match() 的算法描述如下。

算法 2 关键词与概念的匹配

输入：需要判断是否匹配的关键词 k，文本的概念向量 C，本体存储路径集合 Op

输出：转换后的概念 c

```
1. Initialize() //初始化各变量及内存空间
2. for each op in Op
     model= Load(op) //model 存储当前载入的本体信息
     for each cn in model.class //遍历本体中所有类(概念)
        if (cn==k)
           cal.add(cn)
     if(cal.length ==0) //如果没有找到相匹配的概念则遍历本体中所有属性
```

```
    for each pro in model.property
        if (pro==k)
            cal_cp=FindMatchCon(pro) //找到属性 pro 所属的概念,注:这样的概念可能有多个
            for each cn in cal_cp
                cal.add(cn)
    if (cal.length ==g0) //如果上面两步均没有找到相匹配的概念则遍历本体中所有实例
        for each pro in model.instance
            if (ins==k)
                cn=FindMatchCon(ins) //找到实例 ins 所属的概念
                cal.add(cn)
    if (cal.length ! = 0)
        cal_cy.add(MinHighCon(cal)) //保存最低下位概念
    else return null;
3. if (cal_cy.length ! =0&& cal_cp.length ! =1)
     opt= FindOp(C) //找到包含已匹配的概念最多的那个本体
4. c= FindCon(cal_op,opt) //找到概念集合 cal_cp 中存在于路径为 opt 的本体中的那个概念
```

该匹配过程首先将关键词与本体中的概念相比较,如果相同,则返回本体中的概念;否则将关键词与本体中的所有属性相比较,如果相同,则返回该属性所属的最低下位概念,即概念集合中层次深度最大的概念;否则将关键词与本体中的所有实例相比较,如果相同,则返回该实例所属概念。由于是与多个领域本体进行匹配,不同领域本体可能包含有相同的概念,因此通过上面匹配过程找到的概念可能出现重复概念。为了解决该问题,对已匹配的概念使用本体存储路径标识其所属本体,对于获得的待匹配本体概念集合,保留当前标识最多的本体中的概念。如果关键词经过上面 3 种比较仍找不到与之相匹配的本体概念,则返回 null。

通过上面的概念获取过程,可以获得概念文本集合 $VC=\{C_1,C_2,\cdots,C_m\}$,即将一篇文本表示成由若干个概念所构成的集合 $C_d=\{c_1,c_2,\cdots,c_n\}(d=1,2,\cdots,m)$,其中 c_i 表示概念或未登录词。每一项 $c_i(i=1,2,\cdots,n)$表示一个特征项,由这些项构成了一个向量空间,每个项表示一个维度,因此这个概念集合又可称为概念向量。

该转换过程具有如下优点:1)将关键词空间映射到概念空间时,同义词会映射为同一个概念,这样可以大大降低聚类对象空间的维度,减少计算量并提高聚类的准确度。此外,将具有同义关系的关键词映射到一个概念,可以避免一个重要的分类特征因为采用关键词的分散而削弱其权重;2)将关键词空间映射到概念空间后可以通过计算本体中概念的语义距离来计算相似度,节省了聚类方法中计算相似度的时间,在时间效率上优于基于关键词的聚类;3)基于关键词的算法假设关键词之间是相互独立的,而关键词之间事实上存在复杂的同义、多义、相关和互斥等关系,将关键词映射到概念空间可以从一定程度上消除这些关系或者将这些关系转换成本体中的语义关系,既提高了聚类的准确度又降低了复杂度;4)将关键词集合通过多个领域本体与多个领域中的概念相匹配,既考虑到文本中的特定领域特征也考虑到文本中的混合领域特征,比使用单一领域本体的匹配过程更加完全和准确;5)将无法与本体中概念相匹配但出现次数较多的关键词保留下来,可以充分使用文本的特征信息,有效提高文本聚类的效果。

4 相似度计算

将文本关键词向量转换为概念向量后,需要计算新得到的文本向量集合中文本间的相似度,这个相似度将作为下一步文本聚类的依据。

4.1 概念间语义相似度的计算

本文以本体知识库作为提取词语语义的基础,因此同一本体中两个概念的相似度主要是通过知识本体中的语义关系来计算的,它更多地体现了领域专家对概念类别的划分,而不是单纯依赖概念结构图形式上的相似性。由于专家所划分的领域概念体系所具有的权威性,依赖领域知识本体所计算的概念相似度更加合理。

在第 1 节中介绍了本体中概念间语义重合度、语义距离及层次深度的定义,但是在定义语义重合度和语义距离时没有考虑到最顶层概念的特殊性。领域本体中的最顶层概念往往是对该领域某一特征的高度概括,这些最顶层概念间的语义联系往往没有上下位概念间的联系紧密,因此,使用普通意义上的语义重合度和语义距离计算公式无法明确表示出它们之间的联系。本文中对最顶层概念间语义重合度和语义距离有如下约定:

约定 1 本体最顶层概念间的语义重合度为 1。

约定 2 本体最顶层概念间的语义距离等于该本体中最顶层概念的数目。

根据本体中概念的语义重合度、语义距离及层次深度对概念语义相似度产生的影响,本文提出一个计算本体中概念语义相似度的公式:

$$cn\_sim(c_1,c_2)=\frac{|p(c_1)\cap p(c_2)|}{Dis(c_1,c_2)(|h(c_1)-h(c_2)|+1)} \quad (1)$$

其中,$c_1\in d_1$,$c_2\in d_2$。

需要注意的是,若本体中两个概念被定义为同义概念,则其概念相似度直接赋值为 1。

4.2 文本间相似度的计算

转换后的文本概念向量集合由两部分组成:本体中概念和未登录词。因此,本节将两个文本间相似度的计算分为两部分:文本中本体概念相似度和未登录词的相似度。

4.2.1 权值的计算

首先提出本体概念在文本向量集合中的权重的定义。

定义 4(本体概念权重) 本体概念权重是指该本体概念在文本中出现的次数与向量集合中所有特征词出现的次数的比值。若使用 $feq(c_i)$表示词语 c 在概念向量 i 中的频度,则概念权重的计算公式为:

$$wt(c_i)=\frac{feq(c_i)}{\sum_{j=1}^{n}feq(c_i)} \quad (2)$$

未登录词在文本中的权重采用经典的 tf * idf[12] 公式计算。其考虑两个因素:1)词语频率 tf (term frequency):词语在文档中出现的次数;2)词语倒排文档频率 idf (inverse document frequeney):该词语在文档集合中分布情况的一种量化,常用的计算方法为:

$$idf=\log_2(N/n_k+0.01) \quad (3)$$

其中,N 为文档集合中的文档数目,n_k 为出现该词语的文章数。

4.2.2 文本相似度的计算

(1) 文本本体概念相似度：由概念向量集合中本体概念间的相似度所导致的文本相似度。计算公式为：

$$cn\_wsim(c_1,c_2)=cn\_sim(c_1,c_2)*(wc_1+wc_2)/2 \quad (4)$$

其中，$cn_sim(c_1,c_2)$由式(1)计算得到；wc_1 和 wc_2 是概念 c_1 和 c_2 的权重，由式(2)计算得到。

需要注意的是：如果两个概念不属于同一个本体，则概念间相似度置为 0。

计算文本本体概念相似度的算法描述如下。

算法 3 文本本体概念相似度的计算

输入：文本 1 的概念向量 VC_1 和文本 2 的概念向量 VC_2

输出：两个文本的相似度值 *last_tsim*

```
1. Initialize() //初始化各变量及内存空间
2. for each vc1 in VC1//计算以第一个文本中概念为标准的文本概念相似度 tsim11
     for each vc2 in VC2
       if (SameOnto(vc1,vc2)) //判断两个概念是否属于同一个本体
         cn_sim= calSameOntoSim(vc1,vc2) //使用式(1)计算同一个本体中的概念间的相似度
         signUpdate(vc2) //将 vc2 标记为已处理
         cn_wsim= cn_sim *(wc1+wc2)/2 //使用式(4)计算同一本体的概念在文本中的相似度
         else cn_wsim=0
         alsim. add (cn_wsim)
     max_wsim= FindMaxSim(alsim)
     tsim11=tsim11+ max_wsim
3. for each vc2 in VC2//计算以第二个文本中概念为标准的文本概念相似度 tsim12
     for each vc1 in VC1
       if (SameOnto(vc1,vc2))
         cn_sim= calSameOntoSim(vc1,vc2)
         signUpdate(vc1)
         cn_wsim= cn_sim *(wc1+wc2)/2
         alsim. add (cn_wsim)
     max_wsim= FindMaxSim(alsim)
     tsim12=tsim12+ max_wsim
4. tsim=(tsim11+ tsim12)/2 //计算两个文本概念相似度 tsim
5. last_tsim= standardSim(tsim) //标准化
```

计算文本概念相似度的主要思想是：对于给定的两个文本向量，首先处理第一个文本中的概念。对于当前处理的概念，使用式(4)依次计算该概念与第二个文本中所有未处理概念的相似度，保存最大相似度值，并把与该相似度相对应的第二个文本中的概念标记为已处理。这是为了保证处理后续概念时，不会选择前面概念已选的第二个文本中的概念来计算相似度值。如果不进行标记，可能导致某些概念重复使用，使得文本相似度的计算不够准确和客观。该概念处理完毕后，继续按照上述方法处理第一个文本中的其他概念。所有概念均处理完毕后，将上面保存的所有本体概念相似度求和，得到以第一个文本为标准的文本间概念相似度。

再处理第二个文本中的概念，注意此时的两个文本为未经处理过的初始文本。对于当前处理的概念，使用式(4)依次计算该概念与第一个文本中所有未处理概念的相似度，保存最大相似度值，并把该相似度所属的第一个文本中的概念标记为已处理。接下来用同样方法处理第二个文本中的其他概念。所有概念均处理完毕后，将上面保存的所有本体中概念相似度求和，得到以第二个文本为标准的文本间概念相似度。

最后，将上面获得的两部分文本间概念相似度求和再除以 2，得到这两个文本间的相似度。对其进行标准化，使得文本间相似度的取值范围在[0,1]之间。计算公式为：

$$last\_tsim1(d_1,d_2)=1-\mu_0^{-tsim1(d_1,d_2)}$$

一般取 $\mu_0=10$。

(2) 文本未登录词相似度：由文本的关键词特征向量中未转换的关键词所导致的文本相似度。

文本的关键词通常并不能完全与本体中的概念相匹配，将那些保留下来的关键词的相似度作为最终文本相似度的一部分，会提高文本特征信息的利用率和文本相似度的精确程度。

文本未登录词向量的相似度的计算公式为：

$$tsim2(d_1,d_2)=\frac{\langle d_1,d_2\rangle}{\sqrt{\langle d_1,d_2\rangle\langle d_1,d_2\rangle}}$$

其中，两个文本的内积为 $\langle d_1,d_2\rangle=\sum_k r_k^2 feq_1(k)feq_2(k)$，$r_k$ 是词 k 的反文本频度，由式(3)计算得到；$feq_i(k)$是两个文本中相同的未登录词 k 出现在文本 i 中的频度。不同的未登录词间的相似度直接赋值为 0。如果两个文本中没有相同的未登录词，则将这两个文本未登录词相似度置为 0。

综上，得到两个文本间的相似度：

$$tsim(d_1,d_2)=\alpha\cdot last\_tsim1(d_1,d_2)+(1-\alpha)\cdot tsim2(d_1,d_2) \quad (5)$$

其中，α 为调节因子，其作用是调节文本中本体概念与未登录词对文本相似度产生的影响。如果主要考虑基于概念的文本相似度，则可以使 α 较大，使计算侧重于前半部分。否则可以使 α 取较小的值，使未登录词部分的相似度起主要作用。

5 凝聚层次聚类过程

通过式(5)计算所有初始文本间的相似度来获得文本相似度集合，将该相似度集合作为凝聚层次聚类的初始输入样本，开始聚类过程。首先定义类间相似度。

定义 5(类间相似度) 两个类间的相似度是指两个类中所有文本的两两文本间相似度之和的平均值。计算公式如下：

$$csim(cs_1,cs_2)=\frac{\sum_{i\in D_1}\sum_{j\in D_2} tsim(d_i,d_j)}{n_1\cdot n_2} \quad (6)$$

其中，n_1 和 n_2 分别为类 cs_1 和 cs_2 中包含的文本数目。

类间相似度是聚类的依据。由上述定义可知，类间相似度考虑了两个类中所有文本间的相似度，使得聚类的过程更加准确。

凝聚层次聚类过程描述如下。

算法 4 凝聚层次聚类过程

输入：概念向量集合 C，相似度值集合 S，聚类数目阈值 *min_num*

输出：聚类结果集合 Ct

```
1. Initialize() //初始化各变量及内存空间
2. for i=1 to n
     Ct. add{i} //簇的初始化
3. while(Ct. Length ! =min_num)
     vector csim= Csim(Ct,S) //保存簇间相似度
     for each d(I,J) in csim
       if(d>d_max) {d_max=d;I_max=I;J_max=J}
```

Merge(I_max,J_max) //合并簇间相似度最大的两个簇

Ct. Length= Ct. Length−1

4. return result cluster

该过程中首先初始样本自成一类;然后使用式(6)计算每对类别间的相似度 $csim(cs_i,cs_j)$ $(1\leqslant i,j\leqslant n)$;选择最大相似度的类对 $max\_csim(cs_i,cs_j)$,并将其合并为一个新类 cs_i,构成新的聚类($cs_1,cs_2,\cdots,cs_{n-1}$)。上述过程不断重复,直到归为一类或达到聚类数目的要求。

6 实验与分析

基于本文提出的文本层次聚类框架和实现技术,我们设计并实现了基于多个领域本体的文本层次聚类方法(Text Hierarchical Clustering Based on Several Domain Ontologies, THCO),通过对路透社新闻集合中的 900 个文档进行系统测试,验证此方法的有效性和准确性。

6.1 实验数据及评价准则

选取的测试文档均取自路透社 RCVI 新闻文本集合[13]。该文本集合中所有文档都经过人工标定类别,并进行了预处理,以固定的格式(文章号+词集)表示各个文档。该文本集合被组织为多层类的结构,其中第 1 层为根;第 2 层为 4 个类别:CCAT,GCAT,MCAT 和 ECAT;第 3 层为 13 个类别;3 层以上有 99 个子类。从其中的 9 个子类中各取 100 篇文档作为测试文本集,如表 1 所列。

表 1 语料库信息

Category Name	Category Description	Parent Category
E11	Economic performance	ECAT:Economics
E12	Monetary/Economic	
C11	Strategy/plans	CCAT: Corporate/Industrial
C12	Legal/Judicial	
C14	Share listings	
GSPO	Sports	GCAT:Government/Social
M11	Equity Markets	MCAT:Markets
M12	Bond Markets	
GEMV	Environment and Natural word	GCAT: Government/Social

针对语料库的信息,选取 5 个已由领域专家构建好的领域本体,它们分别是经济本体(Economic Ontology)、工业本体(Industrial Ontology)、体育本体(Sport Ontology)、市场交易本体(Market Ontology)和环境本体(Environment Ontology)。这些本体都是采用 OWL 来进行表述的。本文中使用 Jena API 作为本体文件的解析工具。

选用的评价准则是簇间的互信息 MI[14] 的标准化 NMI[15],可以消除聚类结果簇的个数对评价聚类结果好坏的影响,优于纯度(purity)和熵(entropy)的方法。NMI 的值越接近 1,表示聚类效果越好。计算公式如下:

$$NMI=\frac{\sum_{h,l}n_{h,l}\log\left(\frac{nn_{h,l}}{n_h n_l}\right)}{\sqrt{\left(\sum_h n_h\log\left(\frac{n_h}{n}\right)\right)\left(\sum_l n_l\log\left(\frac{n_l}{n}\right)\right)}}$$

其中,n_h 表示在语料库已分好的类别 h 中的数据样本数目,n_l 表示在类别 l 中的数据样本数目,$n_{h,l}$ 表示在类别 h 和 l 共有的数据样本的数目,n 表示数据样本总数。

6.2 实验结果

图 1 为聚类数目为 17、文本相似度权重调节因子 α 取不同值时 THCO 方法的聚类效果。其中横轴是调节因子 α 的取值,纵轴是相应的 NMI 值。

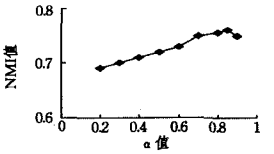

图 1 调节因子 α 与 NMI 的对应关系

图 1 表明,调节因子越大,即在文本相似度计算时较偏重概念文本相似度,聚类效果越好;但当调节因子过大时(大于 0.85),聚类效果将会降低。这说明本体的应用在一定程度上会使得聚类结果更为精确,同时某些出现频繁的关键词的作用也是不可忽视的,因此本文中的 THCO 方法兼顾了二者,得到了较好的聚类效果。

表 2 为当调节因子 $\alpha=0.85$ 时基于领域本体的凝聚层次聚类与其它聚类方法所得到的 NMI 值相比较的结果。

表 2 聚类结果 NMI 值对比

Cluster Num	11	15	17	21
MBM	0.18±0.05	0.19±0.03	0.17±0.02	0.18±0.03
MM	0.52±0.02	0.54±0.04	0.54±0.04	0.56±0.02
CLUTO	0.55±0.02	0.58±0.01	0.58±0.01	0.57±0.01
Co-clustering	0.36±0.01	0.46±0.01	0.50±0.01	0.51±0.01
Domain ontology clustering	0.69±0.02	0.75±0.03	0.76±0.01	0.78±0.01

实验结果表明,THCO 聚类方法能较好地提高聚类的准确度。

图 2 中显示了 THCO 方法对于文本关键词集合的降维效果。横轴是文本中关键词的数目,纵轴是概念向量集合中的概念数目。

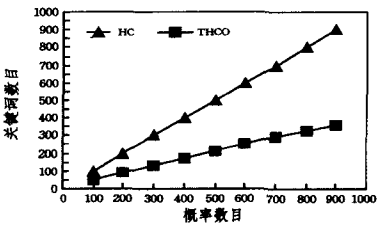

图 2 传统 HC 方法与 THCO 方法的降维效果比较

图 2 表明,THCO 聚类方法中聚类的对象空间远远低于传统聚类方法的对象空间,从而大大降低了聚类复杂性。传统 HC 方法与 THCO 方法的性能比较如表 3 所列。

表 3 传统 HC 方法与 THCO 方法性能比较

样本个数	类别个数	运行时间	
		HC	THCO
100	5	10.969	8.876
100	3	10.469	8.963
100	2	10.328	8.996
200	5	11.308	9.674

实验结果表明,THCO 聚类方法能够较好地提高聚类速度。

结束语 本文提出了基于多个领域本体的文本层次聚类算法,它将关键词集合转换成包含本体概念和未登录词的文本向量集合。该集合作为文本的知识表示形式,既能充分利用文本的信息,又具有较好的降维效果。计算文本向量集合的相似度时,利用本体良好的概念层次结构来计算文本向量中本体概念向量集合的相似度。与通常的相似度计算方法局

限于词表面的联系不同，该计算过程充分考虑了词语间的语义关系。因此，文本聚类过程中使用本体来表示和处理文本，降低了聚类对象的维度，减小了聚类复杂度，提高了聚类效率和准确度。此外，该方法还具有较好的灵活性和适用性，可以实现对任意语言和任意领域文本的聚类。

参 考 文 献

[1] Sanguthevar R. Efficient parallel hierar chical-clustering algo - rithms [J]. IEEE Transactions on Parallel and Distributied Systems, 2005, 16(6): 497-502

[2] Li X G, Yu G, Wang D L. MMPClust: A skew prevention algorithm for model-based document clustering[C]// Proceedings of the 10th International Conference on Database Systems for Advanced Applications(DASFAA'05). 2005: 536-547

[3] Miller G. WordNet: A lexical database for english[J]. Communications of the ACM (CACM'95), 1995, 38(11): 39-41

[4] Agirre E, Rigau G. Word sense disambiguation using conceptual density[C]// Proceedings of the 16th Conference on Computational Linguistics (COLING'96). 1996: 16-22

[5] Dario B, Fulvio C, Federico P. Automatic learning of text-to-concept mappings exploiting WordNet-like lexical networks [J]. ACM, 2005: 1639-1644

[6] Kolte S G, Bhirud S G. Word Sense Disambiguation Using WordNet Domains[J]. ACM, 2008: 1187-1191

[7] Sedding J, kazakov D. Wordnet-based text document clustering [C]// Proceedings of the Third Workshop on Robust Methods in Analysis of Natural Language Data (ROMAND'04). 2004: 104-113

[8] Hotho A, Staab S, Stumme G. Ontologies improve text document clustering[C]// Proceedings of the Third IEEE International Conference on Data Mining(ICDM'03). 2003: 541-544

[9] Hotho A, Maedche A, Staab S. Ontology-based Text Document Clustering[J]. Kunstliche Intelligenz, 2002, 16(4): 48-54

[10] Xie H W, Yan X L, Yu X L. Research on Web Page Clustering Based on Ontology [J]. Computer Science, 2008, 35(9): 153-155

[11] Li P, Tao L, Wang B Z. Measuring semantic similarity in ontology and its application in information retrieval [J]. Computer Engineering and Design, 2007, 28(1): 227-229

[12] Salton G, Buckley C. Term-weighting Approaches in Automatic Text Retrieval [J]. Information Processing and Management, 1988, 24(5): 513-523

[13] David D L, Yang Y, Fan L. RCV1: A New Benchmark Collection for Text Categorization Research[J]. Journal of Machine Learning Research, 2004, 5: 361-397

[14] Krishnamurthy B. On stationarity in Internet measurements th - rough an information-theoretic lens [C] // Proceedings of the 21st IEEE International Conference on Data Engineering (ICDE'05). 2005: 1185-1185

[15] Strehl A, Ghosh J. Cluster ensembles-a knowledge reuse framework for combining partitions[J]. Journal of Machine Learning Research, 2002, 3: 583-617

(上接第 187 页)

的特点实施 DM XML 的结构连接，迅速完成 Twig 查询。

在模式解析中，可能得到许多无关的简单路径，如图 4 中的 A/B/D/D/D∈ A(n_2. Path)，该路径下的所有元素绝不会作为输出元素。另外，在模式解析完成后，多个 $P^{//,[]}$ 子类树模式间可能存在多个相同路径而造成部分子模式结构相同。因此，如何消除结构约束节点的无效路径来减少访问数据库的次数，如何充分利用多个解析模式之间的相同结构来减少结构连接的次数，是将来的研究方向。

参 考 文 献

[1] Miklau G, Suciu D. Containment and equivalence for a fragment of XPath [J]. Journal of the ACM, 2004, 51(1): 2-45

[2] 高军，杨冬青，唐世渭，等. 基于树自动机的 XPath 在 XML 数据流上的高效执行[J]. 软件学报，2005，16(2)：223-232

[3] Busse R, Carey M, Florescu D, et al. Xmark: An XML benchmark project. 2003[EB/OL]. http://monetdb. cwi. nl/xml/index. html

[4] Wang W, Wang H Z, Lu H J, et al. Efficient processing of XML path queries using the disk-based F&B index[C]// Proc. of the 31st Int'l Conf. on Very Large Data Bases (VLDB). 2005: 145-156

[5] Chen Ting, Lu Jiaheng, Tok W L. On boosting holism in XML twig pattern matching using structural indexing techniques[C]// Proc. of ACM SIGMOD Int'l Conf. on Management of data. 2005: 45-46

[6] 周军锋，孟小峰，蒋瑜，等. F-Index：一种加速 Twig 查询处理的扁平结构索引[J]. 软件学报，2007，18(6)：1429-1442

[7] Lu J H, Tok W L, Chart C Y, et al. From region encoding to extended dewey: On efficient processing of xml twig pattern matching[C]// Proc. of VLDB05. 2005: 193-204

[8] Shanmugasundaram J, et al. Relational databases for querying XML documents: limitations and opportunities [C] // Proc. of VLDB. 1999: 302-314

[9] Peter T W. Minimizing simple XPath expressions[C]// Proc. of the 4th Int'l Workshop on Web and Databases (WebDB 2001). ACM Press, 2001: 13-18

[10] 李国良，冯建华，塔娜，等. TwigStar——快速处理 XML Twig 查询中含通配符 * 的算法[J]. 计算机研究与发展，2006，43(增刊)：430-437

[11] Milo T, Suciu D. Index structures for path expressions[C]// Proc. of the Int'l Conf. on Database Theory (ICDT). Springer-Verlag, 1999: 277-295

[12] Arion A, Bonifati A, Manolescu I, et al. Path Summaries and Path Partitioning in Modern XML Databases[J]. World Wide Web, 2008, 11(1): 117-151