一种基于本体相似度计算的文本聚类算法研究

王 刚[1,2] 钟国祥[2]
(安康学院电子与信息工程系 安康 725000)[1] (重庆教育学院科技处 重庆 400067)[2]

摘 要 为了改善文本聚类的质量,得到满意的聚类结果,针对文本聚类缺少涉及概念的内涵及概念间的联系,提出了一种基于本体相似度计算的文本聚类算法 TCBO(Text Clustering Based on Ontology)。该算法把文档用本体来刻画,以便描述概念的内涵及概念间的联系。设计和改进了文本相似度计算算法,应用本体的语义相似度来度量文档间相近程度,设计了具体的根据相似度进行文本聚类的算法。实验证明,该方法从聚类的准确性和聚类的关联度方面改善了聚类质量。

关键词 本体,相似度,文本聚类,语义

Study on Text Clustering Algorithm Based on Similarity Measurement of Ontology

WANG Gang[1,2] ZHONG Guo-xiang[2]
(Dept. of Electronic & Information Engineering, Ankang University, Ankang 725000, China)[1]
(Educational College of Chongqing, Chongqing 400067, China)[2]

Abstract To improve the quality of text clustering and get the satisfactory clustering results, we proposed a text clustering based on similarity of ontology. By organizing text as ontology, we were easy to represent the meanings and relations of concepts. We designed and improved the measurement of similarity and measured the text similarity by similarity of text ontology, we designed the algorithm of text clustering based on similarity. Experiments show that our method can avoid using the term isolation and high-dimensional, and can improve the clustering quality in correction degree and association degree.

Keywords Ontology, Similarity, Text clustering, Semantic

1 引言

概念相似性度量在服务选择[1,2]、自然语言理解、文献检索等领域具有重要的作用。当前主要模型有:(1) 基于距离的语义相似度计算模型,该模型简单、直观,但依赖预先建立好的概念层次网络,网络的结构直接影响到语义相似度的计算;(2) 基于内容的语义相似度计算模型,该模型充分利用了信息理论和概率统计理论的相关知识,但是这种方法不能更细致地区分层次网络中各个概念之间语义相似度的值;(3) 基于属性的语义相似度计算模型,该模型可以很好地模拟人们平时对现实世界中事物之间的认识和辨别,但要求对客观事物的每一个属性进行详细和全面的描述。本体相似性度量[3,4],包括概念相似性度量、关系相似性度量。词汇相似性度量通过计算词语之间的距离来实现,主要是利用本体的层次以及语料库,通过判断词语的相似性来计算[5];(4)认知心理学对相似性研究主要有以下几个模型:几何相似性模型、Tversky 模型以及 Rodriguez_Egenhofer 模型,这些模型都提出了计算相似度的方法与公式。不过,这些模型注重对象相似属性的个数,忽略了属性重要性的差异以及属性的语义扩展。本文把心理学中对相似性研究的成果引入到概念相似性度量,提出了基于语义元支持度的概念相似性度量方法,该方法首先对特征进行语义扩展,用语义元表示概念内涵,引入支持度来表现不同语义元对概念内涵表示的贡献,综合考虑相似性、相关性、差异性、不同义元的支持度以及不对称性,通过比较语义元的相似性,实现概念相似性的度量。把关系作为一种特殊的概念,进行关系的比较,最后得到了基于语义元的本体相似度计算方法。

聚类是将物理或抽象对象集合按有关特性的相似程度进行分组的过程。聚类产生的每一组数据称为一个簇,簇中每一个数据称为一个对象。聚类的目的是使同一簇中对象的特性尽可能地相似,而不同簇对象之间的差异尽可能地增大。聚类作为一种无监督的学习方法,能从数据集中发现数据的分布情况,是一种强有力的信息处理方法。随着 WWW 以及各种文本资源的不断增长,文本聚类也得到越来越多的重视,在很多文本挖掘和信息检索系统中发挥着重要的作用,快速和高质量的文本聚类技术可以将大量信息组织成少数有意义的簇,可以改善检索性能、提供导航与浏览机制、发现相似文档等,文本聚类研究成为数据挖掘的一个非常重要的课题。

到稿日期:2009-10-12 返修日期:2010-01-14 本文受陕西省教育厅项目(09JK317),基于本体的服务研究(AYQDZR200916),智能信息处理技术关键问题及应用研究(2008akxy005)资助。

王 刚(1972—),男,博士,主要研究方向为人工智能、服务计算,E-mail:aktcdawang@163.com;钟国祥(1970—),男,博士,教授,主要研究方向为人工智能。

与以往的聚类应用相比，文本聚类面临的挑战[1]有：(1) 由于非常高的数据维数，要求聚类算法能够处理稀疏矩阵，或者对矩阵降维；(2) 数据库规模可能非常大(例如万维网)，要求聚类算法对大型数据库也要有很高的分析效率；(3) 要有可以理解的聚簇描述，聚簇描述对于非专业用户也应该是可以理解的。目前，很多文本聚类算法，如 Scatter/Gather，K-means，bisecting k-means，TCUAP 等，都不能很好地解决上面提到的问题。一些基于语义相似度的文本聚类算法，例如文献[8]，只是将文档表示成一个个孤立的名词列表，列表中的名词互不相同。以这种文档表示法为基础的聚类分析忽略了概念间的语义联系和语义扩展。还有一种方法是将文档表示成如下形式 $D=\{(w_1,f_1),(w_2,f_2),\cdots,(w_n,f_n)\}$[9]，$w_i$ 表示在一个文档中出现的概念；f_i 为 w_i 在文档中出现的次数。这种文档表示法称为概念列表，由于把概念表示为名词列表，忽略了概念的语义内涵。本文提出了一种新的基于语义相似度的文本聚类算法——TCBO，该算法采用本体表示文档，通过使用改进的相似度计算算法，利用文档间的语义相似度进行聚类，大大减少了数据维数，也利于文档间语义相似度的计算；并且可以利用概念列表中的某些概念来对发现的聚簇进行描述，已经有很多算法用于提取文档中的概念及关系，如通过构建文本分析自动机及构建文本分析器，可以成功地从文本中提取概念和关系。

2 相似度的计算

2.1 概念间相似度的计算

本文把特征和推理的结果称为语义元。语义元是表示语义不能再划分的基本单位[4]，概念 C 表示为 $C=\{\bigcup_{i=1}^{k} e_i\}$，它表示概念由 k 个语义元组成，例如：雪：{冷，冬天，毛衣，棉衣，0℃，风，水，…}等。语义元中，有概念的定义特征，也有概念的特异特征，称为基本语义元，还有语义扩展的结果，称为扩展语义元。概念语义扩展通常是按照推理规则进行推理的，推理的结果是扩展语义元。对语义元，可以按照如下公式来判断其是否匹配。对概念 $C_1:\{e_1,e_2,e_3,\cdots,e_m\}$；$C_2\{e_1',e_2',e_3',\cdots,e_n'\}$

$$\mathrm{sim}(e_i,e_j')=\begin{cases}0, & e_i|\neq e_j'\\ 1, & e_i|=e_{j_j}'\end{cases}$$

不同的语义元，在表现概念内涵方面具有差异，本文用支持度来刻画它们的差异。

设 $C_1:\{e_1:\alpha_1,e_2:\alpha_2,e_3:\alpha_3,\cdots,e_m:\alpha_m\}$；$0<\alpha_i\leqslant 1$，$e_i:\alpha_i$ 表示 e_i 的支持度为 α_i。

可按如下规则进行语义扩展：

(1) $R(A)\rightarrow e_k$：对象 A 按规则 R 推理，得到的语义元 e_k 的支持度为 1；

(2) $R(e_i)\rightarrow e_k$：语义元 e_i 按规则 R 推理，得到的语义元 e_k 的支持度为 $e_i:\alpha_i$

(3) $R(e_i\cap e_m\cap\cdots\cap e_n)\rightarrow e_k$：对语义元的交集按规则 R 推理，得到的语义元 e_k 的支持度为 $\min_{i=1}^{n}(e_i:\alpha_i)$，即取最小的支持度。

(4) $R(e_i\cup e_m\cup\cdots\cup e_n)\rightarrow e_k$：对语义元的并集按规则 R 推理，得到的语义元 e_k 的支持度为 $\max_{i=1}^{n}(e_i:\alpha_i)$，即取最大的支持度。

确定基本语义元 e 的支持度 d 可以按如下的算法进行。

算法 1 确定基本语义元 e_i 的支持度

A_i 为对象集合，e_i 为语义元，k 为包含语义元 e_i 的对象数量

```
Begin
For(i=1 to n)
{
if(e∈A_i)
k=k+1;
}
```

$$d=\frac{k}{\sum_{i}^{n}|A_i|};$$

```
End
```

算法 2 计算概念的相似度

```
Begin
```

(1) 假设有对象 A,B，支持度集合分别为 sd_1,sd_2

(2) $I=A\cap B$，$e_x\in A\cap B$，$sd'_1\in sd_1$，$sd'_2\in sd_2$，$I(A)=\{e_k\}$，$e_k\in A$，$e_k\in I$，$m=|A\cap B|$

(3) $r_1=\frac{a}{a+b}$；其中，$a=\sum_{i=1}^{M}(e_i:\partial_i+e_i':\beta_i')$，$b=\sum_{i=1}^{M}(e_i'':\partial_i'+e_i''':\beta_i'')$ $\partial_i,\beta_i'\geqslant\delta,e_i\in I,\partial_i',\beta_i''<\delta,e_i\in I(A),e'_i\in I(B)$，$e_i''\in I(A),e_i'''\in I(A),e_i=e'_i,e_i''=e_i'''$。

(4) $r_2=\frac{x}{x+y}$；其中 $x=\sum_{i=1}^{M}e_i:\partial_i,e_i\in I(A),\partial_i\geqslant\delta$，$y=\sum_{i=1}^{M}e_i':\partial'_i,e'_i\in I(A),\partial_i'<\delta$

(5) $r_3=\frac{|A'|}{|I|},e_i\in A',e_i:\partial_i\geqslant\delta,e_i\in I$

(6) $r_4=\frac{\sum_{i=1}^{M}|W_i|}{2|I|},w_i:\partial_i\geqslant\delta,w'_i:\partial'_i\geqslant\delta,w_i\in I(A),w'_i\in I(B)$

(7) $p(e_i,e_j)=\frac{1}{e^{|r_1-r_2|}}\times\frac{1}{e^{|r_3-r_4|}}$，表示交集中主要特征支持度差别越小，相似度越大，交集中主要特征个数差越小，相似度越大。

(8) $q(e_i,e_j)=\frac{f(A\cap B)}{f(A\cap B)+\alpha\times f(A\cap B)+\beta\times f(A\cap B)}$，$\alpha$ 和 β 为系数。

(9) $\mathrm{sim\text{-}concept}(A,B)=p(e_i,e_j)\times q(e_i,e_j)$

(10) $\mathrm{sim\text{-}concept}(A,B)$ 就是概念 c_i,c_j 的相似度。

```
End
```

2.2 文本间的相似度计算

本体由概念和关系组成，本体的相似性度量包括概念和关系的相似性度量[4]。本文采用加权平均的方法，先度量本体概念的相似度，然后度量关系的相似度，最后求得本体的相似度。利用前面的算法得到

$$\mathrm{sim\text{-}concept}(O_1,O_2)=\frac{1}{n_1\times n_2}\times\sum_{i=1,j=1}^{i=n_1,j=n_2}\mathrm{sim}(c_i,c_j);n_1,n_2$$

为 O_1,O_2 中概念的个数。

关系由概念和关系核组成。例如最基本的关系 $R_1=\{c_1,c_2,r_1\}$，$R_2=\{c_3,c_4,r_2\}$，表示关系 R_1 包含概念 c_1,c_2，关系核为 r_1，关系 R_2 包含概念 c_3,c_4 和关系核 r_2。按如下公式计算关系 R_i,R_j 的相似度。

定义 $\mathrm{sim\text{-}relation}(R_i,R_j)=\mathrm{sim\text{-}concept}(c_i^1,c_j^1)\times\mathrm{sim\text{-}concept}(c_i^2,c_j^2)\times\mathrm{sim\text{-}concept}(r_i,r_j)$，对于上述关系 R_1,R_2，它们的相似度计算包括概念的相似度与关系名的相似度计算，$\mathrm{sim}(R_1,R_3)=\mathrm{sim}(c_1,c_3)\times\mathrm{sim}(c_2,c_4)\times\mathrm{sim}(r_1,r_3)$。

本体中关系相似度定义为：$\mathrm{sim\text{-}relation}(O_1,O_2)=\frac{1}{m\times n}\sum_{i=1,j=1}^{i=m,j=n}\mathrm{sim\text{-}relation}(R_i,R_j)$，其中 m,n 为 O_1,O_2 中关系的数目。

$R_i \in O_1$, $R_j \in O_2$。由此可以得出本体的相似度：sim-ontology(O_x, O_y) = sim-concept$(O_x, O_y) \times$ sim-relation(O_x, O_y)，文本由本体构成，文本 $T_i = \{O_{i1}, O_{i2}, O_{i3}, \cdots, O_{in}\}$，$T_j = \{O_{j1}, O_{j2}, O_{j3}, \cdots, O_{jp}\}$，$\mathrm{Sim}(T_i, T_j) = \frac{1}{\partial \times \beta} \sum_{p=1}^{p=\partial} \sum_{q=1}^{q=\beta} \mathrm{sim}(O_{ip}, O_{jq})$，$\partial$，$\beta$ 为文本包含的本体数目。

3 文本语义聚类算法

一般地，聚类算法可以分为层次法和划分法[7]。层次法又可以通过两种途径实现：合并和分裂。合并时，先使每个样本各成一类，然后通过合并不同的类来减少类别数目，直到满足终止条件。而分裂法是将所有样本归入一类，然后通过后续分裂来增加类别个数，直到满足终止条件。最典型的划分法是 K-均值聚类及其各种变形。K 均值聚类是从样本中随机取出 c 个样本作为初始的聚类中心，在每步迭代中，计算每个类的质心，每个样本被归入到最近的质心，直到聚类不再有任何改变。本文在 k 中心点聚类算法中引入相似度计算[10,11]，设计算法如下。

Begin

(1)输入 n 个样本

(2)根据 n 个样本，构建对应的本体

(3)随机选择 k 个对象作为初始的中心点 T_j

(4)Repeat

(5)指派每个剩余的对象给相似度最接近的中心点代表的簇

(6)随机选择簇中一个非中心对象 T_{random}

(7)计算 $S_1 = \sum_{i=1}^{i=k} |p_{s_1} - m_i|^2$，计算用 T_{random} 代替 T_j 的代价 $S_2 = \sum_{i=1}^{i=k} |p_{s_2} - n_i|^2$，$p_{s_1}$ 是中心点与其它点的平均值，m_i 是中心点与其它点的相似度，p_{s_2} 是 T_{random} 与其它点相似度的平均值，n_i 是 T_{random} 与其它点相似度的相似度

(8)If $S_1 - S_2 < 0$ then 用 T_{random} 代替 T_j，得到新的集合，否则，认为中心点是可以接受的

(9)Until 不在发生变化

End

4 实验与分析

复旦大学中文自然语言处理开放平台设计了比较全面的语料库，本文选用 20 个库中的 200 篇文档进行测试，从每个库选择 10 篇文档，分别建立其本体，采用 C 语言进行编程，然后从库中剩余的文档中随机选取文档作为测试文档，考察其分类准确率和分类关联度，分类准确率计算如下：利用随机数发生器，随机从 9804 篇文档中选出一篇，构建其本体，然后测试其与已经分类文本的本体相似度，然后把该文档归入最大相似度对应的文本所属类，为了对比，考虑 5 种聚簇，每种聚类重复进行 200 次，然后计算分类成功的百分比。即公式：correct-degree$(T_x) = \frac{a}{200}$，a 为文档 T_x 被分类成功的次数。一个文档既可以属于 C 类，也可以属于 D 类，我们用关联度来度量 C 类文档 T 属于 D 类的程度，从 200 篇文档中采用随机数发生器随机选择 1 篇文档，然后分别计算它与其它文档的相似度，考查属于其它类、同时相似度比较高的文档的个数，然后计算该个数与其原所属类文档个数的比，运行 200 次得到分类关联度。本文考查平均关联度即公式：satisfy-degree$(T_y) = \frac{\sum \frac{y}{x}}{200}$，$x$ 为文档 T_y 所属类文档的个数，y 为 T_y 与其它类文档相似文档的个数。

关联度和部分测试数据及结果如图 1，图 2 所示，TCBO 算法为本文的算法，TCUSS，TCUAP，K-MEANS 为参考文献[9]提供的算法。由于本体构建的工作量很大，本文采用人工方式构建，相似度计算算法采用 C 语言编写，结果表明，TCBO 在聚类准确率方面优于其它算法，在关联度方面，由于其采用了计算本体相似度，使得关联度指标明显优于其它算法。

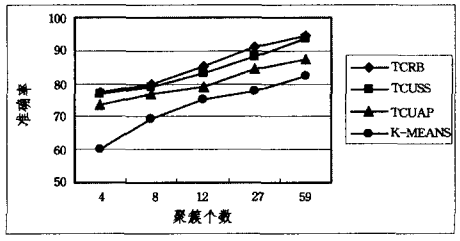

图 1 聚类质量比较

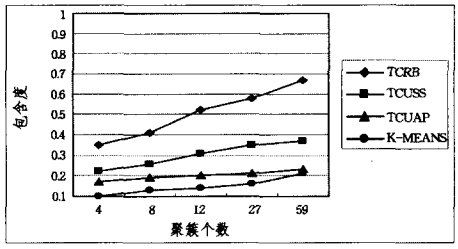

图 2 聚类关联度比较

结束语 本文提出了一种新的基于语义相似度的文本聚类算法——TCBO。它用概念列表表示文档，把文档组织为本体，通过改进的本体相似度计算算法，计算文档的相似度，然后利用文档的相似度进行聚类，通过本体的引入，解决了以往聚类算法数据维数过高和难于描述聚簇的问题，也初步实现了根据语义进行聚类。实验结果表明，TCBO 既提高了聚类的质量，也提高了聚类的包含度。这对于推荐系统以及基于 WEB 的电子商务，提供了一个可行的途径，进一步的工作包括继续改进和优化文档组织为本体的方法，继续改进和研究相似度计算方法，改进和研究聚类的具体算法。

参 考 文 献

[1] Weinstein P, Birmingham W. Comparing concepts in differentiated ontologie[C]// Proc. of KAW-99. 1999

[2] Paolucci M. Semantic Matching of Web Service Capabilities[C]// Proceedings of the First International Semantic Web Conference (ISWC). 2002

[3] Wache H, Vogele T, Visser U, et al. Ontology-Based Integration of Information—A Survey of Existing Approaches[C]// Proc. of the IJCAI-01 Workshop: Ontologies and Information Sharing. Seattle, WA, 2001: 108-117

[4] 王刚，邱玉辉. 一个基于语义元的相似度计算方法[J]. 计算机应用研究，2008(11): 3253-3255

[5] Pandya A, Bhattacharyya P. Text similarity measurement using concept representation of texts[C]// Proceedings of First International Conference on Patttern Recognition and Machine Intelligence. Berlin, Germany: Springer, 2005

（下转第 228 页）

$|Q_i \cap X|=h$，$|Q_j \cap X|=k$，$|Q_i|-|Q_i \cap X|=m$，$|Q_j|-|Q_j \cap X|=n$。此时，$h+k=a$，$m+n=b$。

令 $h+m=h_1$，$k+n=k_1$，则 $a+b=h_1+k_1$，而 $\frac{4}{|U|}\sum_{x\in P_1} B_1(X)(x)(1-B_1(X)(x))=\frac{4}{|U|}(h+k)(1-\frac{h+k}{h_1+k_1})$，$\frac{4}{|U|}\sum_{x\in Q_i\cup Q_j} B_2(X)(x)(1-B_2(X)(x))=\frac{4}{|U|}h(1-\frac{h}{h_1})+\frac{4}{|U|}k(1-\frac{k}{k_1})$。因此，$\frac{4}{|U|}\sum_{x\in P_1} B_1(X)(x)(1-B_1(X)(x))-\frac{4}{|U|}\sum_{x\in Q_i\cup Q_j} B_2(X)(x)(1-B_2(X)(x))=\frac{4}{|U|}\frac{(hk_1-kh_1)^2}{h_1k_1(h_1+k_1)}\geqslant 0$，当且仅当 $\frac{h}{h_1}=\frac{k}{k_1}$ 时等号成立。

综合①，②，③，有 $EB_2(X)\leqslant EB_1(X)$。此时，

$$\frac{|Q_i|}{|U|}\left(1-\frac{|Q_i|}{|U|}\right)+\frac{|Q_j|}{|U|}\left(1-\frac{|Q_j|}{|U|}\right)-\frac{|P_1|}{|U|}\left(1-\frac{|P_1|}{|U|}\right)$$
$$=\frac{|Q_i|}{|U|}\left(1-\frac{|Q_i|}{|U|}\right)+\frac{|Q_j|}{|U|}\left(1-\frac{|Q_j|}{|U|}\right)-\frac{|Q_i|+|Q_j|}{|U|}\left(1-\frac{|Q_i|+|Q_j|}{|U|}\right)$$
$$=\frac{2|Q_i||Q_j|}{|U|}>0$$

故 $E_{B_1}^{BN}(X)<E_{B_2}^{BN}(X)$。因此，$1-E_{B_2}^{BN}(X)<1-E_{B_1}^{BN}(X)$。

综合 1)，2)，3)，有 $EB_2(X)(1-E_{B_2}^{BN}(X))\leqslant EB_1(X)(1-E_{B_1}^{BN}(X))$，即 $F_{B_2}(X)\leqslant F_{B_1}(X)$。而且，当边界域被严格细分时“<”成立。

(4)由式(3)和式(4)易见 $0\leqslant ER(X)\leqslant 1$，$0\leqslant E_R^{BN}(X)<1$，则 $F_R(X)=ER(X)(1-E_R^{BN}(X))=1$ 当且仅当 $ER(X)=1$，且 $E_R^{BN}(X)=0$。由 $ER(X)=1$，知 $\forall x\in U, R(X)(x)=0.5$，故 $\underline{R}X=\emptyset$，$\overline{R}X=U$，所以 X 关于 R 是全不可定义的。而 $E_R^{BN}(X)=0$ 当且仅当存在唯一的 $R_i\in U/R$ 满足 $R_i=BN_R(X)$，又由 $\underline{R}X=\emptyset$，$\overline{R}X=U$ 知 $BN_R(X)=U$，因此 $U/R=\{U\}$。

(5)用反证法。因 $B_1\subseteq B_2\subseteq A$，故 $U/B_2\preceq U/B_1$，由定理 1 的证明知 $\rho_{B_2}(X)\leqslant\rho_{B_1}(X)$，假设 $\rho_{B_1}(X)\neq\rho_{B_2}(X)$，则 $\rho_{B_2}(X)<\rho_{B_1}(X)$，说明边界域 $BN_{B_2}(X))$ 比 $BN_{B_1}(X))$ 要“严格小”，即存在 $P_i(P_i\in U/B_1\wedge P_i\subseteq BN_{B_1}(X))$ 在 U/B_2 中被“严格细分”，因此 $U/B_2\prec U/B_1$。由本命题性质(3)得 $F_{B_2}(X)<F_{B_1}(X)$，这与已知 $F_{B_2}(X)=F_{B_1}(X)$ 矛盾。因此，$\rho_{B_2}(X)=\rho_{B_1}(X)$。证毕。

结束语 粗糙集是用于研究不确定问题的有力工具，所以其本身就带有不确定性。粗糙度和模糊度从不同角度给出了粗糙集不确定性的度量，前者比较直观，而后者较为抽象。粗糙集不确定性来自集合本身及近似空间两个方面，将这两个方面合理地结合起来，才能更好地反映粗糙集的不确定性。本文定义了近似空间中集合的相对知识粒度，将其与 Pawlak 粗糙度结合，给出了一种新的粗糙集粗糙性的度量。该度量既刻画了近似空间对粗糙集不确定性的影响，又去除了负域的干扰，并且满足随着近似空间知识粒的细分粗糙集的不确定性单调递减。粗糙集的模糊度与近似空间知识粒大小也是密切相关的，边界域对于模糊度有决定性的影响。本文结合重新定义的边界熵，给出了新的粗糙集的模糊度量，它可以更合理地反映不同知识粒度对集合的模糊性的影响，同时具备符合人们认知规律的一些性质。

参 考 文 献

[1] 李德毅，刘常昱，杜鹢，等. 不确定性人工智能[J]. 软件学报，2004，15(11)：1583-1594

[2] 王国胤. Rough 集理论与知识获取[M]. 西安：西安交通大学出版社，2001

[3] 苗夺谦，王珏. 粗糙集理论中知识粗糙性与信息熵关系的讨论[J]. 模式识别与人工智能，1998，11(3)：34-40

[4] 苗夺谦，范世栋. 知识的粒度计算及其应用[J]. 系统工程理论与实践，2002，22(1)：48-56

[5] 梁吉业，李德玉. 信息系统中的不确定性与知识获取[M]. 北京：科学出版社，2005

[6] Pawlak Z. Rough sets[J]. International Journal of Computer and Information Science，1982，11：341-356

[7] 李玉榕，乔斌，蒋静坪. 粗糙集理论中不确定性粗糙信息熵表示[J]. 计算机科学，2002，29：101-103

[8] Liang Ji-ye，Shi Zhong-zhi. The information entropy，rough entropy and knowledge granulation in rough set theory[J]. International Journal of Uncertainty，2004，12(1)：37-46

[9] Zadeh L A. Fuzzy sets[J]. Information Control，1965，8：338-353

[10] Chakrabarty K，Biswas R，Nanda S. Fuzziness in rough sets[J]. Fuzzy Sets and Systems，2000，110：247-251

[11] Liang Ji-ye，Chin K S，Dang C Y. A new method for measuring uncertainty and fuzziness in rough set theory[J]. International Journal of General Systems，2002，31(4)：331-342

[12] Mi Ju-sheng，Yee Leung，Wu Wei-Zhi. An uncertainty measure in partition-based fuzzy rough sets[J]. International Journal of General Systems，2005，34：77-90

[13] 王国胤，张清华. 不同知识粒度下粗糙集的不确定性研究[J]. 计算机学报，2008，31(9)：1588-1598

[14] 张文修，吴伟志，梁吉业，等. 粗糙集理论与方法[M]. 北京：科学出版社，2001

(上接第 224 页)

[6] 孙玉娣，裴勇. 基于可视化文本挖掘的本体构建[J]. 情报杂志，2007，26(12)

[7] Roy R，Mili H，Blettner M. Development and application of a metric on semantic nets[J]. IEEE Transaction on System，1989，19(1)

[8] Song Shaoxu，Li Chunping. TCUA P：a novel app roach of text clustering using asymmetricproximity[C]// Proceedings of the 2nd Indian International Conference on Artificial Intelligence. India：IICA I，2005：604-613

[9] 孙爽，章勇. 一种基于语义相似度的文本聚类算法[J]. 南京航空航天大学学报，2006(6)

[10] 范明，孟小峰. 数据挖掘概念与技术[M]. 北京：机械工业出版社，2002

[11] 王元明，熊伟. 异常数据的检测方法[J]. 重庆工学院学报：自然科学版，2009，23(2)：86-89