语义分析与词频统计相结合的中文文本相似度量方法研究 *

华秀丽[1,2], 朱巧明[2], 李培峰[2]
(1. 苏州大学 计算机科学与技术学院, 江苏 苏州 215006; 2. 江苏省计算机信息处理技术重点实验室, 江苏 苏州 215006)

摘 要: 基于统计的文本相似度量方法大多先采用 TF-IDF 方法将文本表示为词频向量,然后利用余弦计算文本之间的相似度。此类方法由于忽略文本中词项的语义信息,不能很好地反映文本之间的相似度。基于语义的方法虽然能够较好地弥补这一缺陷,但需要知识库来构建词语之间的语义关系。研究了以上两类文本相似度计算方法的优缺点,提出了一种新颖的文本相似度量方法,该方法首先对文本进行预处理,然后挑选 TF-IDF 值较高的词项作为特征项,再借助 HowNet 语义词典和 TF-IDF 方法对特征项进行语义分析和词频统计相结合的文本相似度计算,最后利用文本相似度在基准文本数据集合上进行聚类实验。实验结果表明,采用提出的方法得到的 F-度量值明显优于只采用 TF-IDF 方法或词语语义的方法,从而证明了提出的文本相似度计算方法的有效性。

关键词: 向量空间模型; 语义分析; 词频; 概率分布; 文本相似度
中图分类号: TP391 **文献标志码**: A **文章编号**: 1001-3695(2012)03-0833-04
doi:10.3969/j.issn.1001-3695.2012.03.008

Chinese text similarity method research by combining semantic analysis with statistics

HUA Xiu-li[1,2], ZHU Qiao-ming[2], LI Pei-feng[2]
(1. *School of Computer Science & Technology, Soochow University, Suzhou Jiangsu* 215006, *China*; 2. *Provincial Key Laboratory of Computer Information Processing Technology of Jiangsu, Suzhou Jiangsu* 215006, *China*)

Abstract: Based on the statistical text similarity measurements method used TF-IDF method to model text documents as term frequency vectors, and computed similarity between documents by using cosine similarity. This method ignored semantic information of text documents, the similarity value wasn't correct. Although based on semantics method made up for the drawback, but need of knowledge to construct the relationship between words. By studying the advantages and disadvantages of two kinds of methods, this paper presented a novel text similarity method, which firstly pre-processed text, then chose the terms with higher TF-IDF value as the feature items, next used semantic dictionary and TF-IDF method to compute the text similarity, finally used several K-means clustering methods for evaluating performance of the new text document similarity. Experimental results show that the method's F-measure is superior to the others' which proves that the proposed method is effective.

Key words: vector space model; semantic analysis; term frequency; probability distribution; text similarity

0 引言

文本聚类作为信息处理的一个重要方向,通过将大量信息组织成少数有意义的簇,并保证同一簇内的文本之间是相似的,达到改善检索性能的目的。文本相似度量方法是实现快速、高质量文本聚类的重要途径。另外,文本相似度量方法在信息检索[1]、图像检索[2]、文本摘要自动生成[3]、文本复制检测[4]等领域也有广泛的应用基础。目前,文本的相似度量方法主要分为基于统计学和基于语义分析两类。

基于统计学的文本相似度量方法是将文本看做一个个独立的词语,利用词语的词频信息将文本建模为高维而稀疏的向量,并利用向量间余弦相似度、Jaccard 相似度等方法计算文本间的相似度。基于统计学的计算方法主要包括向量空间模型、隐性语义索引模型、基于属性论的方法等。它的缺点表现在:需要大规模语料库的支持;忽略了词语之间存在的语义关系;文本表示模型维数高而且稀疏,处理困难。

基于语义分析的文本相似度量方法则利用特定领域的知识库来构建词语之间语义关系,以此考察文本之间的相似性。与基于统计学的计算方法相比,该方法不需要大规模语料库的支持,而且准确度较高,但知识库的建立是一项复杂而繁琐的工程,因此现有研究一般采用收录词比较完备的词典代替知识库。如文献[5,6]使用 HowNet[7]进行词语和句子的语义相似度研究,文献[8]使用同义词词林计算句子之间的相似度,文献[9]采用 WordNet[10]研究词语消歧等。

收稿日期: 2011-08-23; **修回日期**: 2011-10-15 **基金项目**: 国家自然科学基金资助项目(60970056,61070123,61003155);模式识别国家重点实验室开发课题基金资助项目;江苏省自然科学基金资助项目(BK2008160);高等学校博士学科点专项科研基金资助项目(20093201110006)

作者简介:华秀丽(1986-),女,山东泰安人,硕士研究生,主要研究方向为自然语言处理、文本相似度计算(huaxiuliemail@126.com);朱巧明(1963-),男,江苏苏州人,教授,博导,主要研究方向为中文信息处理;李培峰(1970-),男,江苏苏州人,副教授,博士,主要研究方向为中文信息处理.

本文针对传统方法进行文本相似度计算时存在的缺陷,在对文本表示模型进行有效的降维处理的基础上,提出了一种既考虑词项的概率分布、又兼顾词项之间的语义关系的文本相似度计算方法。同时表明在给定两个文本时,利用本文提出的算法对两者在语义层次以及词频统计层次进行综合衡量计算出的相似度更加有效、准确。

1 相关工作

1.1 向量空间模型(VSM)

向量空间模型是统计学方法中最为经典的一种文本相似度量方法,采用它进行文本相似度计算时,最重要的是计算词项的权重,也就是词项在文本中的重要程度,计算时一般采用 TF-IDF 方法。使用 TF-IDF 方法计算向量中词项的权重时会涉及到两个概念:

a)词频。某个词项在一个文本中出现的次数,通常情况下认为某个词的词频越大,它与文本的主题越相关。

b)逆文本频率[11]。某个词项在文本集合的多篇文本中出现的次数越多,该词项的区分能力越差。例如一个包含了 100 篇的文本集合中,如果某个词项 A 在 50 篇文本中都出现,而另外一个词项 B 只在 5 篇文本中出现,则词项 B 比 A 具有更好的区分能力。

利用上述概念计算每一个词项的 TF-IDF 值:

$$\mathrm{TFIDF}(\omega_i)=\mathrm{tf}(\omega_i)\times\mathrm{idf}(\omega_i)=\mathrm{tf}_j(\omega_i)\times\log(N/\mathrm{df}(\omega_i)) \quad (1)$$

其中:TFIDF(ω_i)表示当前词项 ω_i 的 TF-IDF 值,该值等于词项 ω_i 的词频 tf(ω_i)与逆文本频率 idf(ω_i)的乘积,具体地,文本 j 中任一词项 ω_i 的 TF-IDF 值可以通过 $\mathrm{tf}_j(\omega_i)$ 和 $\log(N/\mathrm{df}(\omega_i))$ 计算得出;$\mathrm{tf}_j(\omega_i)$ 表示当前词项 ω_i 在文本 j 中出现的频率;N 表示文本集合中所有文本的总数;df(ω_i)表示文本集合中有多少篇文本出现了当前词项 ω_i。通过对文本集合中的每个词项进行上述分析,得到每一篇文本中每一个词项的 TF-IDF 值,然后利用这些 TF-IDF 值为每一篇文本建立一个向量空间模型,通过余弦计算得到文本之间的相似性。

1.2 词语语义相似度计算

人们希望获取准确信息的要求越来越高,对仅利用词语的表面信息(如词频)的文本相似度计算方法提出了挑战。不能仅考虑词项在文本中的概率分布情况,还必须从深入挖掘文本语义的角度进行相似度计算。举一个简单例子,这里有两篇文章,一篇是关于目前教科书研究情况的介绍,另外一篇是关于课本研究情况的介绍,两篇文章中分别对“教科书”“课本”提到多次。如果采用传统的基于词频统计的相似度量方法对两篇文章进行相似度计算,则可能会因为两者的词项不同而被认为不相似,尽管“教科书”和“课本”是一对同义词。

正因为如此,人们开始研究词与词之间的相似度。词与词之间的相似度量需要将所有的词组织起来构成一个语义网络,通过考察该网络中词与词之间的边、节点等信息来建立词与词之间的相似度。英文最常用的是普林斯顿大学研究开发的 WordNet[10],而中文是由董振东先生编著的知网,也称做 HowNet[7]。本文采用 HowNet[7] 进行词语语义相似度计算。

文献[1]详细介绍了 HowNet 的知识结构以及知识描述语言的语法等内容,并据此提出了利用 HowNet 进行词语相似度计算的方法和词语相似度计算的公式:

$$\mathrm{sim}(S_1,S_2)=\frac{\alpha}{\alpha+(\mathrm{dist}(S_1,S_2))} \quad (2)$$

其中:S_1、S_2 表示两个义原;dist(S_1,S_2)表示它们的路径长度;α 是一个调节参数,表示相似度为 0.5 时的路径长度。

由于式(2)仅从义原路径长度来考虑两个词语的相似度,而未充分利用 HowNet 体系结构,计算结果不够准确。因此本文在原来算法的基础上对其进行了改进。通过研究发现,影响词语相似度的因素除义原节点之间的路径长度之外,义原所在概念树的深度以及概念树的密度也是影响相似度计算的重要因子。本文在式(2)的基础上,加入了义原所在树的深度信息,采用如下公式进行词语的相似度计算:

$$\mathrm{sim}(S_1,S_2)=\alpha\times(\mathrm{depth}(S_1)+\mathrm{depth}(S_2))/(\alpha\times(\mathrm{depth}(S_1)+\mathrm{depth}(S_2))+\mathrm{dist}(S_1,S_2)) \quad (3)$$

其中:depth(S)表示 S 距离根节点的层次。

同时对 HowNet 中未登录的词也进行了处理。对这些 HowNet 概念中未出现的词语,首先对它们进行切分,然后进行组合语义,最后借助参照概念修正组合语义。

2 文本预处理和特征选择

2.1 文本预处理

在对文本建立词项的词频向量之前,对文本进行适当的预处理是非常有必要的。预处理阶段首先要对文本进行分词。本文采用中国科学院的 ICTCLAS(http://ictclas.org/)分词工具,接着对分词过后的文本进行去除停用词的处理。此外,由于本文中提出的方法还需要对词项进行语义分析,因此除删除停用词外还需要进行下面三个预处理步骤:

a)需要处理文本中的人名、地名、组织机构名称等特殊词项。采用命名实体识别技术来处理文本中的这些特殊词项,将这些特殊词项统一替换为特定的字符串,人名对应于 PER,地名对应于 LOC,组织机构名称对应于 ORG 等。在对文本进行特征选择时,可以忽略这些词项,避免了其对文本聚类的影响。

b)对文本内部出现的同义词进行一致性处理。比如,在同一文本的上下文中可能同时出现“土豆”和“马铃薯”两个意思完全相同的词项,会将它们进行合并,统一用“土豆”或“马铃薯”来表示这个概念。这样做的目的是为了节省文本之间计算词语语义相似度的时间开销。

c)对文本中所有的词项进行词性分析。因为最能表征文本含义的主要是文本中的实词,所以需要给出所有词项的语义属性,即该词项是名词、动词、形容词还是副词等。

2.2 特征项选择

文本预处理结束后,接下来要对整个文本集合中的每一篇文本的词项进行 TF-IDF 值的计算,并将文本中各个词项的 TF-IDF 值表示为一个向量,以此进行文本的相似度计算。这样得到的向量维度非常高而且稀疏,因此需要对其进行降维处理。本文采取的方法是从每一篇文章中挑选若干关键词项来表示文本,这样就可以做到在保证不影响文本特征提取的前提下,最大可能地减少文本特征向量的表示维度。那么哪些词项才算关键词项呢?通过分析语料发现,可以表达文本主要意思

的是句子的主干成分，而主干成分主要由名词、动词构成，所以选择名词和动词作为关键词项。降维处理时将每一篇文本中词项的 TF-IDF 值进行排序，然后从中选取 TOP P（P 为百分比）的关键词项作为文本的特征表示。与传统的 TF-IDF 方法对比，维度下降了 $1-P$，这样大大提高了效率。

3　文本相似度计算

在得到了每篇文本的特征向量后，接下来要考虑如何计算两篇文本之间的相似度。由于一篇文本由特征词项来表示，因此文本的相似度就可以由特征词项向量间的相似度来描述。

设 v_i、v_j 是两篇不同文本的特征词项向量。$v_i=(\omega_{i1},\omega_{i2},\cdots,\omega_{im})$，$v_j=(\omega_{j1},\omega_{j2},\cdots,\omega_{jn})$，定义文本的相似度为

$$\text{textSim}(v_i,v_j)=wf\times\text{vectSim}(v_i,v_j)+(1-wf)\cos\text{Sim}(v_i,v_j) \quad (4)$$

其中：wf 表示关键词向量 v_i 和 v_j 之间相似度的加权因子；vecSim(v_i，v_j）表示关键词向量 v_i 和 v_j 之间的语义相似度，由式(8)计算得出。

$$\cos\text{Sim}(v_i,v_j)=\sum_{k=1}^{\lambda}\text{TFIDF}(\omega_{ik})\times\text{TFIDF}(\omega_{jk})\Big/\sqrt{\sum_{k=1}^{m}(\text{TFIDF}(\omega_{ik}))^2\times\sum_{l=1}^{n}(\text{TFIDF}(\omega_{jl}))^2} \quad (5)$$

式(5)表示向量的 v_i 和 v_j 之间的余弦相似度，其中 λ 指向量 v_i 和 v_j 中出现的相同词项的数目（可以使向量 v_i 和 v_j 表示为相同的维数，本文为了体现公式的通用性，没有进行这样的处理）。

本文基于这样的假设来推导公式，如果两篇文本中彼此相似度较高的词项越多，那么这些词项所占的 TF-IDF 值在各自文档中的比例越高，说明计算这些词项的语义相似度更能反映文本的相似情况。而剩余的词项由于语义相似度偏低，再通过计算语义相似度来得出的文本相似情况可信度不高，但可以利用它们在整个文本集合中的概率分布情况反映相似度。因此需要计算 vecSim(v_i，v_j）的加权因子，而加权因子根据关键词向量中满足相似度阈值条件的关键词的 TF-IDF 值在整篇文本 TF-IDF 值总和中所占的比例计算得到。具体的加权因子计算式由式(6)给出：

$$wf=\frac{1}{2}\left(\frac{\sum_{k\in\Lambda_i}\text{TFIDF}(\omega_{ik})}{\sum_{k=1}^{m}\text{TFIDF}(\omega_{ik})}+\frac{\sum_{l\in\Lambda_j}\text{TFIDF}(\omega_{jl})}{\sum_{l=1}^{n}\text{TFIDF}(\omega_{jl})}\right) \quad (6)$$

其中，TFIDF(ω_{ik}）表示关键词词项 ω_{ik} 的 TF-IDF 值，右端表示关键词向量 v_i 中所有满足相似度阈值条件的关键词项 ω_{ik}（$k\in\Lambda_i$）的 TF-IDF 值在 v_i 所有的词项 TF-IDF 值总和中所占的百分比。式(6)中的集合 Λ_i 和 Λ_j 定义如下：

$$\Lambda_i=\{k:1\leqslant k\leqslant m,\max_{1\leqslant l\leqslant n}\{\text{sim}(\omega_{ik},\omega_{jl})\}\geqslant\mu\}$$
$$\Lambda_j=\{l:1\leqslant l\leqslant m,\max_{1\leqslant k\leqslant n}\{\text{sim}(\omega_{jl},\omega_{ik})\}\geqslant\mu\} \quad (7)$$

如果关键词向量 v_i 中的某个关键词 ω_{ik} 与另一个关键词向量 v_j 中的关键词 ω_{jl}（$l=1,2,\cdots,n$）的相似度超过用户设定的相似度阈值 μ，则将该关键词 ω_{ik} 放入集合 Λ_i。同理集合 Λ_j 中的元素依据集合 Λ_i 的方法对关键词向量 v_j 中的关键词进行选择。

$$\text{vecSim}(v_i,v_j)=\frac{1}{2}\left(\frac{1}{m}\sum_{k=1}^{m}\max_{1\leqslant l\leqslant n}\{\text{sim}(\omega_{ik},\omega_{jl})\}+\frac{1}{n}\sum_{l=1}^{n}\max_{1\leqslant k\leqslant n}\{\text{sim}(\omega_{jl},\omega_{ik})\}\right) \quad (8)$$

其中：sim(ω_{jl}，ω_{ik}）表示关键词 ω_{ik}、ω_{jl} 之间的语义相似度，由式(3)计算得到；vecSim(v_i，v_j）由向量 v_i、v_j 中所包含的词项语义相似度决定，相似的向量必定包含相似度较高的词项，而不相似的向量则彼此所包含的词项相似度较低。

算法 1　语义分析与词频统计相结合的相似度算法

输入：关键词项向量 v_i、v_j 的词项相似度阈值 μ。

输出：关键词项向量 v_i、v_j 的相似度。

a）从向量 v_i 中的词项 ω_{i1} 开始，利用式(3)寻找向量 v_j 中与 ω_{i1} 最为相似的词项 ω_{jk}（即 sim(ω_{i1}，ω_{jk}）词项语义相似度取得最大值），记录词项 ω_{i1} 和 ω_{jk} 之间的相似度，同时判断 sim(ω_{i1}，ω_{jk}）是否大于等于阈值 μ，如果是将 ω_{i1} 放入集合 Λ_i。同理，v_i 中的其他项作相同处理。

b）累加 v_i 中每个项的相似度，除以向量 v_i 中词项的数量，即向量 v_i 的维度，以此作为向量 v_i 和 v_j 的相似度 sim(v_i，v_j）。重复步骤 a）b）的过程，得到向量 v_j 和 v_i 的相似度 sim(v_j，v_i）。

c）计算 sim(v_i，v_j）和 sim(v_j，v_i）的算术平均值，作为向量 v_i 和 v_j 的语义相似度 vecSim(v_i，v_j）。

d）利用式(1)分别为向量 v_i 和 v_j 中的词项计算 TF-IDF 权值，利用式(5)计算向量 v_i 和 v_j 之间的余弦相似度。

e）由于在前面的步骤中已经分别找出了集合 Λ_i 和 Λ_j 中的元素，因此利用式(6)计算加权因子 wf。

f）根据前述一系列步骤，利用式(4)最终得出向量 v_i 和 v_j 之间的文本相似度。

4　实验设计与结果分析

4.1　实验数据

实验数据来源于知网期刊论文，人工收集了 500 篇论文，其中涉及的领域有计算机、机械、电子、航空、化工、物理总共六类文本集合，为方便叙述将这些文本集合称之为数据集。实验中选取数据集中的一些子集用于实验验证，表 1 总结了实验中所用的实验数据摘要。

表 1　实验数据摘要

数据集名称	聚类数目	总的文本数目	聚类中最少文本数目	聚类中最多文本数目	平均聚类文本数目
计算机	6	110	8	20	16
机械	8	105	10	15	13
电子	8	106	7	20	13
航空	5	69	8	16	12
化工	5	60	10	15	11
物理	5	50	8	14	10

实验中首先采用自然语言处理工具 ICTCLAS 对文本集合进行预处理，包括对文本进行分词和词性标注，之后识别文本集合中的人名、地名、组织机构；然后应用 TF-IDF 算法对文本中的所有词项进行权值计算，从中选择特定比例的 TOP 关键词项；再结合本文提出的文本相似度计算算法对实验文本进行相似度计算，这样就会得到文本的相似度矩阵。接下来用得到的相似度矩阵进行聚类。为了证实提出的方法更加有效，同时实现了基于 TF-IDF 得到相似度矩阵进行文本聚类，以及文献[12]提出的基于语义（本文称之为 SemanticSim）得到相似度矩阵进行聚类（文献[12]是针对英文语料采用 WordNet 进行语义相似度计算，本文借助它的实验参数，采用 HowNet 进行了 SemanticSim 实验：5 级上位关系的词项扩展、利用同义词项频率代替词项频率以及词义消歧）的算法，以便对这三种聚类效果进行比较。

实验中为了更客观地反映本文提出的文本相似度算法的有效性，聚类算法的实现采用了 CLUTO[13] 工具包。实验对比了 CLUTO 工具包实现的直接 K-均值（DKM）、二分 K-均值（BKM）以及凝聚 K-均值（AKM）聚类算法。

实验中采用 F-度量值来衡量本文提出的文本相似度。F-度量值是信息检索中一种组合查准率和召回率指标的平衡指标,综合利用查准率、查全率以及 F-度量值可以判断每一篇文本在聚类后是否被正确划分到了所属类别,因此,可以计算每一个聚类 j 所属类别 i 的查准率 $P(i,j)$ 及查全率 $R(i,j)$。查准率 $P(i,j)=n_{ij}/n_j$,n_j 是类别 j 的文本数目,n_{ij} 是聚类 j 中隶属于类别 i 的文本数目。同理,可以得到查全率 $P(i,j)=n_{ij}/n_i$,F-度量值 $F(i,j)=\dfrac{2\times P(i,j)\times R(i,j)}{P(i,j)+R(i,j)}$。

4.2　实验 1　特征项选择

实验中对本文提出的相似度算法中选取 TOP 关键词项的比例问题进行了研究。通过实验发现,选择不同比例的关键词项对文本相似度计算有不同的影响。实验中首先设定关键词项相似度阈值参数 $\mu=0$,即将所有词语的语义相似度看做同样重要的条件,选取 DKM 算法进行聚类。图 1 给出了选择不同比例的 TOP 关键词项对相似度影响的实验结果。实验表明选取文本中 60% 左右的 TOP 关键词项能够取得最好的聚类效果,低于这个比例,由于选取的关键词数目偏少,代表文本特征的信息不足,致使聚类效果欠佳;反之,选择过多的关键词项因引入了噪声项,降低了文本之间相似度计算的准确性。

4.3　实验 2　相似度阈值确定

在确定了 TOP 关键词项比例后,还要确定相似度阈值 μ 对文本相似度计算的影响。本文选择了 60% 的 TOP 关键词项作为文本特征向量,同样利用 DKM 算法进行聚类的条件下,研究同一聚类中的关键词相似度阈值 μ 的不同对聚类效果的影响。

由图 2 得知,随着 μ 的逐渐升高,聚类效果也逐步提升,这是因为随着相似度阈值的提高,文本之间的区分度越来越大,使得聚类效果越来越好,尤其是 $\mu\in[0.7,0.75]$ 时,聚类效果达到最优值,但当相似度阈值超过 0.75 时,继续提高 μ 反而聚类效果下降了。这是因为本文选择使用的基于 HowNet 的词语语义相似计算很少能够有超过 0.75 的相似度值,导致了 F-度量值的下降。

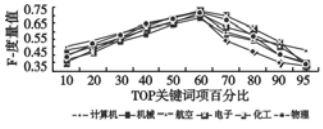

图1　TOP关键词百分比对DKM聚类算法效果的影响

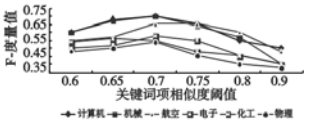

图2　相似度阈值μ对DKM聚类算法效果的影响

4.4　实验 3　三种文本聚类算法的比较实验

实验设定 TOP 关键词项百分比为 60%,$\mu=0.70$ 的条件下,采用本文提出的文本相似度计算方法分别进行 DKM、BKM、AKM 聚类实验,实验效果如图 3 ~5 所示。

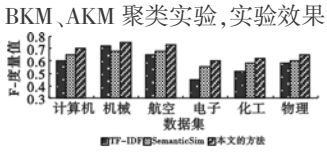

图3　三种相似度计算方法实现DKM聚类算法的比较

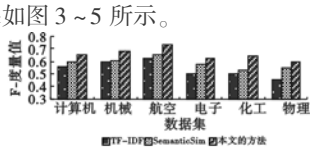

图4　三种相似度计算方法实现BKM聚类算法的比较

从图 3 ~5 可以看出,采用本文提出的文本相似度算法在三种经典的聚类算法下进行聚类都比传统的 TF-IDF 算法以及 SemanticSim 算法具有更好的 F-度量值。这是因为本文提出的文本相似度算法选择性地吸收了另外两种相似度算法的优点,并有效规避了它们的缺点。因此通过聚类实验有效证明,本文提出的文本相似度计算方法比只采用 TF-IDF 方法或词语语义方法得出的文本相似度值更加有效、准确。

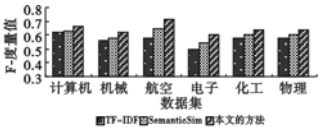

图5　三种相似度计算方法实现AKM聚类算法的比较

5　结束语

本文针对基于统计学以及语义分析进行文本相似度计算时存在的缺陷,提出了一种新颖的文本相似度计算方法。与传统的基于向量空间模型进行文本相似度计算不同,该方法不再仅借助于词语的表面信息进行相似度计算,而是从词语的语义相似度、词语在文本中的概率分布两方面综合衡量文本之间的相似度。同时本文考虑到传统的 TF-IDF 算法在对大量文本进行向量表示时会致使维度很高,而且极度稀疏的问题,对其采取了降维处理。经聚类实验证明,本文的方法是有效的。

后续的工作将在现有探讨词项相似性、词频统计对文本相似度量影响的基础上,进一步深入分析文本相似度所蕴涵的语义相似特征,考虑文本的段落、篇章等结构信息,更好地提高文本相似度计算的效果。

参考文献:

[1] KUMAR N. Approximate string matching algorithm[J]. International Journal on Computer Science and Engineering, 2010, 2(3): 641-644.

[2] COELHO T A S, CALADO P P, SOUZA L V, *et al*. Image retrieval using multiple evidence ranking[J]. IEEE Trans on Knowledge and Data Engineering, 2004, 16(4): 408-417.

[3] KO Y, PARK J, SEO J. Improving text categorization using the importance of sentences[J]. Information Processing and Management, 2004, 40(1): 65-79.

[4] THEOBALD M, SIDDHARTH J. SpotSigs: robust and efficient near duplicate detection in large Web collection[C]//Proc of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM Press, 2008: 563-570.

[5] 刘群,李素建. 基于《知网》的词汇语义相似度计算[C]//第三届汉语词汇语义学研讨会论文集. 2002: 59-76.

[6] 李素建. 基于语义计算的语句相关度研究[J]. 计算机工程与应用, 2002, 38(7): 75-78.

[7] 董振东. 知网[EB/OL]. (2003). http://www.keenage.com.

[8] 车万翔,刘挺,秦兵,等. 面向双语句对检索的汉语句子相似度计算[C]//全国第七届计算语言学联合学术会议. 北京: 清华大学出版社, 2003: 81-88.

[9] PATWARDHAN S, BANERJEE S, PEDERSEN T. Using measures of semantic relatedness for word sense disambiguation[C]//Proc of the 4th International Conference on Intelligent Text Processing and Computational Linguistics. 2003: 301-308.

[10] MILLER G. WordNet: a lexical database for English[J]. Communications of the ACM, 1995, 38(11): 39-41.

[11] SALTON G. The SMART retrieval system-experiments in automatic document processing[M]. Upper Saddle River: Prentice-Hall, 1971: 207-214.

[12] HOTHO A, STAAB S, STUMME G. WordNet improves text document clustering[C]//Proc of SIGIR Semantic Web Workshop. New York: ACM Press, 2003: 505-514.

[13] KARYPIS G. CLUTO: a clustering tookit[R]. Minneapolis: University of Minnesota, 2002.