基于语义词典和词频信息的文本相似度计算

董 苑 钱丽萍

(浙江工业大学计算机科学与技术学院 杭州 310023)

摘 要 为了克服传统的文本相似算法缺乏综合考虑语义理解和词语出现频率的缺点,在基于语义词典的词语相似度计算的基础上,提出了一种基于语义词典和词频信息的文本相似度(TSSDWFI)算法。通过计算两文本词语间的扩展相似度,找出文本词语间最大的相似度配对,从而计算出文本间的相似度。这种相似度计算方法利用语义词典,既考虑了不同文本间词语的相似度关系,又考虑了词语在各自文本中的词频高低。实验结果表明,与传统的语义算法和基于空间向量的文本相似度计算方法相比,TSSDWFI算法计算的文本相似度的准确度有了进一步提高。

关键词 文本挖掘,文本相似度,语义词典,关键词,词频

中图法分类号 TP391 **文献标识码** A

Text Similarity Calculation Based on Semantic Dictionary and Word Frequency Information

DONG Yuan QIAN Li-ping

(Department of Computer Science & Technology, Zhejiang University of Technology, Hangzhou 310023, China)

Abstract Considering the drawbacks of semantic understanding and frequent word appearance, this paper proposed a text similarity algorithm based on semantic dictionary and word frequency information, referred to as TSSDWFI. In particular, the proposed algorithm aims at evaluating the similarity between two texts by calculating the expanded similarity between any two words in texts and the maximum similarity matching between text words. The proposed algorithm adopts semantic dictionary to calculate similarity between texts and takes into account the similarity relationship between different words and the frequency of word appearance in the text. Simulation results show that, compared with the existing algorithms, the proposed algorithm TSSDWFI has higher accuracy.

Keywords Text mining, Text similarity, Semantic dictionary, Keywords, Word frequency

1 引言

随着大数据时代的来临和“互联网+”行动计划的推进,各行各业都加入到互联网大家庭中。大量的行业数据增加了对文本信息的处理和分析的困难。大量的文本不仅造成了存储空间的冗余,而且也加大了重要信息提取的难度。因此人们需要对这些文本进行深入挖掘分析。文本相似度就是文本挖掘领域中的一个重要概念。语言文字存在的天然关系决定了文本之间必然有其相通之处。因此,对文本之间的相似度进行计算,可以对文本进行比较分析、归类总结、分析预测、离群处理等操作。

文本相似度用于描述文本之间的相似程度,文本之间的相似性又能展现出文本之间的相同性或相异性。因此对文本相似度的计算,是文本挖掘领域一个重要的方面。传统的文本相似度计算一般采用向量空间模型[1],通过计算向量间的夹角余弦[2]或使用贝叶斯算法等[3]方法来计算文本间的相似度。郭庆琳等人[4]提出了一种基于VSM(Vector Space Model)的文本相似度计算方法,对TF-IDF算法[5]进行了改进,利用DF(文档频率)算法[6]线性复杂性的特点,提高了计算的精确度。目前常用的文本间相似度计算方法是基于统计特征的文本相似度计算方法。文献[7]提出了基于VSM权重改进的文本相似度计算方法,它通过计算句子关键词之间的距离来比较句子间的相似度,从而获得文本之间的相似度。文献[8]提出结合VSM余弦算法和SimHash来实现相似度的计算。但是基于统计特征的方法存在着一个很明显的问题,即计算过程非常复杂,需要对文本的特征值进行规范化和消除歧义等处理,因此计算的时间复杂度与空间复杂度相当高,效果也不甚理想。实验结果表明,余弦算法VSM由于其局限性,并不适合文本间的相似度计算。而词语之间天然存在的语义联系,使得语义之间的相似度可以很好地表达文本之间的相似度。为了自动消除歧义、合并同近义词,刘杰等[9]提出了一种基于《知网》的词汇语义计算方法,在吴健等[10]提出的基于本体论的基础上,利用最大匹配算法获取文本相似度。文献[11]基于《知网》的语言组织特点,提出了一种改进的语言相似度计算方法,通过义原深度的主次关系来计算词语之间的相似度,使计算结果更加精确。肖志军[12-13]等提出了基于《知网》义原空间的文本相似度计算,利用义原向量夹角计算文本相似度,然后使用文本聚类的方法对算法进行实验,表明基于语义的相似度计算优于基于统计特征的相似度计算。文献[14]提出了基于《知网》的文本相似度研究方法,利用《知网》计算词语的相似度,对VSM算法进行改进,只用少量特征值的TF/IDF值作为VSM向量中的权重,以降低维度、提高计算准确率。基于《知网》语义的方法虽然使用了语义词典,但实质上它还是利用统计特征的方法来获取文本间的相

董 苑(1990-),女,硕士生,主要研究方向为大数据分析、数据挖掘,E-mail:9973347@qq.com;钱丽萍(1981-),女,博士,副教授,主要研究方向为无线网络通信、网络优化、大数据分析。

似度，虽然在一定程度上达到了歧义的消除，但实际效果并不佳。文献[15-17]利用 LDA 模型为文本建模，利用聚类的方法来判断文本之间的相似关系。

近年来，随着语义词典《同义词词林》的出现，更好的词语编码和更多的同近义词组合使得对文本的挖掘和对相似度的计算更为方便。文献[18]提出了一种词语相似度计算方法，根据《同义词词林》对词语编码，选择适当的特征值和权重，来计算词语之间的相似度。文献[19]在向量空间模型的基础上，利用维基百科知识库计算语义相关度，使用 K-mean 方法对文本进行聚类分析，从而获得文本的相似关系。本文在《同义词词林》词语相似度计算的基础上，结合词频信息，提出了一种计算文本的相似度算法，即基于语义词典和词频信息的文本相似度算法(Text Similarity Based on Semantic Dictionary and Word Frequency Information，TSSDWFI)。

2　基本概念定义

2.1　词语相似度

词语相似度展示了词语间的语义关系，通常是用语义距离来表示词语之间的关联性大小。本文采用哈工大《同义词词林(扩展版)》[20]来计算词语之间的语义相似度。哈工大的《同义词词林(扩展版)》中的每个词都含有若干个编码，每个编码是由 5 层代码和 1 个标志位构成。其编码格式如下：

$$Code_i = C_{i1}C_{i2}C_{i3}C_{i4}C_{i5}F_i \tag{1}$$

其中，$Code_i$ 表示第 i 个字的编码；C_{i1} 表示第 i 个字的大类编码；C_{i2} 表示第 i 个字的中类编码；C_{i3} 表示第 i 个字的小类编码；C_{i4} 表示第 i 个字的词群编码；C_{i5} 表示第 i 个字的原子词群编码；F 表示标志位，由“=”“#”“@”组成。

定义 1　假设词语 W1 和 W2 在《同义词词林(扩展版)》的编码分别为 $Code_{11}$，$Code_{12}$，$Code_{13}$，…，$Code_{1m}$ 和 $Code_{21}$，$Code_{22}$，$Code_{23}$，…，$Code_{2n}$，则它们之间的距离定义为：

$$Distance(W_i,W_j)=\min_{i=1,2,\cdots,m,j=1,2,\cdots,n} Dis(Code_{1i},Code_{2j}) \tag{2}$$

根据 Agirre E[21]提出的概念层次树的深度模型，当两个节点路径长度相同时，如果它们所处的概念层次越高，那么它们之间的语义距离就越大。假设 W_1 和 W_2 的编码是从第 i 层不相同，则 i 越小，二者语义距离将越大。基于不同层次间词义距离的大小，为每个层次分配不同的权重系数，从而更好地计算词语间的相似度。此处采用文献[22]中设定的基数，编码差异出现的层数为 i，定义权重比例 $weight_i$ 依次为 0.65，0.8，0.9，0.96，0.5，0.1，词语 W_1 和 W_2 的相似度 $Sim(W_1,W_2)$ 为：

$$Sim(W_1,W_2)=\begin{cases} f, & i=1 \\ weight_i\times\cos\left(n\times\frac{\pi}{180}\right)\left(\frac{n-d+1}{n}\right), & i=1,2,3,4 \end{cases} \tag{3}$$

其中，n 为分支层节点总数，d 为 W_1 和 W_2 两个分支之间的距离。

2.2　扩展文本相似度

关键词之间初始相似度值只能代表词语之间的相互联系。对于文本来说，文本之间的相似度不仅与关键词相关，还与关键词出现的次数有关。例如，对于文本 C 和文本 D 来说，它们的关键词完全相同，但是每个关键词在文本中所占的比例不一样，因此不能判定文本 C 和文本 D 是相同的文章，而只能归类到相似文章中去。因此对于文本关键词间的相似度计算，本文还考虑了其在各自文本中所占的比例。

定义 4　对于文本 T 而言，每个关键词在初始关键词中所占的权重大小为 $factorT[i]$，定义为：

$$factorT[i]=\frac{times[i]}{\sum_{i=0}^{n} times[i]} \tag{7}$$

其中，n 为关键词的个数。每一列定义为文本 A 的关键词 KWA_i 的文本相似度链，每一行定义为文本 B 的关键词

2.3　初始文本相似度

中文文本由词语组成，每篇中文文本都有其主旨内容。关键词是一篇文章的主旨，也是对文本内容高度概括的词语。比较两个文本之间的相似度可以转化为比较两个文本的关键词之间的相似度。通过计算文本关键词之间的相似度来判断文本之间的相似性，既降低了存储空间，又降低了计算时间。文中采用中科院的 NLPIR 汉语分词系统对文本进行分词，根据词频高低获取文本的关键词。

对于测试样本中的文本 T，可以表示为 $T=(KW_1,KW_2,KW_3,\cdots,KW_n)$，其中 n 是关键词个数。对于文本 A 和文本 B 而言，分别表示为：

$$TA=(KWA_1,KWA_2,KWA_3,\cdots,KWA_n) \tag{4}$$

$$TB=(KWB_1,KWB_2,KWB_3,\cdots,KWB_n) \tag{5}$$

文本 A 和文本 B 之间的相似度 $Sim(TA,TB)=Sim(KWA,KWB)$，其中 KWA 和 KWB 表示文本 A 和文本 B 的关键词集合。

定义 3　文本 A 和文本 B 的关键词两两间的相似度 $Sim(KWA,KWB)$ 定义为：

$$\begin{bmatrix} Sim(KWA_1,KWB_1) & Sim(KWA_2,KWB_1) & \cdots & Sim(KWA_n,KWB_1) \\ Sim(KWA_1,KWB_2) & Sim(KWA_2,KWB_2) & \cdots & Sim(KWA_n,KWB_2) \\ \vdots & \vdots & \cdots & \vdots \\ Sim(KWA_1,KWB_n) & Sim(KWA_2,KWB_n) & \cdots & Sim(KWA_n,KWB_n) \end{bmatrix} \tag{6}$$

定义 5　设文本 A 和文本 B 的关键词频分别为 $factor(A)$ 和 $factor(B)$，则它们的关键词的扩展相似度 $exSim[i][j]$ 定义为：

$$exSim(KWA_i,KWB_j)=0.5*Sim(KWA_i,KWB_j)*(factorA[i]+fatorB[j]) \tag{8}$$

其中，i 为文本 A 中的关键词序号，j 为文本 B 中的关键词序号。则文本 A 和文本 B 关键词之间的扩展相似度 $exSimAB$ 定义为：

$$\begin{bmatrix} exSim(KWA_1,KWB_1) & exSim(KWA_2,KWB_1) & \cdots & exSim(KWA_n,KWB_1) \\ exSim(KWA_1,KWB_2) & exSim(KWA_2,KWB_2) & \cdots & exSim(KWA_n,KWB_2) \\ \vdots & \vdots & \cdots & \vdots \\ exSim(KWA_1,KWB_n) & exSim(KWA_2,KWB_n) & \cdots & exSim(KWA_n,KWB_n) \end{bmatrix} \tag{9}$$

KWB_j 的文本相似度链。

3 基于语义词典和词频信息的文本相似度计算

3.1 算法思想描述

每个文本都有若干个关键词，每个关键词与另一篇文本的关键词之间都各有联系与差异。对于文本 A 和文本 B 来说，文本 A 的每一个关键词 KWA_i 都与文本 B 的关键词 KWB 有联系。找出两个文本之间的相似关系，就是找出其中一个文本中能代表该文本主旨的关键词，而且该关键词与另一个文本关键词间的相似度最高。这样，文本之间的相似度就转化为寻找对内代表文本主旨，对外代表文本间联系的词语间的相似度总和。因此对外来说，就是计算一个文本中每个关键词对应于其他文本中每个关键词之间的相似度总和；对内就是找出这个相似度总和所对应的关键词文本相似度链中的最大值。

例如，文本 A 和文本 B 的关键词的扩展相似度如表 1 所列，其中选取的关键词个数和扩展相似度数值仅作为选取关键词匹配计算过程的示例。

表 1 文本 A 和文本 B 的关键词的扩展相似度

关键词	$A1$	$A2$	$A3$	$A4$	$A5$
$B1$	0.4	0.4	0.3	0	0
$B2$	0.2	0.7	0.2	0.7	0
$B3$	0.15	0.5	0.3	0.7	0.4
$B4$	0.1	0.2	0.1	0.3	0.4
$B5$	0.63	0.5	0.3	0.7	0.85
allExSim	1.48	2.3	1.6	2.4	1.65

其中，$A1$，$A2$，$A3$，$A4$，$A5$ 是文本 A 的关键词，$B1$，$B2$，$B3$，$B4$，$B5$ 是文本 B 的关键词，通过计算获得扩展相似度和 *allExSim*。选择相似度最高的两个关键词并不能代表两篇文章的相似度，例如 $A5$ 和 $B5$ 的扩展相似度虽然最高，但是关键词 $A5$ 和 $B1$，$B2$ 的扩展相似度都为 0；而关键词 $A2$，$A4$ 虽然单个关键词的扩展相似度不大，但是综合起来，扩展相似度和的值却非常高，这代表关键词与文本之间的联系更大，更应该作为文本之间相关程度的考虑。选择相似度也不能仅依靠文本中关键词出现的频率从大到小来进行选择，例如文本 A 中第一个关键词 $A1$，与文本 B 的关键词扩展相似度总和并不高。

从表 1 中可以看出，$A4$ 的关键词扩展相似度和最高，表明该关键词与文本 B 的相关性越大，虽然它在文本 A 中出现的频率排在第四位，但更能代表文本 A 与文本 B 之间的联系，因此优先选择关键词 $A4$，即 $i=4$。在 $A4$ 与文本 B 关键词的相似度中，$A4$ 和 $B2$，$B3$，$B5$ 的相似度都是 0.7，但是 $B2$ 在文本 B 中所占比例大，因此计算 $A4$ 和 $B5$ 的相似度，该数值排在第一位。*allExSim* 中，$A2$ 关键词对应的扩展相似度最高，由于在上一轮计算中去除了 $B2$ 关键词，因此在剩下的关键词中，$A2$ 与 $B3$ 的相似度最高。以此类推，直至所有的关键词都被选择。最终选择的扩展相似度取值依次为以下匹配：$A4B2$，$A2B3$，$A5B5$，$A3B1$，$A1B4$。使用此最大相似度的方法可以最大程度地挖掘出不同文本之间的相关性。

算法的流程如图 1 所示。

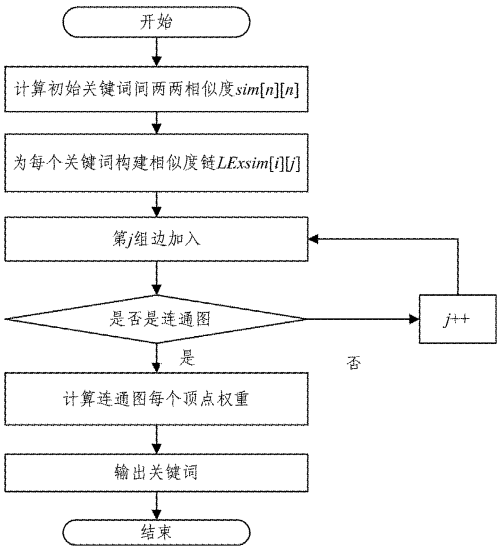

图 1 TSSDWFI 算法的流程

3.2 算法步骤

基于语义词类和词频信息的文本相似度计算的步骤如下。

Step1 提取两个文本的关键词，假定提取的关键词个数为 n，那么文本 A 和文本 B 的关键词分别定义为：

$$KWA=\{KWA_1,KWA_2,KWA_3,\cdots,KWA_n\} \tag{10}$$

$$KWB=\{KWB_1,KWB_2,KWB_3,\cdots,KWB_n\} \tag{11}$$

初始化一个 *ignore* 列表为空集，用于存放后面选取的 i，j。定义文本 A 与文本 B 之间的最终相似度为 *txtSim*，初始化为 0。

Step2 计算文本 A 和文本 B 关键词之间的扩展相似度 $exSim[i][j]$，计算式(8)，则文本 A 对于文本 B 的扩展相似度 $exSim[n][n]$ 定义为：

$$\begin{bmatrix} exSim(KWA_1,KWB_1) & exSim(KWA_2,KWB_1) & \cdots & exSim(KWA_7,KWB_1) \\ exSim(KWA_1,KWB_2) & exSim(KWA_2,KWB_2) & \cdots & exSim(KWA_7,KWB_2) \\ \vdots & \vdots & \cdots & \vdots \\ exSim(KWA_1,KWB_n) & exSim(KWA_2,KWB_n) & \cdots & exSim(KWA_7,KWB_n) \end{bmatrix} \tag{12}$$

Step3 计算文本 A 中第 i 个关键词与文本 B 的每个关键词扩展相似度之和，定义为 $allExSimA[i]$，计算式如下：

$$allExSimA[i]=\sum_{j=1}^{7} exSim[i][j] \tag{13}$$

其中，i 是文本 A 的关键词序号，j 是文本 B 的关键词序号。则文本 A 的关键词文本相似度集合记作 $allExSimA$，那么：

$$allExSimA=\{allExSimA[1],allExSimA[2],\cdots,allExSimA[n]\} \tag{14}$$

定义文本 A 对文本 B 的文本相似度为 *txtSimA*，初始化为 0。

Step4 选择扩展相似度和 *allExSimA* 中最大值对应文本 A 的 i，记录为 p，根据 i 找到 $exSim[p][q]$ 中最大值对应文本 B 的 j，记录为 q，将这个最大值 $exSim[p][q]$ 加到 *txtSimA* 中。将此时的 p，q 加入 *ignore* 列表中。

Step5 判断 *ignore* 列表中的数值，凡是与 i 和 j 相关的 $exSim[i][j]$ 项都不再取出。重复 Step4，直到 *ignore* 列表中包含了所有的 i 和 j 的位置，此时的 *txtSimA* 即为文本 A 对于文本 B 的文本相似度。

Step6　对文本 B 同样重复 Step3 — Step5，得到 $txtSimB$。

Step7　计算 $txtSimA$ 和 $txtSimB$ 的平均值，即为文本 A 和文本 B 的最终相似度 $txtSim$。

4　算法测试与分析

本文采用复旦大学李荣陆[22]提供的文本分类语料库作为数据集。其中有 20 个类别，共 9833 个文件。分别为 Art，Literature，Education，Philosophy，History，Space，Energy，Electronics，Communication，Computer，Mine，Transport，Environment，Agriculture，Economy，Law，Medical，Military，Politics 和 Sports 类，每个类的文本数目各不相同。文中选取其中 7 个作为测试对象。

实验 1　随机选取语料库中不同的 7 个类别，分别计算类中文本之间的相似度平均值和文本间相似度不为 0 的关键词个数，改进的语义相似度算法 TSSDWFI、普通的语义相似度算法以及空间向量的相似度算法的结果如图 2－图 4 所示，其中图 2 给出了文本间的平均相似度，图 3 给出了重复实验 7 次所取的平均值，图 4 给出了文本间关键词相似度非 0 的关键词个数。

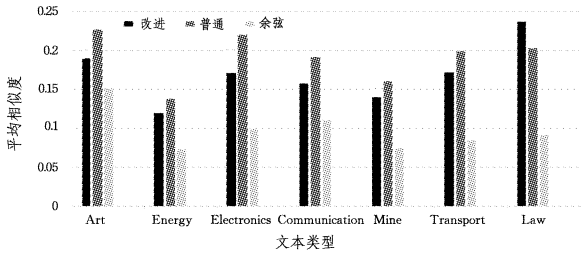

图 2　3 种算法的平均相似度值

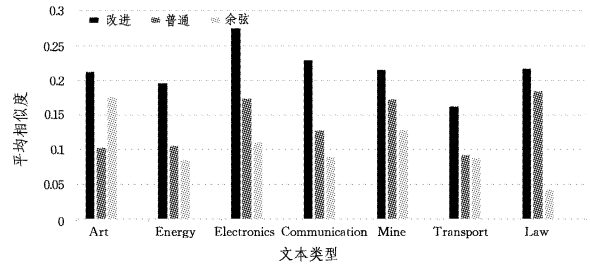

图 3　3 种算法的平均相似度均值

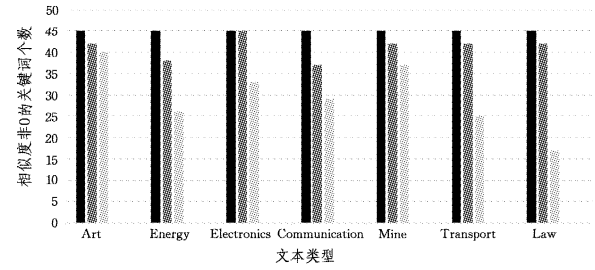

图 4　3 种算法文本间关键词相似度非 0 的关键词个数

同类文本之间的相似度表示一个类的相关程度，即同类文本间平均相似度越高，代表类之间的相似性越大。关键词之间的相似度不为 0，表示关键词匹配方式更容易挖掘出文本之间的相似关系。表 2 列出了文本 A 和文本 B 的关键词与它在各自文本中出现的概率和关键词之间的语义相似度；表 3 列出了由表 2 计算出的关键词间的扩展相似度。

表 2　文本关键词频与相似度表

关键词相似度	$A1/50\%$	$A2/30\%$	$A3/20\%$	$A4/10\%$
$B1/40\%$	0\1	0.7	0.3	0.8
$B2/30\%$	1	0\1	0.4	0.9
$B3/20\%$	0.2	0.6	0\1	0.9
$B4/10\%$	0	0.1	0.9	0\1

表 3　文本关键词扩展相似度表

关键词相似度	$A1$	$A2$	$A3$	$A4$
$B1$	0\0.45	0.245	0.06	0.2
$B2$	0.4	0\0.3	0.08	0.18
$B3$	0.07	0.15	0\0.15	0.135
$B4$	0	0.02	0.09	0\0.1
$exSim(AiBi)=0$ 时之和	0.47	0.415	0.23	0.515
$exSim(AiBi)=1$ 时之和	0.92	0.715	0.38	0.615

从表 2 和表 3 可以看出，在极端条件下，即当对应的 Ai 关键词和 Bi 关键词的相似度分别为 0 和 1 时，改进的语义相似度算法 TSSDWFI 的扩展相似度取值顺序分别为：$A4B1$，$A1B2$，$A2B3$，$A3B4$ 和 $A1B1$，$A2B2$，$A4B3$，$A3B4$，即扩展相似度之和为：0.2＋0.4＋0.15＋0.09＝0.84 和 0.45＋0.3＋0.135＋0.09＝0.975。普通的相似度算法的计算结果分别为 0 和 1。虽然当 Ai 关键词和 Bi 关键词的相似度在极端条件下都为 0，但其他关键词之间还是有关联的。从实验结果图 4 可以看出，关键词之间相似度计算为 0 的匹配是存在的，并且在一个样本中出现的次数不可忽略。因此计算样本间的平均相似度时，改进的语义相似度算法 TSSDWFI 结合语义与词频计算出来的数值相对稳定且准确。

实验结果表明，改进的语义相似度算法 TSSDWFI 的计算结果优于普通的语义相似度算法和向量空间的相似度算法。其原因在于：相较于普通的语义相似度算法而言，改进的语义相似度算法 TSSDWFI 摒弃了普通语义相似度算法逐词匹配的方式，通过词频与语义相结合，选择了更佳的关键词匹配策略计算相似度，从而找出文本间的相似关系，计算结果更稳定。相较于向量空间的相似度算法，改进的语义相似度算法 TSSDWFI 充分考虑到了语义之间的同义、近义关系，因此不会出现明明是同义词，计算出来的相似度却接近于 0，使得计算的结果整体偏低的现象。因此该文提出的改进方法在这一方面更加精确。

实验 2　选择一个类作为基础类，从每个类中选取一篇文本计算与基础类的平均相似度。本次实验选择 Law 类作为基础类，从 8 个类别中分别选出一篇文本与之计算平均相似度。测试结果如图 5 和图 6 所示。图 5 给出了样本与类之间的平均相似度，图 6 给出了样本与类之间的最高相似度。

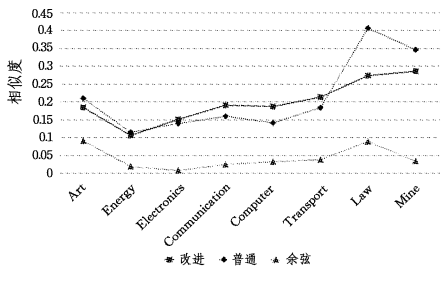

图 5　3 种相似度算法的单文本与类相似度平均值

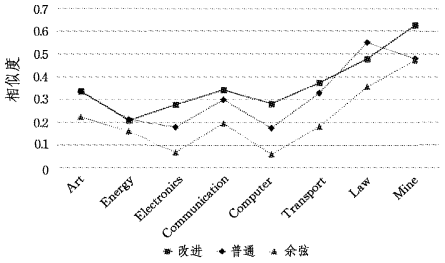

图 6 3 种相似度算法的单文本与类最高相似度值

从图 5 可以看出，改进的语义相似度算法 TSSDWFI，在 Mine 中取出的文本与所选的 Law 类相似度最高；而普通的语义相似度算法和向量空间的相似度算法都有明显的回落。从图 6 中可以看出，改进的语义相似度算法 TSSDWFI 和向量空间的相似度算法，在 Mine 中取出的文本与所选的 Law 类计算出的最高相似度都是最高的。通过测试，找出改进语义相似度算法中出现的最高值，如图 7 所示。

C23-Mine21. txt-C35-Law085. txt-0. 6221102659804485
C23-Mine21. txt-C35-Law066. txt-0. 4763805775853648
C23-Mine21. txt-C35-Law014. txt-0. 4679747003498
C23-Mine21. txt-C35-Law093. txt-0. 4673084579098524
C23-Mine21. txt-C35-Law103. txt-0. 4542121091948408
C23-Mine21. txt-C35-Law058. txt-0. 4379638848722498
C23-Mine21. txt-C35-Law062. txt-0. 4379494965445343
C23-Mine21. txt-C35-Law060. txt-0. 4379268704567433
C23-Mine21. txt-C35-Law068. txt-0. 4379268704567433
C23-Mine21. txt-C35-Law095. txt-0. 4379268704567433
C23-Mine21. txt-C35-Law087. txt-0. 4041538449186736
C23-Mine21. txt-C35-Law089. txt-0. 4041538449186736
C23-Mine21. txt-C35-Law083. txt-0. 3996492763947576
C23-Mine21. txt-C35-Law052. txt-0. 39480277533410635
C23-Mine21. txt-C35-Law030. txt-0. 374990495013244
C23-Mine21. txt-C35-Law097. txt-0. 3649766757422156
C23-Mine21. txt-C35-Law018. txt-0. 35198805809400313
C23-Mine21. txt-C35-Law026. txt-0. 34834461088212787
C23-Mine21. txt-C35-Law007. txt-0. 3456943591481694
C23-Mine21. txt-C35-Law064. txt-0. 3346792890453159

图 7 改进的语义相似度算法 TSSDWFI 的测试结果

C23-Mine21. txt-C35-Law014. txt-0. 47760808388102965
C23-Mine21. txt-C35-Law020. txt-0. 4729483564536469
C23-Mine21. txt-C35-Law010. txt-0. 4664974827908556
C23-Mine21. txt-C35-Law077. txt-0. 46646303321376914
C23-Mine21. txt-C35-Law062. txt-0. 46635754289375264
C23-Mine21. txt-C35-Law056. txt-0. 46277859286293915
C23-Mine21. txt-C35-Law066. txt-0. 4559691846001806
C23-Mine21. txt-C35-Law022. txt-0. 4426105207944419
C23-Mine21. txt-C35-Law030. txt-0. 4350798683447915
C23-Mine21. txt-C35-Law048. txt-0. 4350798683447915
C23-Mine21. txt-C35-Law028. txt-0. 4339315249199455
C23-Mine21. txt-C35-Law026. txt-0. 4338501745561948
C23-Mine21. txt-C35-Law018. txt-0. 4314019571215404
C23-Mine21. txt-C35-Law007. txt-0. 4309710653465212
C23-Mine21. txt-C35-Law058. txt-0. 42767225825812744
C23-Mine21. txt-C35-Law085. txt-0. 4211752593076382
C23-Mine21. txt-C35-LawL03. txt-0. 4149698292768219
C23-Mine21. txt-C35-Law042. txt-0. 40305582733520545
C23-Mine21. txt-C35-Law079. txt-0. 4030513214248466
C23-Mine21. txt-C35-Law060. txt-0. 4030392514467138

图 8 普通的语义相似度算法测试结果

C23-Mine21. txt-C35-Law085. txt-0. 470000000000000003
C23-Mine21. txt-C35-Law032. txt-0. 23540647259997272
C23-Mine21. txt-C35-Law020. txt-0. 21000000000000002
C23-Mine21. txt-C35-Law083. txt-0. 195
C23-Mine21. txt-C35-Law093. txt-0. 195
C23-Mine21. txt-C35-Law046. txt-0. 14566161603278333
C23-Mine21. txt-C35-Law058. txt-0. 0796084166404533
C23-Mine21. txt-C35-Law014. txt-0. 07
C23-Mine21. txt-C35-Law044. txt-0. 04666666666666667
C23-Mine21. txt-C35-Law034. txt-0. 04427188724235732
C23-Mine21. txt-C35-Law002. txt-0. 0
C23-Mine21. txt-C35-Law003. txt-0. 0
C23-Mine21. txt-C35-Law005. txt-0. 0
C23-Mine21. txt-C35-Law007. txt-0. 0
C23-Mine21. txt-C35-Law009. txt-0. 0
C23-Mine21. txt-C35-Law0L0. txt-0. 0
C23-Mine21. txt-C35-Law012. txt-0. 0
C23-Mine21. txt-C35-Law016. txt-0. 0
C23-Mine21. txt-C35-Law018. txt-0. 0

图 9 向量空间的相似度算法测试结果

其中，Law 类中文本序号为 85 的文本与 Mine 中选取的文本相似度最高。反观普通的语义相似度算法和向量空间的相似度算法，它们的测试结果分别如图 8 和图 9 所示，都没有出现如此突增的值。

提取文本 C23-Mine21. txt 和 C35-Law085. txt 的关键词，如表 4 所列。

表 4 文本关键词表

文本名	1	2	3	4	5
C23-Mine21	稀土 10	内蒙古 6	发展 4	资源 4	开发 4
C35-Law085	开发 9	活动 2	发展 1	有关 1	

从提取的关键词可以看出，C35-Law085. txt 中只提取到了 4 个关键词。查找原文本发现，这是一篇关于法律的文本，其内容是关于资源开发和发展活动的。对比 C23-Mine21 这篇文本，发现该篇文本的篇幅远小于前者，但是它具体描述了矿产资源的开发与利用问题，因此这两篇文本相似度自然偏高。改进的语义相似度算法 TSSDWFI 综合了普通语义相似度算法和向量空间的相似度算法的优点，既考虑了语义间同近义的关系，又考虑了词频等特征值的影响。

为了使样本更具有代表性，实验再选取 Mine 类中的另一篇文本与基础类之间的计算相似度，其结果如图 10 和图 11 所示。

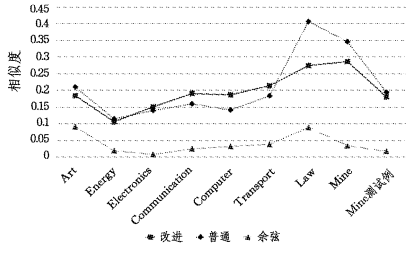

图 10 单文本与类相似度

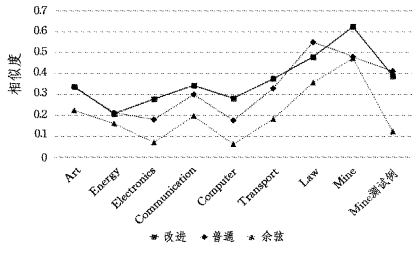

图 11 单文本与类最高相似度

此次实验 Mine 类测试例与基础类的相似度均有明显回落。重复实验 2 10 次，选择 8 个类，每个类中各选取一篇文本(10 次实验选取不同的文本)与 Law 类之间计算平均相似度，并取计算结果的平均值，其结果如图 12 所示。

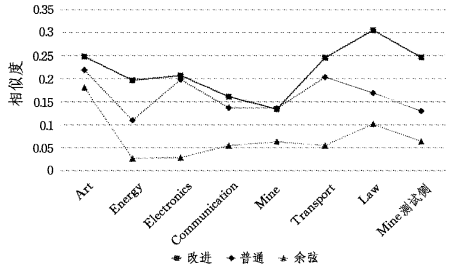

图 12　单文本与 Law 类的相似度平均值

实验结果表明，改进的语义相似度算法 TSSDWFI 很好地结合了词频与语义之间的联系，在挖掘文本之间的相似关系上具有更明显的优势，计算出的文本相似度结果更具有稳定性和准确性。

结束语　基于语义词典和词频信息的文本相似度计算与传统的方法相比，显示出了其优越性，有效地考虑了词语间的语义关系和词语间的比例关系。计算文本之间的相似度是文本挖掘过程中的一个重要步骤。现有的文本相似度计算方法大多数是利用特征空间的数值进行计算的，缺乏对词语之间语义相关性的考虑。本文在现有的词语相似度计算的基础上，提出一种改进算法，改进后的算法对文本间的相似度计算更加准确和稳定。在此基础上，可以进一步进行文本的归类以及文本中离群点的剔除工作。

参 考 文 献

[1] 陈飞宏. 基于向量空间模型的中文文本相似度算法研究[D]. 成都：电子科技大学，2011.

[2] 张振亚，王进，程红梅，等. 基于余弦相似度的文本空间索引方法研究[J]. 计算机科学，2005，32(9)：160-163.

[3] 吴奎，周献中，王建宇，等. 基于贝叶斯估计的概念语义相似度算法[J]. 中文信息学报，2010，24(2)：52-57.

[4] 郭庆琳，李艳梅，唐琦. 基于 VSM 的文本相似度计算的研究[J]. 计算机应用研究，2008，25(11)：3256-3258.

[5] 卫驰. 基于 TFIDF 的文本分类算法[D]. 杭州：浙江大学，2015.

[6] 冯荣俊. 基于文档频率的特征提取算法的改进及应用[D]. 南京：南京邮电大学，2005.

[7] 韩如冰，叶得学. 基于 VSM 的权重改进文档相似度算法研究[J]. 软件，2012，33(10)：103-105.

[8] 王格，吴钊，李向. 基于全文检索的文本相似度算法应用研究[J]. 计算机与数字工程，2016，44(4)：567-571.

[9] 刘杰，郭宇，汤世平，等. 基于《知网》2008 的词语相似度计算[J]. 小型微型计算机系统，2015，36(8)：1728-1733.

[10] 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[J]. 计算机学报，2005，28(4)：595-602.

[11] 张沪寅，刘道波，温春艳. 基于《知网》的词语语义相似度改进算法研究[J]. 计算机工程，2015，41(2)：151-156.

[12] 肖志军，冯广丽. 基于《知网》义原空间的文本相似度计算[J]. 科学技术与工程，2013，13(29)：8651-8656.

[13] 孙润志. 基于语义理解的文本相似度计算研究与实现[D]. 辽宁：中国科学院研究生院(沈阳计算技术研究所)，2015.

[14] 袁晓峰. 基于《知网》的文本相似度研究[J]. 成都大学学报(自然科学版)，2014，33(3)：251-253.

[15] 陈攀，杨浩，吕品，等. 基于 LDA 模型的文本相似度研究[J]. 计算机技术与发展，2016，26(4)：82-85.

[16] 王蒙. 基于 LDA 的文本推荐算法的研究及在文献检索的应用[D]. 沈阳：辽宁大学，2015.

[17] 王振振，何明，杜永萍. 基于 LDA 主题模型的文本相似度计算[J]. 计算机科学，2013，40(12)：229-232.

[18] 田久乐，赵蔚. 基于同义词词林的词语相似度计算方法[J]. 吉林大学学报(信息科学版)，2010，28(6)：603-608.

[19] 杜坤，刘怀亮，王帮金. 基于语义相关度的中文文本聚类方法研究[J]. 情报理论与实践，2016，39(2)：129-133.

[20] 《同义词词林扩展版》[EB/OL]. http://www.ir-lab.org.

[21] AGIRRE E，RIGAU G. A Proposal for Word Sense Disambiguantion Using Conceptual Distance[C]// Proc. of Recent Advances in NLP(RANLP). 1995：258-264.

[22] 李荣陆. Reuters-21578 语料说明[EB/OL]. http://more.datatang.com/data/43318.

(上接第 390 页)

[15] KEIM D，ANDRIENKO G，FEKETE J D，et al. Visual analytics：Definition，process，and challenges[M]. Springer Berlin Heidelberg，2008.

[16] PIKE W A，STASKO J，CHANG R，et al. The science of interaction[J]. Information Visualization，2009，8(4)：263-274.

[17] KAUFMAN L，ROUSSEEUW P J. Finding groups in data：an introduction to cluster analysis[M]. John Wiley & Sons，2009.

[18] EICK S G. Graphically displaying text[J]. Journal of Computational and Graphical Statistics，1994，3：127-142.

[19] STASKO J. Information visualization[OL]. http://www.cc.gatech.edu/ classes /AY2004/cs7450_spring.

[20] FENG Y D，WANG G P，DONG S H. Information Visualization[J]. Journal of Engineering Graphics，2001：324-329.

[21] SMITH M A，SHNEIDERMAN B，MILIC-FRAYLIN N，et al. Analyzing (social media) networks with NodeXL[C]// Proceedings of the Fourth International Conference on Communities and Technologies. ACM，2009：255-264.

[22] HENRY N，FEKETE J D. MatrixExplorer：a Dual-Representation System to Explore Social Networks[J]. IEEE Transactions on Visualization & Computer Graphics，2006，12(5)：677-684.

[23] FRUCHTERMANN T M J，REINGOLD E M. Graph drawing by force-directed placement[J]. Software：Practice and experience，1991，21(11)：1129-1164.