文章编号：1004-4353(2020)02-0140-05

基于编辑距离的词序敏感相似度度量方法

张雷， 崔荣一*
（延边大学 工学院，吉林 延吉 133002）

摘要：为改善余弦相似度不能反映词袋模型中词项间顺序差异的缺点，提出了一种基于编辑距离的文档相似度度量方法. 首先分析了基于 tf-idf 的词袋模型和余弦相似度计算方法所存在的问题；其次利用 Jaccard 系数和编辑距离描述两个字符串的公共子串中词语之间的顺序差异，并提出了一种词序敏感相似度计算方法；最后利用实验数据对算法的有效性进行了验证，结果显示本文方法在 Top1、Top3 上的 F1 指标比原始的余弦相似度方法分别提高了 0.082 5、0.112 6，表明本文方法能够有效地提升信息检索系统的性能，具有很好的应用价值.

关键词：文本相似度；词袋模型；编辑距离；词序
中图分类号：TP391.1 **文献标识码**：A

A word order sensitive similarity measure based on edit distance

ZHANG Lei， CUI Rongyi*
（*College of Engineering*, *Yanbian University*, *Yanji* 133002, *China*）

Abstract: In this paper, a method is proposed to calculate the similarity between documents based on edit distance in order to improve the shortcoming that the cosine similarity method cannot reflect the order difference between the terms in the bag-of-words model. Firstly, the problems of the bag-of-words model based on tf-idf and the calculation method of cosine similarity are analyzed. Secondly, the order difference between the words in the common substrings of the two character strings is described by the Jaccard coefficient and the edit distance, and a word order sensitive similarity calculation method is proposed. Finally, the experimental data is used to verify the algorithm. The results show that the F1 value of this method on Top1 and Top3 is improved by 0.082 5 and 0.112 6 respectively compared with the original cosine similarity method. It shows that the method in this paper can effectively improve the performance of the information retrieval system and has good application value.

Keywords: text similarity; bag-of-words model; edit distance; word order

0 引言

随着信息技术的快速发展和文档数据的日益增加，如何从海量文档中找到期望的信息和有效地计算查询文本与文档的相似度逐渐成为人们关注的问题[1]. 目前，文本表示模型主要有词袋模型、主题模型和嵌入模型等[2]. 其中词袋模型是先将文档表示为词典大小维度的向量，然后通过计算两文档向量夹角的余弦值来度量二者的相似度，具有易于构建的优点[3]；但词袋模型忽略了特征词之间的词序、语法及语义等要素，即将目标对象仅仅看作是由若干个无序单词组成的集合，因

收稿日期：2020-04-14

* **通信作者**：崔荣一(1962—)，男，博士，教授，研究方向为自然语言处理、模式识别、智能计算.

此由该模型得到的词袋特征缺乏表达能力和区分度[4]. 为改善余弦相似度方法难以反映出用词袋模型表示的文本的不同词序所带来的语义变化，本文提出一种基于编辑距离的词序敏感相似度算法，并通过实验验证本文方法的有效性.

1　基于词袋模型的相似度计算方法

1.1　基于 tf-idf 的词袋模型

基于 tf-idf 的词袋模型是信息检索领域中一种常用的文本表示模型，其中 tf-idf 可以综合衡量一个词对所在文档和整个文档集的重要程度[5]. 在文档 d 中，词 t 的 tf-idf 值的计算方法为

$$tf\text{-}idf(t,d)=tf_{t,d}\cdot idf_t, \tag{1}$$

其中：

$$tf_{t,d}=\log_{10}(1+n_{t,d}), \tag{2}$$

$$idf_t=\log\frac{N}{df_t}. \tag{3}$$

上式中：$tf_{t,d}$ 是词 t 在文档 d 中的频率特征(term frequency，TF)，它反映的是一个词在所在文档中的重要程度；$n_{t,d}$ 为词 t 在文档 d 中出现的次数，词 t 出现的次数越多表示它的频率特征越大. 为了抑制某些高频词的权重，使出现频率不高的词语也能发挥作用，在式(2) 中对原始词的频数取对数，并加了平滑项. idf_t 为逆文档频率(inverse document frequency，IDF)；N 为文档集中的文档数量；df_t 为文档集频率(document frequency，DF). idf 的假设是：出现词 t 的文档数越少，则词 t 对文档集越重要，反之词 t 越不重要.

将一篇文档用词袋模型表示后，无论这些词语如何排列，每个词的权重都与原来相同，且文档的向量也不变. 语序与语义密切相关，如“狗/咬/人”和“人/咬/狗”，这里的词虽然相同，但顺序不一样其语义完全不同. 为此，本文提出一种基于编辑距离的词序敏感相似度度量方法，即在相似度计算中考虑语序因素.

1.2　文本相似度的计算方法

目前，常见的文本相似度计算方法有编辑距离、Jaccard 系数、N-gram 相似度、SimHash、余弦相似度、欧氏距离等[6]. 其中余弦相似度方法能够体现出参与计算的两个向量之间各维度的共性，且效果较好，因此该方法在文本相似度计算场景中被广泛采用. 余弦相似度的值仅与参与计算的两个向量的方向有关，而与向量的大小无关. 当文本表示为向量形式且采用 $tf\text{-}idf$ 作为权重时，余弦相似度值的范围为[0，1]. 将查询和文档分别用 $tf\text{-}idf$ 向量 $\boldsymbol{V}_q$ 和 $\boldsymbol{V}_d$ 表示后，则 $\boldsymbol{V}_q$ 和 $\boldsymbol{V}_d$ 的相似性可通过式(4) 进行度量.

$$\cos(\boldsymbol{V}_q,\boldsymbol{V}_d)=\frac{\boldsymbol{V}_q\cdot\boldsymbol{V}_d}{\|\boldsymbol{V}_q\|\cdot\|\boldsymbol{V}_d\|}. \tag{4}$$

依据公式(4)，计算查询与文档集中的所有文档的相似度后，按相似度值大小进行排序，并选取相似度最大的文本作为与查询文本最相似的文本. 采用 $tf\text{-}idf$ 作为词项权重时，通常采用 lnc.ltc 作为权重的计算机制[7]. lnc 指文档向量采用的是对数词频，不采用 idf 因子(同时基于效率和效果考虑) 及余弦归一化；ltc 指查询向量采用的是对数词频、idf 权重因子及余弦归一化.

用余弦相似度方法衡量用词袋模型表示的文本的相似度虽然具有很好的实用性，但该方法无法区分因不同词序产生的不同语义. 例如“太阳/从/东边/升起/，/从/西边/落下”和“太阳/从/西边/升起/，/从/东边/落下”，这两个句子的词语顺序稍有变动就会导致语义发生变化，但二者的余弦相似度值却为 1，即余弦相似度方法将两个句子视为完全相同，这与事实不符.

2　基于编辑距离的词序敏感相似度计算方法

下面以信息检索中的相似度计算为例对本文提出的方法进行说明. 假设查询 q 与文档 d 均已进行预处理，即已分词、去停用词、关键词归一，形成词语列表(分别用 W_q 和 W_d 表示). 将 W_q 和 W_d 表示为词袋模型，并用 $tf\text{-}idf$ 权重作为词语的特征. 构建文本向量 $\boldsymbol{V}_q$ 与 $\boldsymbol{V}_d$，两个向量间的夹角余弦值可以表示为 $\cos(\boldsymbol{V}_q,\boldsymbol{V}_d)$，即 q 与 d 的余弦相似度. 将向量归一化后，因余弦相似度能够表现出两个向量之间各维度的共性，因此余弦相似度能够度量文本中的共有词项程度. 为了体现词项之间的顺序对相似度的影响，继而能够辨别不同语义，本文通过引进能够体现符号串共性和差异性的度量因子来改造单一的余弦相似度的度量方法，其总体思路是：

1)用符号串之间的共性描述关注词序的必要性,因为符号之间的顺序重要性仅在相同符号的排列中存在,如果符号串之间没有公共部分则不存在符号顺序的问题.

2)用符号之间的差异性描述词序差异的程度,该差异表现在相同词项集合的不同排列之间的差异所带来的不同语义.差异性与共性相互作用造就了词序敏感性.

本文采用Jaccard系数(Jaccard coefficient)和编辑距离量化实现上述思想.

2.1 Jaccard系数

Jaccard系数是用于计算两个集合的公共部分占比大小的指标,因其计算简单而被广泛应用于学术论文查重、电子文档版权、文本聚类、文本分类、问卷调查整理、搜索引擎去重等涉及相似度计算的领域[8].Jaccard系数的计算公式为

$$J(A,B)=\frac{\sharp\left|A\cap B\right|}{\sharp\left|A\cup B\right|},\tag{5}$$

其中 $\sharp$ 表示集合中词语的数量,Jaccard系数的取值范围是$[0,1]$.式(5)表明,文档A和文档B的Jaccard系数等于二者交集大小与并集大小的比值.当A与B没有公共词语时,$J(A,B)=0$;当A与B的词语完全一样时,$J(A,B)=1$.例如:集合$A=\{a,b,c,d\}$,集合$B=\{c,d,e,f\}$,则$A\cap B=\{c,d\}$,$A\cup B=\{a,b,c,d,e,f\}$.因交集中有2个元素,并集中有6个元素,因此A与B的Jaccard系数为$J(A,B)=2/6=1/3$.

2.2 编辑距离

编辑距离[9]是衡量两个字符串差异程度的指标,其度量方式是一个字符串变为另一个字符串的最少操作数,因此其广泛应用于文本纠错、抄袭检测等.

莱文斯坦距离(Levenshtein distance)[10]是编辑距离的一种,其定义的操作包括插入、删除和替换3种.而本文因只需计算两序列的公共部分的编辑距离,因此只定义插入、删除两种操作即可满足需求.

本文根据动态规划思想将所定义的编辑距离算法描述如下:

Algorithm Edit Distance
Input: 两个中文句子A、B
Output: A与B的编辑距离
Step1 分别对A和B分词,形成词语列表L_A、L_B
Step2 计算L_A和L_B的长度:
$N=LENGTH(L_A)$; $M=LENGTH(L_B)$
Step3 创建编辑距离矩阵$E[N+1,M+1]$
Step4 初始化:
$E[0,0]=0$
for each row i from 1 to N do
$E[i,0]\leftarrow E[i-1,0]+$ del-cost($A[i]$)
for each column j from 1 to M do
$E[0,j]\leftarrow E[0,j-1]+$ ins-cost($B[j]$)
Step5 循环执行:
for each row i from 1 to N do
for each column j from 1 to M do
$E[i,j]\leftarrow$ MIN($E[i-1,j]+$ del-cost($A[i]$), $E[i,j-1]+$ ins-cost($B[j]$))
Step6 返回$E[N,M]$

例如:A="海南/比/黑龙江/温度/更高",B="黑龙江/比/海南/温度/更高",则二者的编辑距离为4.

当编辑距离的操作种类为插入、删除两种,且每种操作代价为1时,编辑距离的取值范围为$[0,\max\{2(M-1),2(N-1)\}]$,且取整数.当两个句子的顺序完全一致时编辑距离为0,当两个句子的单词互为逆序时编辑距离取最大值.编辑距离的时间复杂度与空间复杂度均为$O(MN)$.

2.3 改进的相似度计算方法

假设q_c和d_c分别是保留文本中原词序的公共子序列集合;$E(q_c,d_c)$表示q_c和d_c的编辑距离,用以衡量具有公共词语的两序列中词语顺序的差异性;$J(W_q,W_d)$表示q与d中词语集合的Jaccard系数,用以度量编辑距离的重要性,其假设是查询与文档公共词语数量越多,语序的影响就越大.因此$J(W_q,W_d)\cdot E(q_c,d_c)$可以综合衡量两序列的差异性.

$J\cdot E$因素按乘积方式(乘积记为x)对余弦相似度产生修正作用.这种修正为衰减型,即x越大对余弦相似度的衰减作用越大.当两个符号串没有公共部分或者完全相同时,$x=0$,即其相似度还原为余弦相似度.衰减函数有多种,本文在此仅考虑3个具有代表性的函数,如式(6)—(8)所示.图1为这3个函数的衰减趋势.由图1可以看

出：函数(6)和(7)虽然简单，但它们的衰减趋势过于强烈，难以保持余弦相似度的固有优点；而对数函数可以有效缓解衰减趋势，如式(8)所示. 因此本文按对数方式修整 $J \cdot E$ 因子，以保持余弦相似度度量方法的固有合理性.

$$y_1 = \mathrm{e}^{-x}, \tag{6}$$

$$y_2 = \frac{1}{1+x}, \tag{7}$$

$$y_3 = \frac{1}{1+\log_{10}(1+x)}. \tag{8}$$

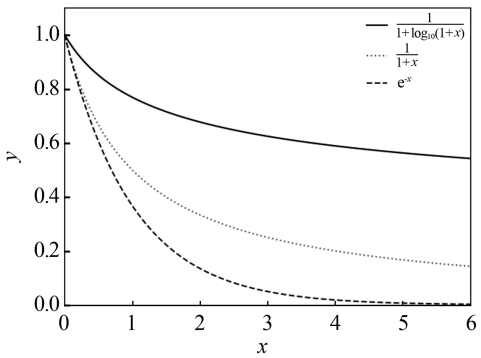

图 1　3 种衰减函数的对比

因此，查询 q 与文档 d 的词序敏感相似度可以表示为

$$\mathrm{sim}(q,d) = K_{q,d} \cdot \cos(\boldsymbol{V}_q, \boldsymbol{V}_d). \tag{9}$$

其中 $K_{q,d}$ 为余弦相似度的抑制因子，其表达式为

$$K_{q,d} = \frac{1}{1+\log_{10}[1+J(W_q,W_d)\cdot E(q_c,d_c)]}. \tag{10}$$

在式(9)、(10)中，$\cos(\boldsymbol{V}_q,\boldsymbol{V}_d) \in [0,1]$，$J(W_q, W_d) \in [0,1]$，$E(q_c,d_c) \in [0, \max\{2(\mathrm{len}(W_q)-1), 2(\mathrm{len}(W_d)-1)\}]$，$K_{q,d} \in (0,1]$，因此词序敏感相似度的取值范围为 $\mathrm{sim}(q,d) \in [0,1]$. 由式(9)、(10)可知，本文方法在计算两序列的相似度时考虑了序列中词语的顺序因素. 词序敏感算法的核心机制是抑制因子 $K_{q,d}$，$K_{q,d}$ 值越小对原余弦相似度值的抑制能力越大. 式(9)存在以下几种情况：

1) 当两个序列的公共子序列的顺序存在不一致时，则根据不一致程度对相似度值进行一定的削弱.

2) 当两个序列的语序越趋向于一致时，二者的相似度值越接近于余弦相似度值.

3) 当两个序列的所有公共子序列的顺序完全一致时，此时式(10)的值为 1，式(9)退化为余弦相似度.

4) 当两个序列完全不同时，$\cos(\boldsymbol{V}_q,\boldsymbol{V}_d)=0$，此时 $\mathrm{sim}(q,d)=0$.

3　实验结果与分析

3.1　实验数据的构建

本文实验使用的语料是人工撰写的 307 组 C 语言相关问题，每组为相似问题的 6 个不同的表述方式，并将其中的 1 个表述设为主问题，其他 5 种表述设为副问题. 副问题与主问题的语义有些是相同的，有些是不同的. 实验语料如表 1 所示.

表 1　相似度实验语料

ID	主问题	副问题 1	副问题 2	…	副问题 5	同义标签
1	数组名是指针吗？	指针是数组名吗？	数组名和指针一样吗？	…	数组名和指针的区别是什么？	[1,0,0,1,1]
2	可以在头文件中说明或定义变量吗？	在头文件中可以说明或定义变量吗？	说明或定义变量可以在头文件中吗？	…	头文件的作用是什么？	[1,1,0,1,0]
3	使用＝＝来判定两个结构是否相等为什么是不合法的？	如何判断两个结构是否相等？	＝＝可以判断两个结构是否相等吗？	…	判定两个结构是否相等的合法方式是什么？	[0,1,0,1,0]
…	…	…	…	…	…	…
306	数组可以是左值吗？	左值可以是数组吗？	＝左面可以是数组吗？	…	什么是左值？	[0,1,1,0,0]
307	浮点常量为什么存储成 double 格式而不是 float 格式？	浮点常量为什么存储成 double 格式？	浮点常量为什么不存储成 float 格式？	…	浮点常量存储成 double 格式还是 float 格式？	[0,0,1,0,1]

表1中“同义标签”中的数组元素,分别表示每个副问题与主问题的语义是否相同.其中元素值为1表示语义相同,0表示语义不同.如[1,0,0,1,1]表示第1、4、5个副问题与主问题是同义句,第2、3个副问题与主问题存在语义区别.

3.2 实验结果与分析

分别使用原始余弦相似度方法和基于编辑距离的词序敏感相似度计算方法对上述C语言问题集进行文本检索实验,结果如表2所示.图2为表2的可视化图.在表2和图2中,P、R、$F1$分别代表准确率、召回率、$F1$值,@K表示在Top$K$上的指标.

以表1中的第1组为例进行说明.q=“数组名/是/指针/吗?”,d_1=“指针/是/数组名/吗?”,d_2=“数组名/和/指针/一样/吗?”.由式(10)可以算出,$K_{q,d_1}=0.5886$,$K_{q,d_2}=0.7686$.由此可见,词序敏感算法的确能够抑制因语序的不同而产生的语义差异.

从表2和图2可以看出,在Top1、Top3上,词序敏感算法的F1值比余弦相似度算法的F1值分别提高了0.0825、0.1126.由此表明,词序敏感算法的准确率和召回率的综合值显著优于余弦相似度算法的准确率和召回率的综合值.

表2 不同相似度计算方法的实验结果

方法	$P@1$	$R@1$	$F1@1$	$P@3$	$R@3$	$F1@3$	$P@5$	$R@5$	$F1@5$
余弦相似度算法	0.6612	0.3079	0.4201	0.7068	0.6816	0.6940	0.6124	0.9074	0.7313
词序敏感算法	0.8795	0.3518	0.5026	0.7720	0.7724	0.7722	0.6078	0.9385	0.7378

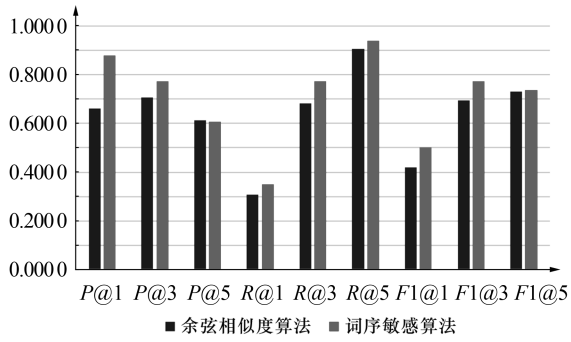

图2 不同算法的性能对比

4 结论

研究表明,本文所提出的基于编辑距离的词序敏感相似度算法能够反映出句子之间的语序差异;因此,利用本文提出的方法在计算词袋模型表示的文本相似度时,其性能优于余弦相似度方法.本文在研究中没有分析何种语序会影响语义,为此我们今后将建立词序敏感词库,并标注查询和文档的词性,研究不同词序变化而导致的语义变化,以进一步提升本文方法的性能.

参考文献:

[1] 刘娇,崔荣一,赵亚慧,等.跨语言文献相似度的分析方法[J].延边大学学报(自然科学版),2016,42(2):151-155.

[2] 骆梅柳.文本表示模型在文本挖掘中的应用[J].现代信息科技,2019,3(7):24-25.

[3] 赵雪,崔荣一.基于N层向量空间模型的文本相似度计算方法[J].延边大学学报(自然科学版),2016,42(3):231-234.

[4] 陈行健,胡雪娇,薛卫.基于关系拓展的改进词袋模型研究[J].小型微型计算机系统,2019,40(5):1040-1044.

[5] MA H, ZHOU R, LIU F, et al. Effectively classifying short texts via improved lexical category and semantic features[C]//Proc of International Conference on Intelligent Computing, 2016, Part I: 163-174.

[6] 马思丹.基于加权Word2vec的微博文本相似度计算方法研究[J].西安电子科技大学,2019:15-16.

[7] MANNING C D, RAGHAVAN P, SCHÜTZE H. Introduction to Information Retrieval [M]. Cambridge: Cambridge University Press, 2008.

[8] 刘宝超,崔荣一.基于最大Jaccard相似度的互激励实体验证算法[J].延边大学学报(自然科学版),2015,41(1):42-45.

[9] 陈鑫,李伟康,洪宇,等.面向问句复述识别的多卷积自交互匹配方法研究[J].中文信息学报,2019,33(10):99-108.

[10] LEVENSHTEIN V. Binary codes capable of correcting spurious insertions and deletions of ones [J]. Problems of Information Transmission, 1965,1(1):8-17.