计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2017,26(12):137−142 [doi: 10.15888/j.cnki.csa.006123]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于改进的 Jaccard 系数文档相似度计算方法①

俞婷婷, 徐彭娜, 江育娥, 林　劼

(福建师范大学 软件学院, 福州 350108)
通讯作者: 林　劼, E-mail: linjie891@163.com

摘　要: 文本相似度主要应用于学术论文查重检测、搜索引擎去重等领域, 而传统的文本相似度计算方法中的特征项提取与分词环节过于冗杂, 而且元素的随机挑选也会产生权重的不确定性. 为了解决传统方法的不足, 提出一种基于改进的 Jaccard 系数确定文档相似度的方法, 该算法综合考虑了各元素、样本在文档中的权重及其对多个文档相似度的贡献程度. 实验结果表明, 基于改进的 Jaccard 系数的文档相似度算法具有实效性并且能够得到较高的准确率, 适用于各种长度的中英文文档, 有效地解决现有技术中存在的文档间相似度计算不精的问题.

关键词: 文本相似度; Jaccard 系数; 文本分析; 文本查重; 文本检索

引用格式: 俞婷婷,徐彭娜,江育娥,林劼.基于改进的 Jaccard 系数文档相似度计算方法.计算机系统应用,2017,26(12):137–142. http://www.c-s-a.org.cn/1003-3254/6123.html

Text Similarity Method Based on the Improved Jaccard Coefficient

YU Ting-Ting, XU Peng-Na, JIANG Yu-E, LIN Jie

(Faculty of Software, Fujian Normal University, Fuzhou 350108, China)

Abstract: Text similarity check is mainly used in Re-check detection of Papers, the deduplication of search engines and other fields. However, it's extremely fussy to extract feature items with the traditional methods for computing the text similarity. In addition, it will bring uncertainty to select elements randomly. To solve these problems, a text similarity method based on improved Jaccard coefficient is proposed. This method takes into account the weights of elements and samples in the document, even the contribution degree to multiple text similarity. The results suggest that the text similarity method based on the improved Jaccard coefficient has been proved to be effective with a satisfactory accuracy, which can be applicable to various lengths of Chinese, English documents. It effectively solves the problem of inexact computing with existing technologies.

Key words: text similarity; Jaccard coefficient; text analysis; text checking; text retrieval

随着现代计算机技术的快速发展与网络的飞速普及, 网上数据资源也在急速增加, 丰富的数据资源为人们的生活提供了便利, 也提高了人们的工作效率. 在这些数据资源给人们提供便利的同时, 也出现了不少问题, 如学术论文抄袭、新闻转载等. 在这样的背景下, 文档查重检测应运而生. 相似度计算具有广泛的应用前景, 目前主要应用于学术论文查重检测、电子档版权、文本聚类、文本分类、问卷调查整理、搜索引擎去重等.

相似性数据的检测数据量十分庞大. 在百度百科上, 以中国学位论文全文数据库收录的学位论文为例[1], 截止 2011 年 10 月, 论文总量达 200 万篇以上, 每年以 30 万篇以上的速度在增长. 再如, 2016 年 4 月份中国 50 所高校在线发表论文数量高达 62 000 篇以上[2], 其

① 基金项目: 国家自然科学基金 (61472082); 福建省自然科学基金 (2014J01220)
收稿时间: 2017-03-21; 修改时间: 2017-04-13; 采用时间: 2017-04-17

中大部分的科研论文都需要进行相似性检测. 如此庞大的数据, 借助一种基于改进的 Jaccard 系数确定多个文档相似度的方法进行检测, 实现多个文档之间的相似性比对是很有必要的. 一个好的计算文档相似度的方法在学术论文相似性检测、文本聚类、文本分类、舆情调查等领域中具有重要意义.

本文第 1 节介绍文本相似度计算方法的现状; 第 2 节介绍本文提出的相似度计算方法; 第 3 节通过实验验证并分析本文方法的有效性; 第 4 节对实验结果进行总结并给出下一步研究工作.

1 相关研究工作

文本相似度是文本挖掘的一个重要内容, 近年来不少研究人员对文本挖掘的研究也集中在文本相似度的计算上.

传统的文本相似度计算方法一般采用向量空间模型[3], 实际上就是将语义相似度用空间上的相似度来表达, 对文本进行特征项选取后再对其做加权处理, 用向量来表示特征项权重, 使这些特征项权重从离散的数字转化为一个个带向量的分量, 于是文本的相似度计算就转化成特征项权重在高维空间内的相似度计算[4]. 这种计算方法直观易懂, 有效地将文本处理的问题转化为数学问题. 但是在对特征项进行加权时向量空间模型没有考虑到特征项在文本中的位置信息, 并且忽略了各个特征项的语义在文本之间的关联性. 许鑫[5]在实验中对传统的向量空间模型做了优化, 提出了四种实验方案, 认为不仅要考虑到主题间内容上的语义相似度, 还要兼顾到语义相似度低的链接中可能存在的相关关系. 基于编辑距离的基础, G Sidorov[6]提出使用一种树编辑距离的算法来计算文本相似度, 实验结果的准确率高于编辑距离. 贾惠娟[7]在有特征词知识库支持的前提条件下, 提出将编辑距离与向量空间模型相结合构建一种新的文本相似度计算模型, 虽然在数据预处理的过程中可能会丢失一些文本特征项, 但是用于领域文档查询也取得不错的效果.

为了解决传统方法的不足, 王小林[8]考虑到特征项在文本中的位置对权重的影响, 对特征项添加了位置权重, 进行信息增益和熵值计算, 虽然该算法在一定程度上提高了查全率和查准率, 但该算法的时间复杂度较高, 还需进一步改进才能运用在实际环境中. 考虑到不同文本中特征项的频率波动也会影响到特征项权值, 周丽杰[9]将得到的特征项权值经过马尔科夫模型与向量空间模型的结合, 得到一个总体相似度, 提高了准确率, 忽略了关键词在不同文档中的权重问题. 与传统的文本相似度算法不同, 何维[10]从文本分析的粒度出发, 将文本相似度用句子相似度来表示, 结合 KNN 算法, 得到的准确率和回归率较为可观.

为了得到更加精确的文本相似度结果, 除了上述计算方法外, 还有其他比较具有代表性的算法, Yang Liu[11]结合五种不同特征使用支持向量机对句子间语义相似度的得分进行预测, 实验结果表明该方案具有良好的泛化能力. Jiyi Li[12]通过给定的一个文档集合充分利用不同的模型来评估文档的语义相似度, 观察模型与线性或非线性因素的融合关系来选出一个最合适的模型表示. 李圣文[13]提出的一种基于熵的文本相似性算法, 该方法基于字符的角度, 考虑到文本间共同的字符串对相似度的影响, 在提取文本间的字符信息后, 对共同的字符串进行了维度上的度量, 再用熵的方法进行相似度计算. 这种相似度方法在一定程度上避免了对长文本的特征提取, 而是直接进行相似度计算, 并且考虑到了字符串在不同长度文本中所占的比重. 更有一种基于图形编辑距离的算法被 Schuhm- acher[14]提出, 用于计算两个文本的语义距离, 为文本相似度的计算提供了新思路.

N Kowsalya[15]提出的 K-nearest 模型有效解决了在文档分类中会遇到的特征的高维性、数据的高容量等问题. 涂建军[16]在特征提取算法中, 通过对嵌套词串的处理, 有效地避免了在降维过程中存在丢失重要信息的问题. 在对短文本的处理上, X Yan[17]构建的 biterm 主题模型可以发现较为突出和连贯的主题, 该实验结果优于 R Řehůřek[18]用传统的 LDA 算法得到的结果. Zhifei Zhang[19]先是利用 LDA 对主题建模后对短文本中同义词与多义词的权重进行调整, 得到较好的实验结果. 王贤明[20]提出一种基于 n-Gram 的相似度算法操作简单, 避免了传统文本相似度计算方法中提取特征项这一繁杂的环节, 有效地提高了计算效率, 但在计算权重的评价函数过程中, 采用随机挑选元素的方法, 造成了元素权重的不确定性.

也有学者利用 Jaccard 相似度来实现文本相似度的计算, 孙宇[21]利用 Jaccard 相似度实现了社团发现, 并完成了聚类研究. Jaccard 系数是两元素交集与并集的个数之比, 不考虑个体间具体差异值的大小, 仅关注

个体间是否存在共同的特征. 基于 Jaccard 系数可以成对评估数据的相似性和多样性, Niwattanakul[22]将其用于比较关键词之间的相似性, 并在关键词聚类上取得了较好的结果. Huang[23]对 7 个数据集采用 5 种最为广泛使用的相似性措施, 并表明 Jaccard 系数能达到最好的效果. 与现有的相似度计算方法不同, 邓琨[24]提出的 JS 和 JSJ 相似度计算方法, 不仅可以反映出数据的局部相似性, 还可以高效地从整体上来评估其相似关系.

针对传统方法的不足, 本文运用一种基于改进的 Jaccard 模型的计算方法, 提出一种兼顾特征项权重与计算效率的文本相似度计算方法, 用以获得更准确的文本信息描述, 提高文本分类性能.

2 方法及原理

文本相似度是指在两篇或者多篇文档中出现的词语、句子、段落或者篇章的吻合程度. 两篇文档在词语、句子、段落或者篇章上越相同或相似部分越多, 代表着这两篇文档的相似度越高. 文档相同是特殊的相似, 即相似度为 100%.

2.1 主要步骤

以下介绍本文方法的主要步骤:

(1) 给定参数 K, K 为文档中移动窗口大小. 给定两个文档长度分别为 n_1、n_2 的文档 X 和文档 Y. 确定文档中长度为 K 的元素个数, 并计算每个元素在文档中所占的比重;

(2) 计算每个元素的 Jaccard 相似度;

(3) 计算每个元素在所有长度为 K 的元素中所占的比重;

(4) 确定每个 K 字元素的权重;

(5) 汇总所有 K 字元素相似度, 计算文档相似度.

2.2 步骤的详细描述

以下是上述步骤的详细解析.

(1) 确定文档中长度为 K 的元素个数, 并计算每个元素在文档中所占的比重.

假设文档 X 和 Y 的文档长度分别为 n_1 和 n_2, 则文档 X 中含有 n_1 个长度为 1 的元素X_{w_1}, 含有 (n_1-1) 个长度为 2 的元素X_{w_2}, 依此类推, 文档 X 中含有 1 个长度为 n_1 的元素, 这些元素的滑动窗口的大小为 1, 该滑动窗口从文本起始位置滑向终止位置进而形成了 n-Gram. 所以在文档 X 中含有 (n_1-K+1) 个长度为 K 的 K 字元素 ($1\le K\le n_1$), 文档 Y 中含有 (n_2-K+1) 个长度为 K 的 K 字元素 ($1\le K\le n_2$).

而每个 K 字元素在文档 X、Y 中所占的权重分别为:

$$NX_w = \frac{|X_w|}{n_1 - K + 1} \tag{1}$$

$$NY_w = \frac{|Y_w|}{n_2 - K + 1} \tag{2}$$

(2) 计算元素的 Jaccard 相似度.

根据 Jaccard 相似度原理, 文档 X 和文档 Y 的 Jaccard 相似度等于文档 X 和文档 Y 的交集大小与并集大小的比值. 若有元素w_i同时存在于文档 X、Y 中, 那么该元素对应的两文档改进的 Jaccard 相似度为:

$$\begin{aligned} C_J(X_{w_i}, Y_{w_i}) &= \frac{|X_{w_i} \cap Y_{w_i}|}{|X_{w_i} \cup Y_{w_i}|} \\ &= \frac{\min(|X_{w_i}|, |Y_{w_i}|)}{\max(|X_{w_i}|, |Y_{w_i}|)} = \frac{\min(NX_{w_i}, NY_{w_i})}{\max(NX_{w_i}, NY_{w_i})} \end{aligned} \tag{3}$$

(3) 计算每个元素在所有长度为 K 的元素中所占的比重.

用$\varepsilon(w_i)$代表元素w_i在文档 X 和文档 Y 所有 n-Gram 长度为 K 的元素中的所占的权重$\varepsilon(w_i)$, 则有:

$$\varepsilon(w_i) = \frac{|X_{w_i}| + |Y_{w_i}|}{n_1 - K + 1 + n_2 - K + 1} \tag{4}$$

(4) 确定元素对文档相似度是否有贡献.

用$F(w_i)$代表元素 w_i 在文档 X 和文档 Y 是否同时出现, 则:

$$F(w_i) = \begin{cases} 1 & (w_i \in X \cap Y) \\ 0 & (w_i \notin X \cap Y) \end{cases} \tag{5}$$

若元素同时出现在两个文档中, 则该元素对文档 X 和 Y 的文档相似度有贡献, $F(w_i)$的值为 1; 否则, $F(w_i)$的值为 0.

(5) 计算文档相似度.

文档 X 和文档 Y 的相似度评价函数如下:

$$\begin{aligned} &Similarity_{c_J}(X, Y) \\ &= \frac{\sum_{w_i \in \sum^K} C_J(X_{w_i}, Y_{w_i})\varepsilon(w_i)}{\sum_{w_i \in \sum^K} F(w_i)\varepsilon(w_i)} \end{aligned} \tag{6}$$

2.3 示例

以下以具体文档为实例来介绍本文的文本相似度方法.

例如在文档 X=“abcabc123”与文档 Y=“123abc”中, 他们的文档长度分别为 n_1=9 和 n_2=6.

假设 n-Gram 长度 K=3, 那么在文档 X 中含有 7 个 n-Gram 长度为 3 的 L 字元素: {abc, bca, cab, abc,

bc1, c12, 123}, 在文档 Y 中含有 4 个 n-Gram 长度为 3 的 L 字元素: {123, 23a, 3ab, abc}.

在文档 X 中 n-Gram 长度为 3 的元素为{abc, bca, cab, bc1, c12, 123}, 对应的数量分别为{2, 1, 1, 1, 1, 1}; 在文档 Y 中 n-Gram 长度为 3 的元素为{123, 23a, 3ab, abc}, 对应的数量分别为{1, 1, 1, 1}.

在文档 X 中, 元素 w_{abc}, w_{bca}, w_{cab}, w_{bc1}, w_{c12}, w_{123} 所占的权重分别是$\frac{2}{7},\frac{1}{7},\frac{1}{7},\frac{1}{7},\frac{1}{7},\frac{1}{7}$; 在文档 Y 中, 元素 w_{123}, w_{23a}, w_{3ab}, w_{abc} 所占的权重分别是$\frac{1}{4},\frac{1}{4},\frac{1}{4},\frac{1}{4}$.

由于元素 w_{abc} 和 w_{123} 同时出现在文档 X 和文档 Y 中, 所以 X、Y 两文档改进的 Jaccard 相似度为:

$$C_J(X_{w_{abc}},Y_{w_{abc}})=\frac{|X_{w_{abc}}\cap Y_{w_{abc}}|}{|X_{w_{abc}}\cup Y_{w_{abc}}|}=\frac{\min(|X_{w_{abc}}|,|Y_{w_{abc}}|)}{\max(|X_{w_{abc}}|,|Y_{w_{abc}}|)}=\frac{\min(NX_{w_{abc}},NY_{w_{abc}})}{\max(NX_{w_{abc}},NY_{w_{abc}})}=\frac{1/4}{2/7}=\frac{7}{8}$$

$$C_J(X_{w_{123}},Y_{w_{123}})=\frac{|X_{w_{123}}\cap Y_{w_{123}}|}{|X_{w_{123}}\cup Y_{w_{123}}|}=\frac{\min(|X_{w_{123}}|,|Y_{w_{123}}|)}{\max(|X_{w_{123}}|,|Y_{w_{123}}|)}=\frac{\min(NX_{w_{123}},NY_{w_{123}})}{\max(NX_{w_{123}},NY_{w_{123}})}=\frac{1/7}{1/4}=\frac{4}{7}$$

$$C_J(X_{w_{bca}},Y_{w_{bca}})=C_J(X_{w_{cab}},Y_{w_{cab}})=C_J(X_{w_{bc1}},Y_{w_{bc1}})=C_J(X_{w_{c12}},Y_{w_{c12}})=C_J(X_{w_{23a}},Y_{w_{23a}})=C_J(X_{w_{3ab}},Y_{w_{3ab}})=0$$

文档 X、Y 中所有元素为: w_{abc}, w_{bca}, w_{cab}, w_{bc1}, w_{c12}, w_{123}, w_{23a}, w_{3ab}, 那么这些元素在文档 X 和文档 Y 所有 n-Gram 长度为 3 的元素中的所占的比重是:

$$\varepsilon(w_{abc})=\frac{3}{11},\varepsilon(w_{bca})=\frac{1}{11},\varepsilon(w_{cab})=\frac{1}{11}$$

$$\varepsilon(w_{bc1})=\frac{1}{11},\varepsilon(w_{c12})=\frac{1}{11},\varepsilon(w_{123})=\frac{2}{11}$$

$$\varepsilon(w_{23a})=\frac{1}{11},\varepsilon(w_{3ab})=\frac{1}{11}$$

由于元素 w_{abc} 和 w_{123} 同时出现在文档 X 和文档 Y 中, 所以:

$$F(w_{abc})=1,F(w_{123})=1$$

而元素 w_{bca}, w_{cab}, w_{bc1}, w_{c12}, w_{23a}, w_{3ab} 不是同时出现在文档 X 和文档 Y 中, 所以:

$$F(w_{bca})=F(w_{cab})=F(w_{bc1})=F(w_{c12})=F(w_{23a})=F(w_{3ab})=0$$

所以, 文档 X 和文档 Y 的相似度为:

$$Similarity_{c_J}(X,Y)=\frac{\sum_{w_h\in\sum^K}C_J(X_{w_h},Y_{w_h})\varepsilon(w_h)}{\sum_{w_h\in\sum^K}F(w_h)\varepsilon(w_h)}=\frac{C_J(X_{w_{abc}},Y_{w_{abc}})\varepsilon(w_{abc})+C_J(X_{w_{123}},Y_{w_{123}})\varepsilon(w_{123})}{F(w_{abc})\varepsilon(w_{abc})+F(w_{123})\varepsilon(w_{123})}\approx 0.75$$

3 实验设计及结果分析

3.1 实验目的

本实验的目的是验证上述技术方案的有效性与准确度, 且探讨该技术方案下, 元素的 L 字长度与相似度的关系.

3.2 实验方案

本文的实验数据来源于搜狗实验室提供的文本分类语料库[25], 一共有 43 565 篇文本, 从其中随机选出 8 000 篇长短不一的文档. 为了减小实验误差, 对 8 000 篇文本进行字符处理, 去掉文本中的空格与标点符号及一些特殊符号, 如“●”, 即不将上述符号计入文本长度中. 经过上述处理后, 再筛选出文本长度超过 10k 的 100 篇文档. 在接下来的实验中, 用与这些文档的内容不相关的字符来对这 100 篇文档进行字符替换, 每篇文档每次替换的字符数占其总字符数的 5%、10%、…、95% 这 19 个比例, 即替换字符后的文档与原文档的字符重复比例为 95%、90%、…、5%. 替换的位置是从每篇文本中任意筛选出来的, 即经过字符替换后得到的文档与原文档的字符相异之处是随机的. 本实验的文档内容涉及领域较为广泛, 如汽车、财经、健康、旅游等.

本实验着重从两个方面来验证上述提出的计算文本相似度的有效性: (1) 验证本方法计算得到的相似度与实际重复率之间的关系. (2) 分析 L 参数的选择与计算精度的关系. 其中实验一是通过改变 L 字计算不同字符重复比例下的文本相似度值; 实验二则是利用实验一得出的重复比例与相似度的相关性 (精度), 计算在不同 L 字长度的元素下相关性 (精度) 发生的变化.

实验一. 验证本方法计算得到的相似度与实际重复率之间的关系.

经过数据预处理, 对得到的文本长度超过 10 k 的 100 个文档, 分别对其进行如下操作 (为简化问题, 本实验主要针对一篇文档对不同 L 字时字符重复比例与相似度的关系进行探讨与说明, 而其余文档的处理方式可以对照参考).

对于文档 A, 分别按照不同重复比例对其进行字

符替换, 依次得到 19 个文档: A_1, A_2, …, A_{19}. 即文档 A_1, A_2, …, A_{19} 与文档 A 的字符重复比例分别为 5%, 10%, …, 95%. 分别选择不同长度的 L 字 (L=2, 3, …, 7) 元素, 计算在该元素长度下, 文档 A_1, A_2, …, A_{19} 与文档 A 的文本相似度. 并计算 100 篇文档 19 种重复比例的平均相似度.

实验二. 分析 L 参数的选择与计算精度的关系.

分别选择不同长度的 L 字 (L=2, 3, …, 7), 计算实验得到的字符重复比例与文本相似度的关系, 即相关性 (精度), 分析在不同 L 字长度的元素下相关性 (精度) 发生的变化.

3.3 实验结果与分析

(1) 不同 L 字时重复比例与相似度的关系

对实验数据的 100 个文档与其在不同 L 字时重复比例与平均文本相似度的关系进行计算, 得到的结果如图 1 所示, 横坐标表示重复比例, 纵坐标表示该重复比例下对应的平均文本相似度.

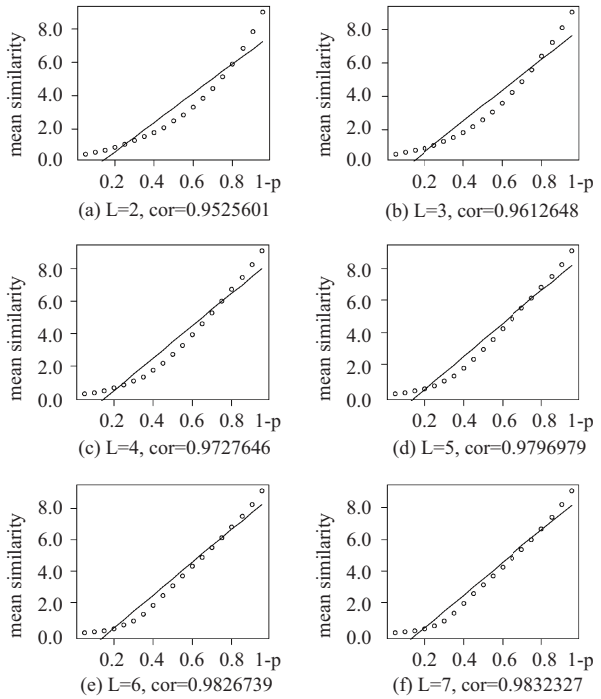

图 1 不同 L 字时重复比例与相似度的关系

对图 1 进行以下分析:

1) 观察图中 6 种 L 字长度下重复比例与平均相似度的关系可以发现, 在所有 L 字的实验中, 相似度总是随着重复率的增加而增加, 并且重复比例与平均相似度基本成线性关系. 随着 L 字的增加, 重复比例与相似度的线性回归关系越明显.

2) 相似度与重复比例不一定总呈现相等关系. 在实际应用过程中, 可对训练文本进行训练得到相似度与重复率的对应关系, 进而对测试文本进行计算, 可根据计算所得的相似度结果推断出实际重复率的值.

3) 从整体趋势来看, 图中 6 种 L 字长度所对应的曲线没有太大的差别. 这说明影响两文本相似度的因素不只是所选 L 字长度. 并且随着长度 L 字的不断增加, 相似度曲线趋于平稳, 如 L 字为 6、7 的散点图所对应的曲线明显比 2 对应的曲线起伏更小, 甚至接近一条直线. 这主要是因为, 随着 L 字不断增加, 相似度计算越精确, 从而使最终结果趋于平稳.

(2) 计算相关性 (精度) 与 L 字长度的关系

如图 2 所示, 在 L=2 至 7 时, 随着 L 字长度的不断增大, 字符重复比例与相似度的相关性也呈现出增长的趋势, 即 L 字长度与相关性呈线性关系, L 字的值取的越大, 相似度计算得越精确, 相关性递增得比较明显. 但是 L 字大于 7 以后, 相关性的递增幅度为负数, L 越大, 相关性越低. 因此, 在实际应用过程中, 推荐 L 字的取值在 7 左右.

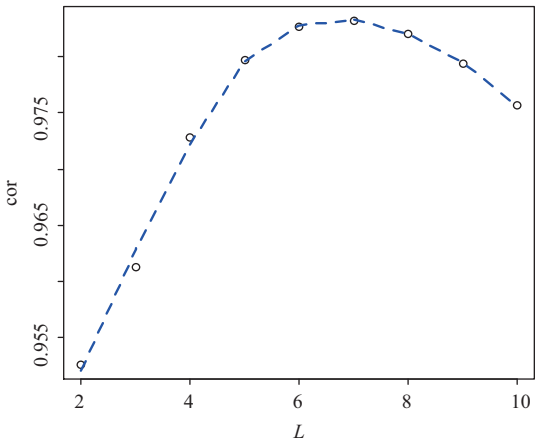

图 2 计算相关性 (精度) 与 L 字长度的关系

4 结语

本文提出的一种基于改进的 Jaccard 系数确定文档相似度的方法, 综合考虑了各元素、样本在文档中的权重及其对多个文档相似度的贡献程度, 可以有效地解决现有技术中存在的文档间相似度计算不精的问题. 另外, 将本文提出的方法运用到多文档相似度的确定, 可以有效地避免元素因随机挑选所带来的权重的不确定性, 还可以避免传统文本相似度计算方法中不

可避免的特征项提取与分词环节中出现的高维问题. 此外, 本文的方法适用于各种长度的中、英文文档. 实验结果表明, 一种基于改进的 Jaccard 系数确定文档相似度的方法在一定程度上可以提高文档间相似度计算的精度. 该方法计算简单, 速度快, 精度高, 可以应用在文本聚类、文本分类等文本挖掘技术中.

参考文献

1 百度百科. 中国学位论文全文数据库. http://baike.baidu.com/view/7134347.htm, 2011.

2 中国科技论文在线. 2016 年 4 月份各高校在线发表论文数量统计排序. 2016. http://www.edu.cn/rd/gao_xiao_cheng_guo/shu_ju_pai_hang/201605/t20160503_1393357.shtml. [2016-10-11]

3 Sidorov G, Velasquez F, Stamatatos E, *et al*. Syntactic N-grams as machine learning features for natural language processing. Expert Systems with Applications, 2014, 41(3): 853–860. [doi: 10.1016/j.eswa.2013.08.015]

4 薛苏琴, 牛永洁. 基于向量空间模型的中文文本相似度的研究. 电子设计工程, 2016, 24(10): 28–31. [doi: 10.3969/j.issn.1674-6236.2016.10.008]

5 许鑫, 苏晓兰. 基于文本计算和链接分析的主题导航优化-以 ERS 网站为例. 情报学报, 2015, 34(9): 938–948.

6 Sidorov G, Gómez-Adorno H, Markov I, *et al*. Computing text similarity using tree edit distance. 2015 Annual Conference of the North American Fuzzy Information Processing Society (NAFIPS) Held Jointly with 2015 5th World Conference on Soft Computing (WConSC). Redmond, WA, USA. 2015. 1–4.

7 贾惠娟. 一种改进的文本相似度算法在政务系统中的应用. 信息技术与信息化, 2016, (7): 49–52.

8 王小林, 肖慧, 邰伟鹏. 基于 Hadoop 平台的文本相似度检测系统的研究. 计算机技术与发展, 2015, 25(8): 90–93.

9 周丽杰, 于伟海, 郭成. 基于改进的 TF-IDF 方法的文本相似度算法研究. 泰山学院学报, 2015, 37(3): 18–22.

10 何维. 基于多示例学习的中文文本表示及分类研究. 大连理工大学[硕士学位论文]. 大连: 大连理工大学, 2009.

11 Liu Y, Sun CJ, Lin L, *et al*. Computing semantic text similarity using rich features. 29th Pacific Asia Conference on Language, Information and Computation. Shanghai, China. 2015. 44–52.

12 Li JY, Shimizu T, Yoshikawa M. Document similarity computation by combining multiple representation models. 2015 16th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD). Takamatsu, Japan. 2015. 1–6.

13 李圣文, 凌微, 龚君芳, 等. 一种基于熵的文本相似性计算方法. 计算机应用研究, 2016, 33(3): 665–668.

14 Schuhmacher M, Ponzetto SP. Knowledge-based graph document modeling. Proc. of the 7th ACM International Conference on Web Search and Data Mining. New York, USA. 2014. 543–552.

15 Lin YS, Jiang JY, Lee SJ. A similarity measure for text classification and clustering. IEEE Trans. on Knowledge and Data Engineering, 2014, 26(7): 1575–1590. [doi: 10.1109/TKDE.2013.19]

16 涂建军, 何汉林. 基于语义分析的降维特征提取. 情报学报, 2014, 33(9): 952–958.

17 Yan XH, Guo JF, Lan YY, *et al*. A biterm topic model for short texts. Proc. of the 22nd International Conference on World Wide Web. Rio de Janeiro, Brazil. 2013. 1445–1456.

18 Řehůřek R, Sojka P. Software framework for topic modelling with large corpora. Proc. of LREC 2010 Workshop New Challenges for NLP Frameworks. Valletta, Malta. 2010. 45–50.

19 Zhang ZF, Miao DQ, Yue XD. Similarity measure for short texts using topic models and rough sets. Journal of Computational Information Systems, 2013, 9(16): 6603–6611.

20 王贤明, 胡智文, 谷琼. 一种基于随机 n-Grams 的文本相似度计算方法. 情报学报, 2013, 32(7): 716–723.

21 孙宇. 一种基于 Jaccard 相似度的社团发现方法. 电子技术与软件工程, 2016, (3): 20.

22 Niwattanakul S, Singthongchai J, Naenudorn E, *et al*. Using of Jaccard coefficient for keywords similarity. Proc. of the International MultiConference of Engineers and Computer Scientists. Hong Kong, China. 2013. 380–384.

23 Huang A. Similarity measures for text document clustering. New Zealand: The University of Waikato, Hamilton, 2008.

24 邓琨, 张尧学, 周悦芝. 一种整体性的相似度计算方法. 情报学报, 2014, 33(11): 1133–1145.

25 搜狗实验室. 数据资源. http://www.sougou.com/labs/resources.html. [2012-04-21].