基于 Hadoop 平台的文本相似度检测系统的研究

王小林,肖 慧,邰伟鹏
(安徽工业大学 计算机与技术学院,安徽 马鞍山 243002)

摘 要:在现有的文本相似度计算方法中,获取关键词权值的 TFIDF 算法没有完全考虑到关键词在文本中的位置和其在文本库中的离散度对权值的影响,且当处理的文本库中信息量过大时,运行效率较低。针对上述问题,文中提出一种基于语义的信息熵与信息增益的 TFIDF 算法(TFIDFWGE)。该算法通过对给定的关键词添加位置权重与计算熵值和信息增益,得到关键词的最终权值,并利用 Hadoop 平台的 Map/Reduce 框架来实现 TFIDFWGE 算法和向量空间模型(VSM)的文本相似度计算过程。通过对两组真实的数据集进行的实验结果表明,与现有的 TFIDF 算法相比,TFIDFWGE 算法的查全率和查准率更高,且在 Hadoop 平台上实现的文本相似度检测系统对信息量大的文本库处理效率更加高效。

关键词:文本相似度;语义;Map/Reduce 框架;TFIDF 算法;TFIDFWGE 算法
中图分类号:TP391 **文献标识码**:A **文章编号**:1673-629X(2015)08-0090-04
doi:10.3969/j.issn.1673-629X.2015.08.019

Research on Text Similarity Detection System Based on Hadoop

WANG Xiao-lin, XIAO Hui, TAI Wei-peng
(School of Computer Science, Anhui University of Technology, Ma'anshan 243002, China)

Abstract: In existing method of calculating similarity, TFIDF which is usually used to obtain weights of key words doesn't take into consideration the influence of key words' position and their dispersion in text library, and moreover runs in low efficiency when dealing with large quantity of data. To tackle the problems above, propose a kind of TFIDF algorithm (TFIDFWGE) based on the semantic information entropy and information gain by adding position weight to key words and calculating the entropy and information gain to acquire final value. The algorithm adds position weight and calculation entropy and information gain for given keywords to get the final weights of keywords, and use Map/Reduce framework of Hadoop platform to achieve TFIDFWGE algorithms and Vector Space Model (VSM) in the text similarity calculation process. Experimental results on two real datasets show that compared with the existing TFIDF, TFIDFWGE's recall and precision is higher, and in the Hadoop platform text similarity detection system is more efficient for information large text database processing.

Key words: text similarity; semantic; Map/Reduce framework; TFIDF; TFIDFWGE

0 引 言

文本相似度计算是文本检索技术的重要模块,广泛应用于信息检索、机器翻译、自动问答系统和文本挖掘等领域,目前主要的计算模型有向量空间模型(VSM)、潜在语义标引模型(LSI)、概率模型和基于本体论模型(Ontology)。其中 Salton 教授提出了经典的向量空间模型[1]。为了更好地计算文本相似度,Salton 和 Buckley 又提出一种添加权值的自动检索方法[2]。文本相似度计算过程中最常用的是经典的 TFIDF 权值算法[3]。由于经典的 TFIDF 算法存在一定的局限性,有些研究者将其与语义、位置等因素相结合提高查全率和查准率[4-5]。张玉芳等还将信息熵和信息增益[6-8]引入到 TFIDF 算法中进行进一步研究,效果明显。

上述算法及改进模型在一定程度上提高了传统的文本相似度计算效率和准确率。然而,并没有完全考虑到关键词在文本中位置和其在文本库中分布情况对权重的影响,且当文本相似度计算过程中算法较为复杂或者文本库信息量过大时,都会影响整个文本相似度检测系统的运行效率。为了解决这些问题,文中的主要贡献如下:

(1)提出一种基于语义和信息熵与信息增益的

收稿日期:2014-09-28 **修回日期**:2014-12-30 **网络出版时间**:2015-07-21
基金项目:国家自然科学基金资助项目(6100311);安徽省自然科学研究重点项目(KJ2013Z023,KJ2013A058)
作者简介:王小林(1964-),男,硕士,研究方向为人工智能、中文信息处理;肖 慧(1988-),男,硕士研究生,研究方向为中文信息处理。
网络出版地址:http://www.cnki.net/kcms/detail/61.1450.TP.20150721.1448.060.html

TFIDF 算法(TFIDFWGE)计算关键词的权值,并从理论上分析了这种改进算法的有效性。

(2)通过对两个真实数据集的实验分别验证了提出的 TFIDFWGE 算法的正确性和使用 Map/Reduce 框架设计的文本相似度检测系统的高效性。

1　背景知识

1.1　向量空间模型及 TFIDF 算法

G. Salton 教授提出的经典向量空间模型是一种采用线性代数的理论和方法,通过构建查询文档向量和文库向量,并计算这两个向量之间的夹角的余弦值来得到文档间的相似度。VSM 能够有效地匹配算法设计并且可以获得令人较为满意的处理结果,已经成为文本处理领域的经典方法之一。文中也是利用这一经典方法,具体可以用几何上定义的计算两个向量之间夹角的余弦值来计算相似度,如式(1)所示:

$$\mathrm{Sim}(\boldsymbol{d}_n,\boldsymbol{q})=\frac{\boldsymbol{d}_n \bullet \boldsymbol{q}}{|\boldsymbol{d}_n|\times|\boldsymbol{q}|}=\frac{\sum_{i=1}^{t} w_{i,n}\times w_{i,q}}{\sqrt{\sum_{i=1}^{t} w_{i,n}^2}\times\sqrt{\sum_{i=1}^{t} w_{i,q}^2}} \tag{1}$$

其中,$\boldsymbol{d}_n$ 为文库中的第 n 个文本的向量;$\boldsymbol{q}$ 为查询文本的向量;t 表示查询文本中的关键词个数;$w_{i,q}$ 表示查询文本第 i 个关键词的权值。$\boldsymbol{d}_n$ 中关键词个数要与 $\boldsymbol{q}$ 中的个数一致,保证都为 t 个;$w_{i,n}$ 表示文库中第 n 个文本的第 i 个关键词的权值。式中最关键参数 $w_{i,n}$ 可以根据 TFIDF 算法计算得到。

Salton 教授提出的经典的 TFIDF 算法,已经在信息检索领域计算关键词权值时得到广泛使用,这种方法主要考虑关键词在文本中出现的频率因素。其公式如下:

$$w_{i,j}=\mathrm{fre}_{i,j}(d_i)\times\mathrm{idf}(t_j)=\frac{\mathrm{fre}_{i,j}(d_i)\times\mathrm{lb}\left(\frac{N}{n_j}+0.01\right)}{\sqrt{\sum_{j=1}^{n}\left((\mathrm{fre}_{i,j}(d_i))^2\times\mathrm{lb}\left(\frac{N}{n_j}+0.01\right)^2\right)}} \tag{2}$$

其中,$\mathrm{fre}_{i,j}(d_i)$ 表示关键词 t_j 在文本 d_i 中出现的频率;$\mathrm{idf}(t_j)$ 表示关键词 t_j 的文本强度;N 表示文本库中的文本总数;n_j 表示关键词 t_j 的文本频数。

这种经典的 TFIDF 算法尽管具有很好的效果,但是还存在一定的局限性,若给定的关键词是均匀地分布在文本库中的同一个类别的每个文本中,则表明该关键词可以很好地代表此类,应该赋予较高的权值;当关键词在一个类别中大量出现,而在其他类别中出现的很少,说明这个关键词能很好地代表该类文本特征,它的分类能力应该很强,也应该赋予较高的权值;针对这一特点,应该考虑给定的关键词在整个文本库中的类内离散度和类间的分布情况对权值的影响,再次改进算法。

1.2　基于信息熵和信息增益的 TFIDF 算法

通过对经典的 TFIDF 算法的局限性分析后,文献[6]中引入信息熵加权因子,而文献[7]针对类间特征词分布的问题引入了信息增益因子来改进特征词权重计算的精度。改进后的关键词权值计算公式分别为公式(3)和(4):

$$w_{i,j}=\mathrm{fre}_{i,j}(d_i)\times\mathrm{idf}(t_j)\times E_{\mathrm{w}} \tag{3}$$

其中,信息熵加权因子 $E_{\mathrm{w}}=-\sum_{i=1}^{N}\left(\frac{\mathrm{fre}_{ji}}{\mathrm{fre}_j}\times\mathrm{lb}\left(\frac{\mathrm{fre}_{ji}}{\mathrm{fre}_j}\right)\right)+1$,$E_{\mathrm{w}}$ 中 fre_{ji} 表示关键词 t_j 在文本库的 C_k 类别的第 i 个文本中出现的频率,fre_j 表示关键词 t_j 在 C_k 类别所有的文本中出现的总频率,N 表示 C_k 类别中文本的个数。

$$w_{i,j}=\mathrm{fre}_{i,j}(d_i)\times\mathrm{idf}(t_j)\times\mathrm{IG}(C,t_k) \tag{4}$$

其中,信息增益因子 $\mathrm{IG}(C,t_k)=-\sum_{i=1}^{m}p(C_k)\times\mathrm{lb}(p(C_k))+\sum_{i=1}^{m}p(C_k/t)$,$\mathrm{IG}(C,t_k)$ 中 C 为文本库的类别集合,$P(C_k)$ 表示类别 C_k 的概率,$P(C_k/t)$ 表示关键词在类别 C_k 中出现的概率。文献[8]中综合考虑两方面因素,结合信息熵和信息增益对 TFIDF 算法进行改进,如式(5)所示:

$$w_{i,j}=\mathrm{fre}_{i,j}(d_i)\times\mathrm{idf}(t_j)\times\mathrm{IG}(C,t_k)\times E_{\mathrm{w}} \tag{5}$$

尽管这些改进后的 TFIDF 算法很高效,但是没有完全考虑到关键词在文本中的位置不同,对文本的影响也有所不同。对于这一因素,文献[4]中在经典的 TFIDF 算法基础上进一步考虑了语义加权因子。

2　TFIDFWGE 算法

结合上述对基于信息熵和信息增益的 TFIDF 算法的分析,文中在式(5)的基础上添加语义加权因子,通过语义加权因子和信息熵因子可以很好地反映关键词在类内的真实情况,即综合考虑到关键词在类内的具体位置和其分布的情况对最终权值的影响,具体公式如下所示:

$$w_{i,j}=\mathrm{fre}_{i,j}(d_i)\times\mathrm{idf}(t_j)\times W_{i,j}\times\mathrm{IG}(C,t_k)\times E_{\mathrm{w}} \tag{6}$$

其中,$W_{i,j}$ 表示关键词 t_j 在文本 d_i 中位置的权值,计算方法为 $W_{i,j}=\frac{\sum_{x=0}^{n}W_x}{n}$,$\sum_{x=0}^{n}W_x$ 为关键词 t_j 在文本 d_i 中所有的权值总和,n 为关键词 t_j 在文本 d_i 中出现的

频率。

从式(6)中可以看出,关键词的最终权值会根据语义加权因子 $W_{i,j}$ 的变化而变化,这样可以更好地反映关键词在文本中的真实分布情况。

首先,根据 Hadoop 平台处理大量小文本[9-10]特点给文本库进行预处理。然后,利用 Map/Reduce 框架[11-14]处理经过文本预处理之后的文本,依次计算出 TFIDFWGE 算法中的参数,并以键值对的形式存储到 HDFS 文件系统上。最后,读取 HDFS 文件系统上的文件,通过截取字符串获得相应的参数存储到与之对应的 Map 集合中,其中集合 hs 中存储着<文本名称@关键词,权值标记>,集合 idfmap 中存储着<关键词@,文本名称>用于建立倒排索引,集合 hc 中存储着<文本名称@关键词,关键词在此文本中出现的频率>,集合 hsc 中存储着<关键词,关键词在文库中出现的频率>,集合 hsm 中存储着<关键词,关键词在文本库的文本中出现的频率>。基于以上描述,将 TFIDFWGE 算法思想整理并描述如下:

```
1 Initialize two HashMap collections hm and map;
2 use keySet method traverse the collection hc;
3 for each hasNext()! =null do
4 execute next() method obtain the key of the collection hc;
5 if the collection hsc contains the key
6 obtain the value of this key,execute the formula(2);
7 if the collection hm contains the key then
8 obtain the value of this key and plus the current value;
9 insert <key,value> into the collection hm;
   else insert<key,0> into the collection hm;
10 use keySet method traverse the collection hs;
11 for each hasNext()! =null do
12 execute next() method obtain the key of the collection hs;
13 if the collection hc and hsc contain the key
14 obtain the value of this key,
15 execute the formula(3) and multiplied by hm.get(key);
16 insert <key,value> into the collection map;
```

TFIDFWGE 算法在基于信息熵和信息增益的 TFIDF 算法的基础上结合了语义加权因子,得到的关键词权值更加接近实际情况。首先,统计经过文本预处理后的文本中关键词出现的频率,同时可以得到关键词在整个文库中出现的总频率。然后,给关键词构建倒排索引列表,为文本相似度计算做准备。最后,遍历存储不同参数的集合,按照公式(5)计算出给定关键词的权值。

利用 TFIDFWGE 算法计算可以得到给定的关键词最终权值,从而获得文本库中每个文本的关键词权值的 Map 集合,再将这些 Map 集合作为 value,文本名称作为 key 构建一个新的 Map 集合,遍历这个新的 Map 集合,然后结合式(1)可以计算文本的相似度。

3 实验结果与分析

实验平台搭建的系统共为四台机器,每台机器配置为一个双核的 Intel(R)Core(TM)2DuoCPU T5750,主频 2.00 GHz,内存 2 G,硬盘 250 G。每台机器上的操作系统为 CentOS 6.3,Hadoop 的版本是 1.2.1。实验中采用了两个数据集,第一个数据集是从复旦大学语料库中选择 800 篇文章,其中包括计算机类、经济类、体育类、哲学类、艺术类、历史类、文化类、通信类各 100 篇。实验在单机环境下对第一个数据集,分别采用经典的 TFIDF 算法、基于信息熵和信息增益的 TFIDF 算法(TFIDFIGE 算法)和 TFIDFWGE 算法对文本进行关键词权值计算,然后用 KNN 算法分别对这三种关键词权值计算方法的结果进行对比分析,其中 K 的取值均为 100。实验结果如表 1 和表 2 所示。

表 1 三种关键词权值算法的查全率 %

类别	TFIDF 算法	TFIDFIGE 算法	TFIDFWGE 算法
计算机	90.5	67.0	72.7
经济	95.4	94.0	93.0
体育	90.1	90.0	94.0
哲学	80.0	79.0	88.7
艺术	86.2	91.2	95.8
历史	73.4	82.4	78.2
文化	61.0	83.9	80.0
通信	90.0	93.0	98.9
均值	83.3	85.1	87.7

从表 1 中可以看出,TFIDFIGE 算法的查全率都比经典的 TFIDF 算法提高了 1.8%。而 TFIDFWGE 算法的查全率又比 TFIDFIGE 算法提高了 2.5%。通过实验数据可以看出,改进后的算法查全率比现有的 TFIDF 算法要高。

表 2 三种关键词权值算法的查准率 %

类别	TFIDF 算法	TFIDFIGE 算法	TFIDFWGE 算法
计算机	76.0	75.0	79.5
经济	80.3	85.4	92.3
体育	83.3	80.5	76.1
哲学	78.1	84.0	87.4
艺术	84.4	87.1	90.1
历史	69.0	78.7	70.8
文化	80.4	82.3	83.8
通信	91.8	92.4	95.3
均值	80.4	83.2	84.4

从表 2 中可以看出,TFIDFIGE 算法的查准率都比经典的 TFIDF 算法提高了 2.8%。而 TFIDFWGE 算法的查准率比 TFIDFIGE 算法提高了 1.2%。这说明改

进后的算法不仅仅只是提高了查全率,对查准率还有一定程度的提高。通过对表 1 和表 2 的实验数据分析,验证了 TFIDFWGE 算法的正确性和高效性。

为了验证在 Hadoop 平台上利用 TFIDFWGE 算法实现的文本相似度检测系统的高效性,第二组实验采用的数据集是来自搜狐新闻 2008 年 1 ~ 6 月期间奥运会、体育、国内、国际等 18 个频道包含 15 个类别的搜狐新闻数据(SogouCS),其中每个文本中都提供了 URL 和正文信息,SogouCS 的大小为 3.33 G。利用 SogouCS 数据集在不同节点个数的 Hadoop 集群上运行文本相似度计算,实验结果如图 1 所示。

从图 1 中的实验结果可以看出,在 Hadoop 平台下实现的文本相似度检测系统,在采用复杂的 TFIDFWGE 算法处理大信息量的文本时仍然高效。当集群的节点数从 1 个逐渐增加到 4 个时,文本相似度的计算时间虽然逐渐减少,但不是呈线性递减,主要原因是随着集群节点个数的增加,各个节点间通信和任务分配调度的耗时增多,从而影响整个系统的运行速度。

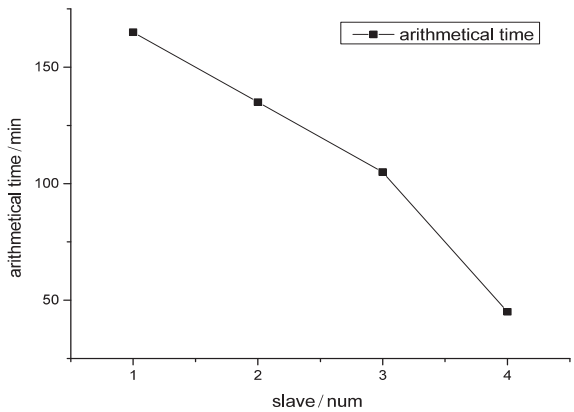

图 1　文本相似度计算时间

4　结束语

文中提出的改进后的 TFIDF 算法(TFIDFWGE),在查全率和查准率方面比现有的 TFIDF 算法有一定的提高,鉴于算法实现过程较为繁琐,采用 Hadoop 平台的 Map/Reduce 架构实现该算法,可以让其在处理大信息量的文本时运行效率依然能够得到保证。在此基础上,利用 Map/Reduce 架构实现文本相似度计算,进一步验证 Hadoop 平台处理大信息量文本库的高效性。

在使用文中提出的方法处理时也存在几个问题:由于实验条件有限,对新的方法只是进行了初步仿真,实验中使用的文库也较小,且只是在少量节点的集群上做了测试;另外,程序整体运行效率不是特别理想,需要改进。下一步,可以通过优化算法和提高 Hadoop 平台的整体效能从而在一定程度上提高文中提出方法的处理效率。

参考文献:

[1] Salton G, Wong A, Yang C S. A vector space model for automatic indexing[J]. Communication of ACM, 1975, 18(11): 613-620.

[2] Salton G, Buckley C. Term-weighting approaches in automatic retrieval[J]. Information Processing and Management, 1988, 24(5): 513-523.

[3] 施聪莺,徐朝军,杨晓江. TFIDF 算法研究综述[J]. 计算机应用,2009,29(S1):167-170.

[4] 龚　静,周经野. 一种基于多重因子加权的文本特征项权值计算方法[J]. 计算技术与自动化,2007,26(1):81-83.

[5] 李媛媛,马永强. 基于潜在语义索引的文本特征词权重计算方法[J]. 计算机应用,2008,28(6):1460-1462.

[6] 周炎涛,唐剑波,王家琴. 基于信息熵的改进 TFIDF 特征选择算法[J]. 计算机工程与应用,2007,43(35):156-158.

[7] 张玉芳,陈小莉,熊忠阳. 基于信息增益的特征词权重调整算法研究[J]. 计算机工程与应用,2007,43(35):159-161.

[8] 李学明,李海瑞,薛　亮,等. 基于信息增益和信息熵的 TFIDF 算法[J]. 计算机工程,2012,38(8):37-40.

[9] 向小军,高　阳,商　琳,等. 基于 Hadoop 平台的海量文本分类的并行化[J]. 计算机科学,2011,38(10):184-188.

[10] 王润华. 基于 Hadoop 集群的分布式日志分析系统研究[J]. 科学信息,2007(15):60-60.

[11] Dean J. Experiences with MapReduce, an abstraction for large-scale computation[C]//Proc of 15th international conference on parallel architectures and compilation techniques. [s. l.]:[s. n.],2006.

[12] TOM Wbite. Hadoop: the definitive guide[M]. US: O'Reilly, 2005.

[13] Lammel R. Google's MapReduce programming model-revisited[J]. Science of Computer Programming, 2008, 70(6): 1-30.

[14] 李成华,张新访,金　海,等. MapReduce:新型的分布式并行计算编程模型[J]. 计算机工程与科学,2011,33(3):129-135.

基于Hadoop平台的文本相似度检测系统的研究

作者： 王小林， 肖慧， 邰伟鹏， WANG Xiao-lin， XIAO Hui， TAI Wei-peng
作者单位： 安徽工业大学 计算机与技术学院,安徽 马鞍山,243002
刊名： 计算机技术与发展 ISTIC
英文刊名： Computer Technology and Development
年，卷(期)： 2015(8)