基于云计算的改进余弦向量度量法索引项权值算法①

付永贵, 尚成国, 马尚才

(山西财经大学 信息管理学院, 太原 030031)

摘 要: 针对用户对索引项重要程度无区分以及普通计算平台承载能力受限提出基于云计算的改进余弦向量度量法索引项权值算法(ICVMMITWCC 算法), 该算法通过从包含全部索引项的文本集中计算索引项平均权值对经典的余弦向量度量法索引项权值算法(CCVMMITW 算法)修改求得相对权值; 通过实验对比 ICVMMITWCC 算法与 CCVMMITW 算法下文本的排序效率, 说明 ICVMMITWCC 算法更贴近用户查询需求.

关键词: ICVMMITWCC 算法; 云计算; 平均权值; 相对权值; 排序效率

Improved Cosine Vector Measuring Method Indexing Term Weight Algorithm Based on Cloud Computing

FU Yong-Gui, SHANG Cheng-Guo, MA Shang-Cai

(School of Information Management, Shanxi University of Finance and Economics, Taiyuan 030031, China)

Abstract: Aiming the importance degree of indexing terms hadn't distinguish to users and the common computing platform carrying capacity limited proposed the improved cosine vector measuring method indexing term weight algorithm based on cloud computing (ICVMMITWCC algorithm), the algorithm corrected the classic cosine vector measuring method indexing term weight algorithm (CCVMMITW algorithm) to acquire relative weight by computing indexing term average weight from texts set that included all indexing terms; comparing the efficiency of sorting of texts in ICVMMITWCC algorithm and CCVMMITW algorithm by experiment, indicating ICVMMITWCC algorithm is closer to the query requirement of users.

Key words: ICVMMITWCC algorithm; cloud computing; average weight; relative weight; the efficiency of sorting

1 引言

在进行文本检索时, 经典的向量空间模型余弦向量度量法索引项权值是运用 tf-idf 公式来确定的, 这样当不同索引项在文本中出现的频率不同时, 不同索引项在文本与查询相似度计算中的贡献将不同; 在用户对各索引项重要程度无区分的情况下, 如果某一索引项由于“词义”或“词性”等语言环境特性在查询相关文本集中的文本中出现的频率表现出较其余索引项普遍较高的趋势时, 这一索引项将会在文本与查询相似度计算中表现出普遍较大的贡献, 这种忽略索引项所处的语言环境的索引项权值计算方法会降低查询的效率. 针对经典余弦向量度量法索引项权值设定的不足, 学者们提出了不同的设定方法, 比如苏小虎[1]提出将经典的余弦向量度量法权值计算方法与索引项本身的重要程度结合起来计算索引项权值, 但其研究没能反映索引项本身“词义”或“词性”等语言环境特性对索引项在文本集中文本出现频率的影响. 蓝海洋等[2]提出根据索引项在文本中出现的频率与在整个文本集中出现的平均频率之间的相对值来计算索引项的权值, 但其在计算索引项在文本集中出现的平均频率时忽略了文本集中文本与查询的相似程度不同所反映其文本结构的个性化特征, 相似程度很低的文本中索引项出现频率加入平均频率的计算会影响平均频率计算的精度, 同时其研究完全独立于 tf-idf 公式, 不能有效地反映文本长度不同对相似度计算造成的影响. 综合学者们的研究发现, 目前对余弦向量度量法索引项权值设定的

① 基金项目:山西省科技基础条件平台建设项目(2011091001-0101)
收稿时间:2013-01-15;收到修改稿时间:2013-02-28

研究多数假定不同的索引项对区分文本与查询相似性的重要程度是不同的, 少数研究者在不考虑索引项对区分文本与查询相似度重要程度的情况下尝试借助文本集及文本中索引项出现频率来确定文本中索引项权值, 但由于索引项信息提取的片面性及不准确性导致权值设定有效性受到影响.

基于以上分析, 在使用余弦向量度量法进行文本检索时, 本文针对用户对索引项重要程度无区分以及索引项之间有具体的语言环境特性的检索情况, 对经典余弦向量度量法文本检索模型索引项权值计算方法进行改进, 提出了适于这一情况的“改进余弦向量度量法索引项权值算法”(the improved cosine vector measuring method indexing term weight algorithm), 简记为 ICVMMITW 算法.

ICVMMITW 算法的算法复杂度较高, 在用于文本检索时, 当目标文本集文本数量及容量很大时, 会由于算法复杂度及目标文本集容量问题导致通常的计算平台难以承载. 云计算[3]技术的出现及其应用为大容量数据的复杂计算提供了可能, 在云计算平台下可以实现 ICVMMITW 算法对大容量数据的计算求解. 相应地这一平台下的 ICVMMITW 算法本文称为“基于云计算的改进余弦向量度量法索引项权值算法”(the improved cosine vector measuring method indexing term weight algorithm based on cloud computing), 简记为 ICVMMITWCC 算法, 并将“经典的余弦向量度量法索引项权值算法”(the classic cosine vector measuring method indexing term weight algorithm) 简记为 CCVMMITW 算法.

2 经典的余弦向量度量法文本检索模型

余弦向量度量法在应用于文本检索时, 文本和查询均被看成由索引项构成的向量, 比如由 n 个索引项构成的文本检索, 查询 q 和文本 d_j 可以分别表示为: $q=(t_{1q},t_{2q},\cdots,t_{nq})$, $d_j=(s_{1j},s_{2j},\cdots,s_{nj})$, 其中 $t_{kq},s_{kj}(1\leqslant k\leqslant n)$分别表示查询 q 和文本 d_j 所包含的第 k 个索引项. 在具体应用中, 通常用索引项的权值构成的空间向量来表示查询和文本, 索引项在查询 q 和文本 d_j 中的权值表示索引项对查询 q 和文本 d_j 表示的贡献大小, 基于此, 查询 q 和文本 d_j 可以分别表示为: $q=(w_{1q},w_{2q},\cdots,w_{nq})$, $d_j=(v_{1j},v_{2j},\cdots,v_{nj})$, 其中 $w_{kq},v_{kj}(1\leqslant k\leqslant n)$分别表示第 k 个索引项在查询 q 和文本 d_j 中的权值[4].

2.1 CCVMMITW 算法

目前经典的确定余弦向量度量法索引项权值的方法是 *tf-idf* 公式, 则第 k 个索引项在文本 d_j 中的权值 v_{kj} 可以计算为 $v_{kj}=tf_{kj}/max_j\{tf_{kj}\}\times lg(N\times idf_k)$. 其中 tf_{kj} 表示第 k 个索引项在文本 d_j 中出现的频率, tf_{kj} 值表示第 k 个索引项对描述文本内容的能力大小; idf_k 是逆文本频率, 它表示文本集 D 中出现第 k 个索引项的文本数的倒数, idf_k 值表示第 k 个索引项区分其所在文本与其他文本的能力; $max_j\{tf_{kj}\}$表示文本 d_j 中出现频率最高的索引项的频率, $max_j\{tf_{kj}\}$值是为了消除具体应用中长文本相对于短文本的优势而设置的; N 表示文本集 D 中文本的个数[4]. 因为本文的研究只是针对用户对索引项重要程度无区分的情况下索引项由于“词性”、“词义”等语言环境特性的不同对索引项权值计算造成的影响, 所以本文在研究中剔除索引项的 idf_k 值对其权值计算的影响, 将文本中索引项权值计算公式简化为 $v_{kj}=tf_{kj}/max_j\{tf_{kj}\}$, 事实上已经有很多学者在具体领域的研究中也进行过类似的处理, 比如文献[5].

2.2 余弦向量度量法

余弦向量度量法在用于文本检索时是用索引项权值空间向量表示的查询 q 和文本 d_j 夹角的余弦值来表示 q 和 d_j 的相似度, 则查询 q 和文本 d_j 的相似度值 $\mathrm{sim}(d_j,q)$定义为[4]:

$$\mathrm{sim}(d_j,q)=\cos(d_j,q)=\frac{\sum_{k=1}^{n}(w_{kq}\times v_{kj})}{\sqrt{\sum_{k=1}^{n}w_{kq}^{2}}\times\sqrt{\sum_{k=1}^{n}v_{kj}^{2}}}$$

3 云计算概述

云计算是并行计算、分布式计算和网格计算[6]的发展[7], 借助云计算技术, 用户可以畅享网络, 实现大容量数据的超级计算[8]. 云计算向人们隐藏了“云”端底层实现的细节, 用户不必关心数据是存储在哪里, 也不必关心“云”端任务是如何调度的, 用户只要在需要的时候向“云”端请求计算或者其他服务即可.

云计算从 2007 年底开始被人们关注[9], 近几年得到了迅速的发展, 其应用研究涉及到超级计算、数据存储、数据库托管、平台托管等多个领域, 其中最突出的应用研究则是超级计算, 目前学者们借助云计算技术在大容量数据复杂计算领域的研究中取得了进一

步的成果.

4 ICVMMITWCC 算法

基于以上分析及理论, 本文提出的 ICVMMITWCC 算法的具体实现步骤如下:

(1) 根据用户查询要求, 对目标文本集 D 进行初始检索, 获得包含索引项的文本组成相关文本集 D', D'中文本数量记为 N; 从 D'中分离出索引项全部出现的文本构成文本集 D'', D''中文本数量记为 N'.

(2) 设定查询 q 包含各索引项的权值, 因为本文研究中用户对查询所包含的各索引项重要程度无区分, 所以本文设查询 q 由索引项权值所表示的空间向量表达式为 $q=(w_{1q},w_{2q},\cdots,w_{nq})=(1,1,\cdots,1)$; 计算 D'中各文本包含各索引项权值 $v_{kj}=tf_{kj}/max_j\{tf_{kj}\}$.

(3) 对 D''中各文本各索引项权值按照步骤(2)中的计算结果求其平均值, 对于第 k 个索引项其权值的均值记为 $\overline{v_{kj}}=\sum v_{kj}/N'$, 调整相关文本集 D'中各文本各索引项权值为 $v_{kj}=v_{kj}/\overline{v_{kj}}$.

(4) 将 D'中各文本修改后的索引项权值空间向量与查询 q 使用余弦向量度量法进行相似度计算, 并按计算结果降序排列各文本.

其用户查询流程如图 1 所示.

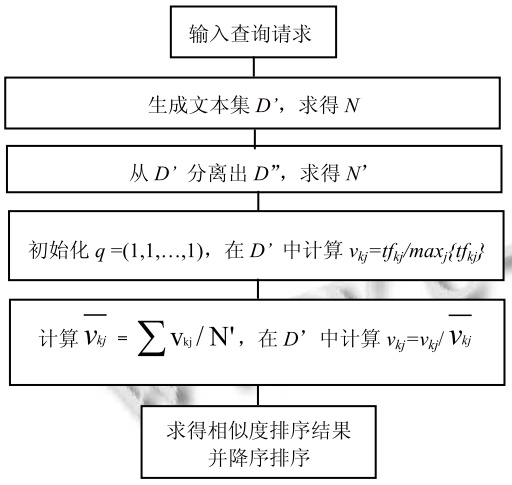

图 1　ICVMMITWCC 算法用户查询流程

5 实验及分析

自从云计算的概念被提出以后, 不断有 IT 厂商推出自己的云计算平台, 其中开源云计算系统在科研实验方面得到了人们的共识, 在这些开源云计算系统中, Hadoop MapReduce 是一个使用简易的软件框架, 在实验研究中得到了人们广泛的使用. 由于篇幅有限, 本文只对 Hadoop MapReduce 做简要的介绍, 有关 Hadoop MapReduce 的详细知识可以参考有关的专业书籍. 在 Hadoop MapReduce 环境中对 CCVMMITW 算法与 ICVMMITWCC 算法计算结果使用余弦向量度量法对文本检索排序的效率进行了对比分析, 验证了云计算平台对大容量数据复杂计算的承载能力.

5.1 MapReduce 简介

MapReduce 是一种分布式编程框架, 这一任务被分为两个处理阶段: map 阶段和 reduce 阶段. 每个阶段都以键/值对作为输入和输出. 其中 map 函数将一个任务分解成为多个任务, 把输入的键/值对进行处理, 形成中间形式的键/值对, 系统按照键/值把中间形式的值集中起来, 传递给用户指定的 reduce 函数. 而 reduce 函数是一个归约过程, 它将具有相同键的值合并在一起并最终输出[10].

对于复杂的算法流程, 可以通过多个 MapReduce 过程来实现.

5.2 实验环境及数据

本文的实验环境由 10 台普通的联想 PC 机组成, 其中 1 台 PC 机作为主结点, 其余 9 台 PC 机作为从结点(从结点数量在实验中可以由 1 到 9 变化). 软件环境使用 Linux 作为操作系统, hadoop 作为云计算平台, 编程环境涉及 eclipse 以及 JDK.

实验数据来源于山西焦煤集团古交五大矿井(屯兰矿、东曲矿、西曲矿、马兰矿、镇城底矿)1997-2011 年的生产调度日志共 22538 个文本, 容量为 1.02G. 在实验中从中随机抽取 100、5000、10000、15000、20000 个文本作为目标文本集 D 进行实验分析. 其中随机抽取 100 个文本是为了比较 CCVMMITW 算法与 ICVMMITWCC 算法计算结果使用余弦向量度量法对文本与查询相似度排序的效率, 其余随机抽取数量的文本是为了验证 Hadoop MapReduce 云计算系统处理大容量数据的能力.

实验中设查询 $q=(t_{1,q},t_{2,q},t_{3,q})$=(煤炭生产, 产量, 安全). 表 1 列出随机抽取的目标文本集 D 中文本数量为 100 个文本时, D 中文本 d_j 按索引项频率值表示的文本结构, 其中 $s_{1,j},s_{2,j},s_{3,j}$ 表示文本 d_j 中的索引项, $tf_{1,j},tf_{2,j},tf_{3,j}$ 分别表示 $s_{1,j},s_{2,j},s_{3,j}$ 在文本 d_j 中的频率值.

5.3 实验过程及结果分析

实验过程中将 CCVMMITW 算法与 ICVMMITWCC

算法计算结果使用余弦向量度量法对文本与查询相似度排序, 实验数据为随机抽取的 5000、10000、15000、20000 个文本, 在 Hadoop MapReduce 中运行时, 当实验环境中的从结点数量由少增多时, 两个算法计算结果使用余弦向量度量法对文本排序的运行时间都对应由长变短, 而且都有对应呈反比例变化的趋势. 当从节点数量不变, 随机抽取的文本数量由少变多, 容量由小变大时, 两个算法计算结果使用余弦向量度量法对文本排序的运行时间也对应由短变长. 从实验可以说明 Hadoop MapReduce 作为云计算平台当实验用从节点数量增多时其运行时间表现出对应呈反比例减少的趋势.

表 1 文本集 D 索引项频率结构

	$tf_{1,j}$	$tf_{2,j}$	$tf_{3,j}$		$tf_{1,j}$	$tf_{2,j}$	$tf_{3,j}$		$tf_{1,j}$	$tf_{2,j}$	$tf_{3,j}$
d_1	0	0	0	d_{35}	0	0	0	d_{69}	10	10	10
d_2	0	0	0	d_{36}	0	0	0	d_{70}	0	0	0
d_3	4	50	0	d_{37}	0	0	0	d_{71}	1	15	2
d_4	0	0	0	d_{38}	0	0	3	d_{72}	0	0	0
d_5	0	0	0	d_{39}	0	0	0	d_{73}	0	0	0
d_6	0	0	0	d_{40}	0	0	0	d_{74}	3	15	2
d_7	0	0	0	d_{41}	0	0	0	d_{75}	0	0	0
d_8	0	0	0	d_{42}	2	46	4	d_{76}	15	78	9
d_9	8	45	3	d_{43}	0	0	0	d_{77}	0	0	0
d_{10}	0	0	0	d_{44}	0	0	0	d_{78}	0	25	19
d_{11}	0	0	0	d_{45}	21	45	11	d_{79}	0	0	0
d_{12}	0	0	0	d_{46}	17	0	29	d_{80}	3	96	3
d_{13}	1	0	0	d_{47}	0	0	0	d_{81}	0	0	0
d_{14}	11	0	7	d_{48}	0	0	0	d_{82}	0	0	0
d_{15}	0	0	0	d_{49}	0	0	0	d_{83}	1	82	6
d_{16}	2	55	4	d_{50}	3	50	0	d_{84}	0	0	0
d_{17}	0	0	0	d_{51}	2	15	4	d_{85}	0	0	0
d_{18}	0	0	0	d_{52}	0	0	0	d_{86}	0	0	3
d_{19}	0	0	0	d_{53}	0	0	0	d_{87}	0	89	4
d_{20}	0	0	0	d_{54}	0	0	0	d_{88}	0	0	0
d_{21}	9	65	4	d_{55}	2	23	2	d_{89}	0	0	0
d_{22}	0	0	0	d_{56}	0	0	0	d_{90}	0	0	0
d_{23}	0	4	0	d_{57}	0	0	0	d_{91}	0	0	0
d_{24}	7	60	11	d_{58}	0	0	0	d_{92}	0	0	0
d_{25}	0	0	0	d_{59}	0	0	0	d_{93}	0	0	0
d_{26}	0	0	0	d_{60}	3	87	14	d_{94}	0	0	0
d_{27}	10	73	7	d_{61}	0	0	0	d_{95}	0	0	0
d_{28}	0	0	0	d_{62}	0	67	2	d_{96}	0	0	0
d_{29}	0	0	0	d_{63}	0	0	0	d_{97}	0	0	0
d_{30}	0	0	0	d_{64}	0	0	0	d_{98}	0	0	0
d_{31}	5	2	3	d_{65}	0	0	0	d_{99}	0	0	0
d_{32}	0	0	0	d_{66}	0	0	0	d_{100}	2	34	3
d_{33}	0	0	0	d_{67}	0	0	0				
d_{34}	1	21	2	d_{68}	0	0	0				

实验抽取目标文本集 D 中文本数量为 20000 个, 文本总容量为 956M, 当从结点数量为 3 台 PC 机时, 求解 ICVMMITWCC 算法计算结果及文本 d_j 与查询 q 相似度的运行时间是 5585 秒, 从结点数量为 9 台 PC 机时, 运行时间是 1971 秒.

由以上分析可知, Hadoop MapReduce 作为云计算平台是可以承载大容量数据的复杂计算的.

以下分别将 CCVMMITW 算法与 ICVMMITWCC 算法计算结果使用余弦向量度量法对文本与查询相似度进行排序计算, 然后对其排序的效率进行对比分析.

5.3.1 CCVMMITW 算法结果使用余弦向量度量法排序

在实验过程中, 对于 D(表 1 所列文本)中不包括任何索引项的文本进行剔除, 求得相关文本集 D'=(d_3,d_9,d_{13},d_{14},d_{16},d_{21},d_{23},d_{24},d_{27},d_{31},d_{34},d_{38},d_{42},d_{45},d_{46},d_{50},d_{51}, d_{55},d_{60},d_{62},d_{69},d_{71},d_{74},d_{76},d_{78},d_{80},d_{83},d_{86},d_{87}, d_{100}).

因为本文研究中用户对查询 q 各索引项重要程度无区分, 所以本文设 q=(w_{1q},w_{2q},w_{3q})=(1,1,1).

对于 D'中文本, 按照 CCVMMITW 算法, 得到 D'中文本的索引项权值结构, 如表 2 所示.

表 2 CCVMMITW 算法索引项权值结构

	$v_{1,j}$	$v_{2,j}$	$v_{3,j}$		$v_{1,j}$	$v_{2,j}$	$v_{3,j}$
d_3	0.08	1	0	d_{50}	0.06	1	0
d_9	0.178	1	0.067	d_{51}	0.133	1	0.267
d_{13}	1	0	0	d_{55}	0.087	1	0.087
d_{14}	1	0	0.636	d_{60}	0.034	1	0.161
d_{16}	0.036	1	0.073	d_{62}	0	1	0.03
d_{21}	0.138	1	0.062	d_{69}	1	1	1
d_{23}	0	1	0	d_{71}	0.067	1	0.133
d_{24}	0.117	1	0.183	d_{74}	0.2	1	0.133
d_{27}	0.137	1	0.096	d_{76}	0.192	1	0.115
d_{31}	1	0.4	0.6	d_{78}	0	1	0.76
d_{34}	0.048	1	0.095	d_{80}	0.031	1	0.031
d_{38}	0	0	1	d_{83}	0.012	1	0.073
d_{42}	0.043	1	0.087	d_{86}	0	0	1
d_{45}	0.467	1	0.244	d_{87}	0	1	0.045
d_{46}	0.586	0	1	d_{100}	0.059	1	0.088

使用余弦向量度量法计算文本 d_j 与查询 q 的相似度值, 其计算结果为:

sim(d_3,q)=0.622,sim(d_9,q)=0.706,sim(d_{13},q)=0.577,
sim(d_{14},q)=0.797,sim(d_{16},q)=0.638,sim(d_{21},q)=0.685,
sim(d_{23},q)=0.577,sim(d_{24},q)=0.733,sim(d_{27},q)=0.702,
sim(d_{31},q)=0.937,sim(d_{34},q)=0.656,sim(d_{38},q)=0.577,
sim(d_{42},q)=0.649,sim(d_{45},q)=0.874,sim(d_{46},q)=0.79,
sim(d_{50},q)=0.611,sim(d_{51},q)=0.775,sim(d_{55},q)=0.673,
sim(d_{60},q)=0.681,sim(d_{62},q)=0.594,sim(d_{69},q)=1,
sim(d_{71},q)=0.685,sim(d_{74},q)=0.748,sim(d_{76},q)=0.736,
sim(d_{78},q)=0.809,sim(d_{80},q)=0.613,sim(d_{83},q)=0.625,
sim(d_{86},q)=0.577,sim(d_{87},q)=0.603,sim(d_{100},q)=0.659.

按相似度计算结果对 D'中文本降序排序, 其排序结果为:

d_{69},d_{31},d_{45},d_{78},d_{14},d_{46},d_{51},d_{74},d_{76},d_{24},d_9,d_{27},d_{21},d_{71},d_{60},d_{55}, d_{100},d_{34},d_{42},d_{16},d_{83},d_3,d_{80},d_{50},d_{87},d_{62},d_{13},d_{23},d_{38},d_{86}.

5.3.2 ICVMMITWCC 算法结果使用余弦向量度量法排序

(1) 从 D'中求得索引项全部出现的文本形成文本集 D''=(d_9,d_{16},d_{21},d_{24},d_{27},d_{31},d_{34},d_{42},d_{45},d_{51},d_{55},d_{60},d_{69},d_{71}, d_{74},d_{76},d_{80},d_{83},d_{100}).

可知 D''中文本数量 N'=19, 对于各个索引项, 在 D''中求各索引项权值的均值, 对于索引项 $s_{1,q}$, 其权值的均值为 $\overline{v_{1j}}=\sum v_{1j}/N'=0.133$, 同理可得: $\overline{v_{2j}}=0.613$, $\overline{v_{3j}}=0.120$.

(2) 修改索引项权值为 $v_{kj}=v_{kj}/\overline{v_{kj}}$，修改后 D'中各文本索引项权值结构如表 3 所示.

表 3　ICVMMITWCC 算法索引项权值结构

	$v_{1,j}$	$v_{2,j}$	$v_{3,j}$		$v_{1,j}$	$v_{2,j}$	$v_{3,j}$
d_3	0.603	1.63	0	d_{50}	0.452	1.63	0
d_9	1.342	1.63	0.559	d_{51}	1.003	1.63	2.228
d_{13}	7.54	0	0	d_{55}	0.656	1.63	0.726
d_{14}	7.54	0	5.307	d_{60}	0.256	1.63	1.344
d_{16}	0.271	1.63	0.609	d_{62}	0	1.63	0.25
d_{21}	1.04	1.63	0.517	d_{69}	7.54	1.63	8.345
d_{23}	0	1.63	0	d_{71}	0.505	1.63	1.11
d_{24}	0.882	1.63	1.527	d_{74}	1.508	1.63	1.11
d_{27}	1.033	1.63	0.801	d_{76}	1.448	1.63	0.96
d_{31}	7.54	0.652	5.007	d_{78}	0	1.63	6.342
d_{34}	0.362	1.63	0.793	d_{80}	0.234	1.63	0.259
d_{38}	0	0	8.345	d_{83}	0.09	1.63	0.609
d_{42}	0.324	1.63	0.726	d_{86}	0	0	8.345
d_{45}	3.521	1.63	2.036	d_{87}	0	1.63	0.376
d_{46}	4.418	0	8.345	d_{100}	0.445	1.63	0.734

使用余弦向量度量法计算文本 dj 与查询 q 的相似度值，其计算结果为:

$sim(d_3,q)=0.742$,$sim(d_9,q)=0.933$,$sim(d_{13},q)=0.577$,
$sim(d_{14},q)=0.804$,$sim(d_{16},q)=0.823$,$sim(d_{21},q)=0.919$,
$sim(d_{23},q)=0.577$,$sim(d_{24},q)=0.971$,$sim(d_{27},q)=0.957$,
$sim(d_{31},q)=0.840$,$sim(d_{34},q)=0.870$,$sim(d_{38},q)=0.577$,
$sim(d_{42},q)=0.853$,$sim(d_{45},q)=0.947$,$sim(d_{46},q)=0.780$,
$sim(d_{50},q)=0.711$,$sim(d_{51},q)=0.956$,$sim(d_{55},q)=0.915$,
$sim(d_{60},q)=0.876$,$sim(d_{62},q)=0.658$,$sim(d_{69},q)=0.890$,
$sim(d_{71},q)=0.920$,$sim(d_{74},q)=0.988$,$sim(d_{76},q)=0.979$,
$sim(d_{78},q)=0.703$,$sim(d_{80},q)=0.735$,$sim(d_{83},q)=0.772$,
$sim(d_{86},q)=0.577$,$sim(d_{87},q)=0.692$,$sim(d_{100},q)=0.880$.

按相似度计算结果对 D'中文本降序排序，其排序结果为:

$d_{74},d_{76},d_{24},d_{27},d_{51},d_{45},d_9,d_{71},d_{21},d_{55},d_{69},d_{100},d_{60},d_{34},d_{42},d_{31},d_{16},d_{14},d_{46},d_{83},d_3,d_{80},d_{50},d_{78},d_{87},d_{62},d_{13},d_{23},d_{38},d_{86}$.

5.3.3 实验结果分析

为了便于对比分析，分别对 CCVMMITW 算法与 ICVMMITWCC 算法计算结果使用余弦向量度量法得出的任一文本 d_j 与查询 q 的相似度值求其相对值，即 $sim(d_j,q)'=sim(d_j,q)/\sum_{j=1}^{30}sim(d_j,q)$ (因为 D'中有 30 个文本)，$sim(d_j,q)'$表示 D'中某一文本 d_j 在不同的索引项权值求解算法下相对于文本集 D'中其他文本来说与查询 q 的相对相似程度，具体如图 2 所示.

对图 2 进行分析可以看出:

(1) CCVMMITW 算法计算结果使用余弦向量度量法对文本排序时文本 d_{69},d_{31} 排在了文本 $d_9,d_{24},d_{27},d_{74},d_{76}$ 前面，索引项没全出现的文本 d_{78} 排在了包含全部索引项的文本 $d_9,d_{16},d_{21},d_{34},d_{42},d_{100}$ 前面，是因为 CCVMMITW 算法没有考虑各个索引项所处的语言环境，没有考虑索引项由于“词性”、“词义”等语言特性对索引项在文本中出现频率的影响，ICVMMITWCC 算法通过对索引项全部出现的文本进行分析提取索引项之间的频率关系对索引项权值计算方法进行改进，使得该算法计算结果使用余弦向量度量法对文本排序时文本 $d_9,d_{24},d_{27},d_{74},d_{76}$ 排在了 d_{69},d_{31} 前面，$d_9,d_{16},d_{21},d_{34},d_{42},d_{100}$ 排在了 d_{78} 前面，这样的排序结果更符合用户需求，同样在 ICVMMITWCC 算法下其它一些更符合用户需求的文本一定程度上排序也得到了前移.

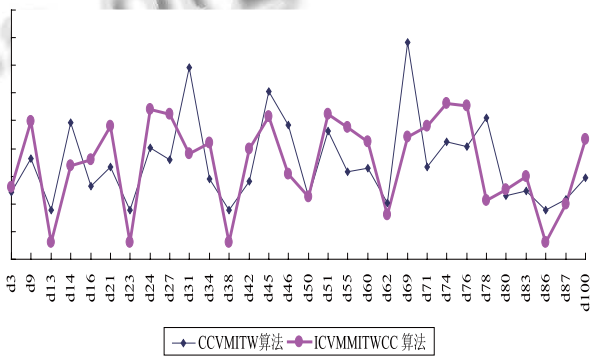

图 2　不同算法下相对相似度对比图

(2) CCVMMITW 算法与 ICVMMITWCC 算法计算结果使用余弦向量度量法对文本的排序具有一定的相似之处，但由于 ICVMMITWCC 算法更多地考虑了索引项所处的语言环境，消除了具体语言环境对索引项权值计算造成的影响，使得其排序更加贴近用户需求.

6　结语

本文针对经典余弦向量度量法在计算索引项权值时缺乏对索引项所处的语言环境进行考虑的不足，以及普通计算平台对大容量数据复杂计算的局限性，提出在云计算平台下从相关文本集一些特定文本中根据索引项所处的语言环境提取索引项在各文本出现的频率信息，通过对这些频率信息进行分析然后修改经典余弦向量度量法索引项权值计算结果的 ICVMMITWCC 算法；借助 Hadoop MapReduce 云计算平台验证了 ICVMMITWCC 算法较 CCVMMITW 算法可以提高余弦向量度量法的检索效率. 本文的研究对具体语言环境下余弦向量度量法索引项权值的计算具有一定的参考价值，以后将继续努力，进一步深化这一领域的研究成果.

(下转第 167 页)

(2) SMSLIB 运行环境配置问题

在 Linux 环境下, JRE 中添加串口通讯组件和 Windows 环境下存在差别: 要将组件中使用到的二进制文件添加到“java.library.path”中[8].

(3) 串口名称问题

在Windows系统下, 串口名称一般为“Com1”或者“Com2”, 但是在 Linux 系统下串口的名称一般为/dev/ttyS0 或者/dev/ttyS1, 系统移植过程中要对相应的参数进行修改.

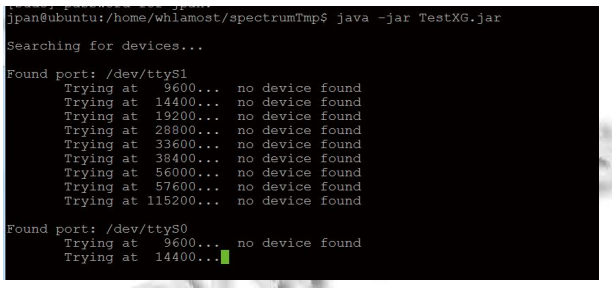

图 4 Ubuntu 中 SMSLIB 对短信发送设备测试的运行界面

3 结语

本文讨论了通过调用 SMSLIB 实现对物理设备 GSM MODEM 的操作以及 SMSLIB 的参数设置, 着重解释了集成了 SMSLIB 的系统如何进行跨平台移植的问题. 由此建立的短信模块已经应用到 LASAC 协同平台中, 为LAMOST光谱分析及协同工作提供短信发送的功能, 取得了很好的应用效果.

参考文献

1 程世繁,汪秉文.基于 SMSLib 的数据采集系统设计和实现.计算机与数字工程,2011,(12):62–65.

2 华敏敏.大学生创意大赛管理系统中基于 Smslib 的短信应用研究.电脑知识与技术,2011,(3):661–662.

3 王薇,杨婧.短信收发平台的设计与实现.嘉兴学院学报,2010,(S1):173–176.

4 金丹.基于 GSM 手机短信平台的设计与实现.江汉大学学报(自然科学版),2009,(2):46–51.

5 赵大成,贾海燕.手机短信收发的 AT 指令控制.信息工程大学学报,2004,(2):90–92.

6 葛磊蛟,姚素娟,毛一之,李歧.基于 C~#.NET 的 GSM MODEM 短信猫应用设计开发.现代电子技术,2009,(6):94–96.

7 张宇,张悦.Java 多线程机制的研究.信息与电脑(理论版),2011,(3):107.

8 http://smslib.org,2012.6.21

(上接第 90 页)

参考文献

1 苏小虎.基于改进 VSM 的句子相似度研究.计算机技术与发展,2009,19(8):113–116.

2 蓝海洋,周杰韩,张和明.文本索引词项相对权重计算方法与应用.计算机工程与应用,2003,15:68–70.

3 Armbrust M, Fox A, Griffith R, et al. Above the Clouds: A Berkeley View of Cloud Computing.[2012-06-20].http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html.

4 王众托,吴江宁,郭崇慧.信息与知识管理.北京:电子工业出版社,2010.

5 汪忠国,吴敏.基于向量空间模型的题库相似度检索方法.计算机系统应用,2010,19(3):214.

6 Kim JS, Nam B, Marsh M, et al. Creating a Robust Desktop Grid using Peer-to-Peer Services.[2012-5-18].ftp://ftp.cs.umd.edu/pub/hpsl/papers/papers-pdf/ngs07.pdf.

7 刘鹏.云计算.北京:电子工业出版社,2010.

8 Fingar P.王灵俊译.云计算:21 世纪的商业平台.北京:电子工业出版社,2009.

9 高晓燕.云计算在图书馆中的应用探究.浙江高校图书情报工作,2010,99:12.

10 樊勇兵,丁圣勇,陈天,汪来富等.解惑云计算.北京:人民邮电出版社,2011.