文章编号:1671 - 6833(2012)05 - 0110 - 04

基于 MapReduce 框架一种文本挖掘算法的设计与实现

朱蔷蔷,张桂芸,刘文龙

(天津师范大学 计算机与信息工程学院,天津 300387)

摘 要:随着文本挖掘在主动信息服务中应用的日益扩展,在文本数据的基础上分析数据的内在特征已经成为目前的研究趋势,本文在 Hadoop 平台上设计并实现了一种文本挖掘算法,该算法利用 MapReduce 框架按照自然语料中相邻词组出现的频数进行降序输出,从而有助于用户挖掘大量数据中各项集之间的联系,实验结果体现了该算法的有效性和良好的加速比.

关键词: Hadoop;MapReduce;相邻词组;降序输出

中图分类号: TP391 **文献标志码**: A doi:10.3969/j.issn.1671 - 6833.2012.05.024

0 引言

随着互联网的大规模普及和社会信息化程度的提高,各种信息机构每天都会产生大量文本数据,从这些文本数据中获取需要的信息成为必然.信息服务机构可以使用文本挖掘技术在已知文本数据的基础上分析数据的内在特征,并以直观的方式将信息模式、数据的关联趋势主动提供给用户,用户可以在此基础上进行趋势分析[1-2].作者在 Hadoop 平台上设计并实现了一种文本挖掘算法,该算法利用 MapReduce 框架在对自然语料进行了词语切分与词性标记后进行处理,并按照相邻词组出现的频数进行降序输出,从而有助于用户挖掘出大量数据中各项集之间的联系.

1 文本挖掘与 MapReduce

文本挖掘是从非结构化的文本文档中提取有趣的、重要的模式和知识,它所针对的文档本身是半结构化或非结构化的,无确定形式并且缺乏机器可理解的语义[3].MapReduce 是一种编程模型,用于大规模数据集的并行运算.当前的软件实现是指定一个 Map(映射)函数,用来把一组键值对 < key, value > 映射成一组新的键值对 < key, value > ,指定 Reduce(化简)函数,用来保证所有映射的键值对中的每一个共享相同的键组.一个 Map/Reduce 作业(Job)通常会把输入的数据集切分为若干独立的数据块,由 Map 任务(task)以完全并行的方式处理它们,框架会对 Map 的输出先进行排序,然后把结果输入给 Reduce 任务.作业的执行流程:作业的提交→Map 任务的分配和执行→Reduce 任务的分配和执行→作业的完成.每个任务的执行过程为:输入的准备 →算法的执行→输出的生成.

2 基于 MapReduce 框架文本挖掘算法的设计

2.1 整体思路

该算法选择基于 Hadoop 平台的 MapReduce 框架,利用 Mapper 类实现词组个数的统计,利用 Reducer 类输出相同词组的频数;Hadoop 中 Reduce 默认按照 key 值进行排序[4],而此时需要对 value 值进行排序,所以将 key 和 value 互换,从而实现按照 key 排序的功能;Hadoop 默认对 IntWritable 按升序排序[5],而我们需要按降序排列,因此实现了一个排序类,并指定这个排序类对输出结果中的 key 进行排序,从而得到按词组频数降序排序的结果.

2.2 思路来源

(1)该数据源数据量相对比较大,而且考虑到后期的扩展性测试,MapRedude 框架无疑是首选.

收稿日期:2012 - 03 - 28;**修订日期**:2012 - 05 - 17

基金项目:国家自然科学基金资助项目(60970060);天津市教委资助项目(20071328);天津市科技支撑计划重点项目(09ZCKFGX00500);天津师大博士基金项目(52LX17)

通信作者:张桂芸(1965 -),女,天津蓟县人,天津师范大学教授,博士后,硕士生导师,主要从事人工智能和数据挖掘研究,E-mail:dyxy1999@126.com

（2）由于输出要求按照词组频数（value）排序，而此功能 MapReduce 框架可自动实现按照 key 值进行排序输出，所以将 key 和 value 互换，从而利用 Hadoop 本身具有的机制实现排序功能.

3　基于 MapReduce 框架文本挖掘算法的实现

3.1　术语解释

为了后续叙述的简便，采用了一些特有术语，具体术语解释见表 1.

表 1　术语解释

Tab.1　The explanation of terms

术语	解释
2 维词组	由两个相邻单词组成的词组（首单词 + 尾单词）
3 维词组	由三个相邻单词组成的词组（上一个 2 维词组 + 尾单词）
首单词	词组中第一个单词
尾单词	词组中最后一个单词
频数	词组出现的次数

3.2　源数据分析

该实验数据采集自新华社 1988 年 1 月份的新闻报道，日期从 19980101 - 19980131. 在实验前对该语料进行了词语切分与词性标记后进行处理[6-7]，具体实验数据如表 2 所示.

表 2　实验数据

Tab.2　The experimental data

输入（示例）：	19980101 - 01 - 001 - 001/m　迈向/v　充满/v　希望/n　的/u　新/a　世纪/n　——/w　一九九八年/t　新年/t　讲话/n　(/w　附/v　图片/n　1/m　张/q　)/w 19980101 - 01 - 001 - 003/m　(/w　一九九七年/t　十二月/t　三十一日/t　)/w …… 19980131 - 04 - 013 - 027/m　才/d　发觉/v　已/d　迷失/v　了/u　来路/n　./w
部分词性标记及其含义：	/m：数词 /v：动词 /n：名词 /v：助词 /a：形容词 /w：标点符号 /t：时间词 /q：量词 /nr：人名 /nt：机构团体

3.3　Mapper 类实现

实现 Map 函数，利用 StringTokenizer 进行字符串的获取，实现词组个数的统计；Map 函数程序流程图如图 1 所示. 具体实现步骤：

（1）首先判断首单词是否为空，如果是，则获取下一个单词，即首单词获得初始化；

（2）进入循环，设置尾单词为下一个单词；

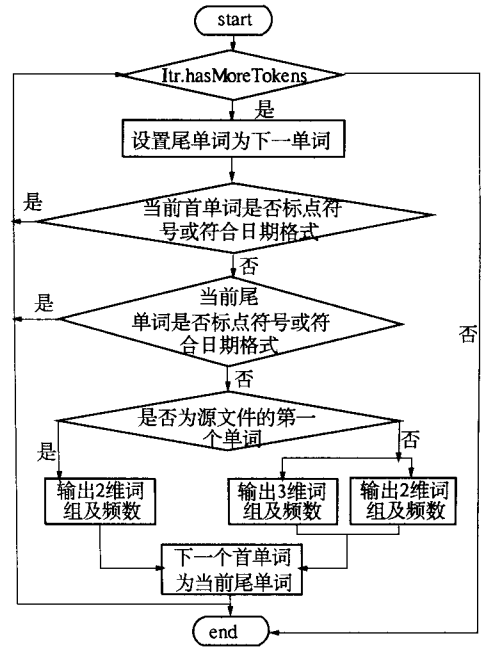

图 1　Map 函数的程序流程图

Fig.1　The flow chart of the Map function

（3）判断当前首单词是否是标点符号，或符合"19980130 - 03 - 004 - 012/m"格式，若是，设下一个首单词为当前尾单词，跳出本次循环，继续下一次循环；

（4）判断当前尾单词是否是标点符号，或符合"19980130 - 03 - 004 - 012/m"格式，若是，设下一个首单词为当前尾单词，跳出本次循环，继续下一次循环；

（5）判断当前首单词是否为源文件的第一个单词，若是，则只需输出 2 维词组，设置频数为 1；若否，则首先输出 3 维词组，设置频数为 1，然后输出 2 维词组，设置频数为 1；

（6）最后设置下一个首单词为当前尾单词.

3.4　Reducer 类实现

实现 Reduce 函数，循环输出相同词组的频数累计和.

3.5　主函数实现

进行作业的配置，与其他实现不同的有：

（1）先将词组的频数统计输出结果写到临时目录中，下一个排序以临时目录为输入目录.

（2）Hadoop 中 Reduce 默认按照 key 值进行排序，而此时需要对 value 值进行排序，所以将 key 和 value 互换，从而利用 Hadoop 本身机制实现排序功能；InverseMapper 类由 hadoop 库提供，作用是实现 map 输出数据对的 key 和 value 交换.

（3）将 Reducer 的个数限定为 1，使得最终输

出到一个结果文件中.

(4)Hadoop 默认对 IntWritable 按升序排序,而需要的是按降序排列.因此实现了一个 IntWritableDecreasingComparator 类,重写了 compare 方法,return -super.compare(a,b),使其返回相反的结果,并指定该类对输出结果中的 key(频数)进行排序.

(5)大致流程如图2所示.

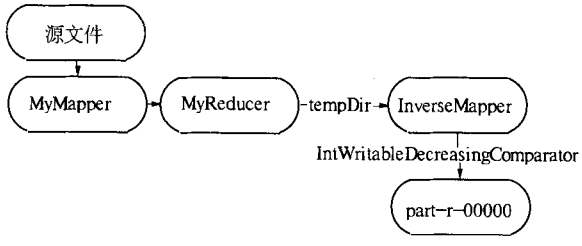

图2 主函数流程图

Fig.2 The flow chart of the main function

4 实验及结果分析

本程序在 Windows 系统下用 Eclipse 进行了开发实现,并用 Cygwin 启动 Hadoop.实验环境如下.Hadoop: hadoop-0.20.2, Eclipse: eclipse-jee-europa-winter-win32, Java: jdk1.6.0_27, Cygwin:模拟 linux 环境[8-9].最后为了测试结果以及性能,分别做了单机处理实验、算法性能实验以及单机与集群对比实验.

4.1 单机处理实验

该实验首先在单机下进行测试,输出结果保存在 HDFSpart-r-00000 文件中,该文件包含了词组以及它们的频数并按降序排列,结果共有1 052 428个词组,表3展示了部分实验结果.

实验结果中词组频数大于500的词组由高到低分别是:的一,新的,这一,这是,的发展,一种,一年,了一,就是.如图3所示,用户可以形象观察出高频率词组,从而挖掘出各个项集之间的关系.

4.2 算法性能实验

为了测试算法的性能,实验中采集了8.41 M,16.3 M,35.2 M,75 M 等4个不同大小的数据集,在实验中,采用加速比作为评价指标.加速比=单个节点运行所需的时间/n 个节点运行时间,实验分别选取1,2,4,6,8个节点参与计算,并统计系统完成任务的时间,实验结果如图4所示,从图4中可以看出:随着集群数目的增加,加速比越来越高,而集群数量较少时加速比提升不明显,主要因为集群间的通信需要消耗一定的时间,而且基于 Hadoop 平台 MapReduce 框架的优势是在大规模集群下处理海量数据;随着数据量的增加,加速比提升越快.实验结果验证了算法良好的加速比.

表3 部分实验结果

Tab.3 The part of experimental results

频数>100	频数:100~80	频数:80~60	频数:60~40	频数:40~20	频数:20~1
的一 845	的艺术 99	领导同志 79	企业和 59	国民经济的 39	这一地区 19
新的 734	中美 99	贯彻落实 79	了很大 59	变得 39	重建家园 19
……	……	……	……	……	……
江泽民 446	改革的 90	的重点 70	今天下午在 50	16日电 30	生活问题 10
不能 442	的文化 90	生产的 70	老同志 50	能不 30	文艺晚会 10
……	……	……	……	……	……
和人民 100	1月22日 80	一块 60	的节日 40	的掌声 20	单位在研制 1
的目标 100	1月19日 80	全国各地 60	新华社发 40	有些 20	从宣传 1

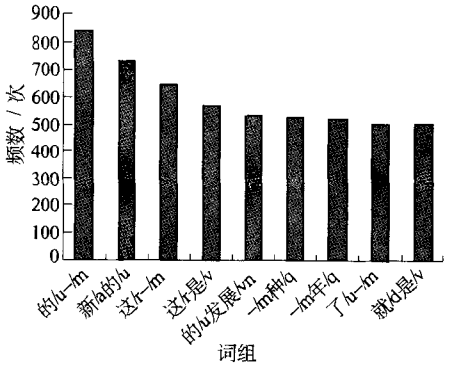

图3 部分实验结果

Fig.3 The part of experimental results

4.3 单机与集群对比实验

为了测试该算法在集群处理时的优势,本实验采取运行时间为指标,统计单机与集群(4台机器)处理8.41 M、16.3 M、35.2 M、75 M 等4个不同大小的数据集的运行时间.从图5中可以看出:当处理8.41 M 较小的数据时,集群运行时间大于单机运行时间,原因在于其运行时间包含了系统

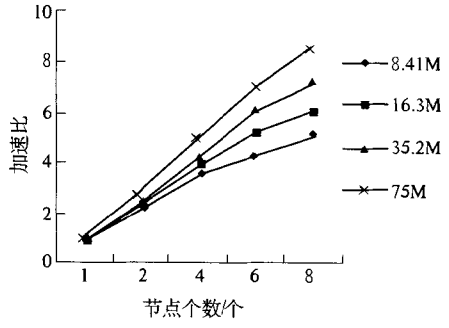

图 4　加速比实验结果

Fig. 4　The results of speedup experiment

初始化时间，以及中间文件、最终文件的生成和传输时间；当处理数据为 16.3 M 时，集群的时效优势开始显露，并行处理平台将数据分割后派给多个 DataNode 进行处理，使集群运行时间小于单机运行时间，此时集群并行计算的时间已经可以抵消通信时的各种消耗；当数据量大于 16.3M 时，随着数据量的增长，集群运行时间波动较小，而单机运行时间则呈线性增长. 实验结果验证了集群下处理大规模数据的优势.

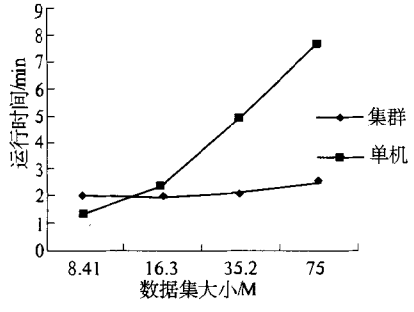

图 5　单机与集群时效对比实验

Fig. 5　The time contrast experiment between single and clusters

5　结论

作者在 Hadoop 平台上利用 MapReduce 框架实现了自然语料中相邻词组出现频率的降序输出，有助于用户挖掘出大量数据中各项集之间的联系，并进行了单机实验、算法性能实验以及单机与集群对比实验，实验结果表明该算法的有效性以及良好的加速比，并进一步验证了集群下处理大规模数据的优势.

参考文献：

[1] Jun Zhu, Ni Lao, Ning Chen, et al. Conditional topical coding: an efficient topic model conditioned on rich features [C] KDD'11, 2011:475 - 482.

[2] JIN Yan, GAO Yang, SHI Ying-huan, et al. P2LSA and P2LSA +: Two paralleled probabilistic latent semantic analysis algorithms based on the mapReduce model [J]. Computer Science, 2011 (6936): 385 - 393.

[3] LI Rui, JU Li, PENG Zhuo, et al. Batch text similarity search with MapReduce [J]. Computer Science, 2011 (6612):412 - 423.

[4] 周戈. 一种基于方向文本频率互信息的文本挖掘算法[J]. 计算机应用研究，2012，29(2)：487 - 489.

[5] 徐东亮，董开坤，李斌，等. 基于文本挖掘的聚类算法研究[J]. 微计算机信息，2011，27(2)：168 - 169.

[6] 刘智勇. 基于云计算的文本挖掘算法研究[D]. 成都：电子科技大学硕士学位论文，2011，3：5 - 19.

[7] 胡军光，刘力，车奇. 基于词性的文本挖掘算法在 IDS 日志中的应用[J]. 计算机与数字工程，2010，38(2)：90 - 93.

[8] 曾理. Hadoop 的重复数据清理模型与研究与实现[D]. 湖南：南华大学，2010：45 - 59.

[9] 程苗，陈华平. 基于 Hadoop 的 Web 日志挖掘[J]. 计算机工程，2011，11：37 - 39.

The Design and Implemention of a Text Mining Algorithm Based on MapReduce Framework

ZHU Qiang-qiang, ZHANG Gui-yun, LIU Wen-long

(College of Computer and Information Engineering, Tianjin Normal University, Tianjin 300387, China)

Abstract: With the expanding application of text mining in active information service, analyzing the inherent characteristics of data based on the text data is becoming a current research trend, this paper designs and implements a text mining algorithm based on the Hadoop platform which outputs the data according to the natural corpora adjacent phrase descending frequency, thus helping the users mine the link between the set in the large quantities of data, In view of the distributed feature of the Hadoop platform, the experimental result shows the efficiency and better speedup.

Key words: Hadoop; MapReduce; adjacent phrase; descending output