第34卷 第15期 计 算 机 工 程 2008年8月
Vol.34 No.15 Computer Engineering August 2008
·软件技术与数据库· 文章编号：1000—3428(2008)15—0072—03 文献标识码：A 中图分类号：TP311.13

在线新事件检测系统中的性能提升策略

王颖颖，张 赟，胡乃静
(上海金融学院信息管理系，上海 201209)

摘 要：现有的关于在线新事件检测(ONED)系统的研究更多地关注如何提高检测的准确率而很少考虑对资源的利用率，使ONED系统在实际应用中存在性能低下的问题。该文分析了传统的事件检测系统存在的性能上的缺点，并在此基础上进行了改进，在基本不降低识别正确率的基础上，通过合理设定技术参数以及对链表索引机制进行预筛选，降低了文档比较过程中的存储和计算开销。实验结果表明，改进的系统提升了检测性能。

关键词：在线新事件检测；话题识别与跟踪；信息检索；预筛选

Performance Improvement Strategy in Online New Event Detection System

WANG Ying-ying, ZHANG Yun, HU Nai-jing
(Department of Information Management, Shanghai Finance University, Shanghai 201209)

【Abstract】 The existing studies focus a lot on the detection accuracy without considering efficiency, which leads a low performance in practical Online New Event Detection(ONED) domain. This paper analyzes the traditional ONED systems about the low performance limitation, and proposes an improved framework. A lot of storage and calculation overhead can be degraded based on the techniques of setting reasonable parameters and pre-filtering index linking tables, without decreasing detection accuracy. Experimental results demonstrate the enhanced performance of the ONED system.

【Key words】 Online New Event Detection(ONED); Topic Detection and Tracking(TDT); Information retrieval; pre-filtering

1 概述

在线新事件检测(Online New Event Detection, ONED)系统是话题识别与跟踪领域的一项重要和基本的子任务[1]，其目标是在一个流文档环境里识别和抓取以前没有讨论过的某一事件的第1篇报道(first story)。这项任务在多个领域有着实际应用，包括信息安全、金融市场数据分析等，其中最主要的应用是在新闻报道领域。如美国政府构建庞大的ONED系统来实时监控新闻、博客、电子邮件等在线信息流以达到反恐的目的。

目前关于ONED系统的研究主要致力于如何提高事件检测准确率，缺乏相应的对资源配置与管理相关的性能调控策略，使得它们很难在实际应用中高效地运行。本文针对这一问题，提出了一个改进的ONED系统架构。通过合理设定和选择存储在内存中的文档，并进一步通过链表索引机制进行预筛选，在基本不降低识别准确率的基础上，大大降低文档比较过程中的存储和计算开销，并验证了实验的准确性。

2 基本的ONED系统

现有的大多数ONED系统都是通过逐一比较新文档和已有文档的相似度来判断一个报道是否是“first story”[2]，为此需要处理大量数目繁多的数据流，效率低下。在实际应用场景中，一个ONED系统需要监控更多的文档源，并需要对实时数据流的“瞬时爆发”特性应付自如，而传统的ONED系统缺乏对这种与资源配置与管理相关的性能调控策略，这也就使得它们很难在实际的应用中高效地运行。

为了更好地理解本文提出的设计思想，首先介绍一个与基本ONED系统相关的术语和概念模型[3]。系统运用了典型的信息检索领域中关于词汇表(vocabulary)、词汇(term)、权重(weight)以及相似度(similarity)等基本概念。目前文档的表示主要采用基于词汇的向量空间模型(Vector Space Model, VSM)。基本思想是以向量来表示文档，向量的成员是词汇的权，词汇越重要，权重越大，其计算公式采用目前流行的TF-IDF公式来定义词频权重和倒转文档频率权重。用相似度来表示文档对之间的相关程度，该值是通过累加所有同时出现在待比较文档中的词汇权重得到的。

首先系统对流中的文档进行分段和预处理，利用分词工具将文档分成词汇的集合，再通过去除停用词等手段将不重要的词汇去除以减少词汇的数目[3]。将每个在文档中出现的词汇进行编号并插入到词汇表中，然后利用TF-IDF公式计算每个词的权重。值得指出的是，由于ONED系统的实时性特点，上述过程一般需要在一个跟测试语料相似的训练语料上进行。

3 改进的ONED系统

原有的ONED系统有2个缺点：(1)随着新的文档流不断涌进，需要被保存的在内存中的文档数目不断增加，最终导致内存溢出[2]；(2)在检测“新事件”的过程中，将一个新的文档与保存在内存中的所有文档进行比较是非常耗时的。这

基金项目：上海市青年科技启明星计划基金资助项目(051430)
作者简介：王颖颖(1973－)，女，讲师、硕士，主研方向：数据库技术；张 赟，讲师、硕士；胡乃静，副教授、博士
收稿日期：2007-09-10 **E-mail：**Wangyingying415@yahoo.com.cn

里必需一个有效的手段，在不降低事件检测成功率的基础上，有效地对存储开销和计算开销进行控制和管理。为此本文提出了一个改进的 ONED 系统架构，如图 1 所示。

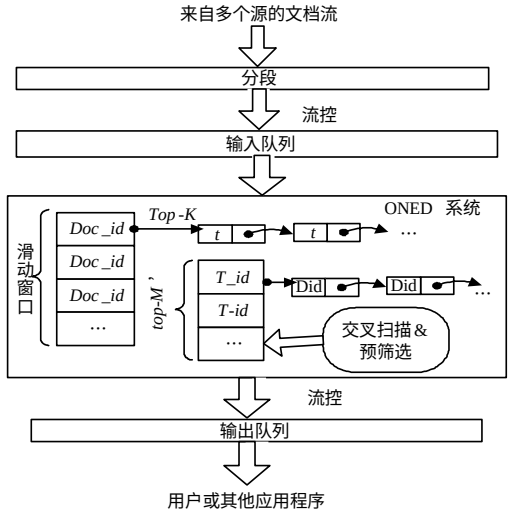

图 1　改进的 ONED 系统框架

3.1　技术参数的设定

由图 1 可以看出，为了防止文档流过快涌入导致系统来不及处理，以及系统输出新事件的速度过快用户来不及接收，使用队列进行流控[3]。该机制的具体实现本文不作重点讨论，侧重点放在 ONED 系统内部的设计与实现上。为此设定了一些技术参数[1,4]。

(1)定义一个滑动窗口 *W*，仅过去 *W* 天的文档被保存在内存中。因为新闻报道领域有着非常明显的时段性特征，即关于一个事件的报道会持续在某一个连续的时间段，在这个时间段里一个文档不太会提及那些相对较早发生的事件。一旦过时文档不在窗口的有效范围里，则从内存中丢弃。合理地定义窗口的值可以缓解存储开销，并降低计算开销进而提高整个系统的吞吐率。*W* 的缺省值为 29。

(2)仅仅在内存中保存关于某一个事件的第 1 篇报道。因为报道了相同事件的文档之间有某种程度的相似性，所以无须将一个新的文档去与所有与该事件相关的文档逐一比较。

(3)限制参与相似度计算的词汇个数 *K* 和交叉扫描词汇个数 *M*。一个文档中所有的词汇可以依据它们的 *TF*×*IDF* 值降序排列，一般来说，该值越大的词汇对文档而言越重要[1]，因而挑选贡献了大多数相似度的 *Top-K* 个词汇参加计算，可以在保证识别准确率的基础上有效地降低计算复杂度，提高系统的效率。*K* 的缺省值为 90。而交叉扫描词汇个数 *M* 通常设定为 10 左右，在下面的预筛选策略中将详细讨论该值。

3.2　预筛选策略

下面讨论如何利用上面设定的技术参数对保存的文档进行预筛选。一般，如果 2 个文档 D_1 和 D_2 是对同一事件的报道，那么它们的前 *Top-M* 个词汇一定会有交叉，反之，如果它们没有交叉，那么认为 D_2 相对于 D_1 是一个“新事件”。这是由新闻报道领域自然语言的语义特点决定的，寻求“第 1 个新事件”的过程可以充分利用这一特性进行第 1 层筛选。在此基础之上(即 *Top-M* 个词汇有交叉的情形)，进行旨在降低求相似值过程中的计算开销的第 2 层筛选，过滤掉一部分不需要参加比较计算的文档，从而减少了大量的计算开销。

为此，用 *Saved_Docu* 表示保存文档，用 *Doc_id* 表示唯一标识文档的 ID 号，它是由文档到达的时间戳生成的。对于 *Saved_Docu* 中的每个节点 D_S，用指针连接一个该文档所包含前 *K* 个词汇 *term* 的链表。每个 *term* 节点由 2 个域构成，*Term-id* 和 *tf*，其中 *Term_id* 是指该 *term* 在词汇表中的标识号，tf 表示该 *term* 在 D_S 中出现的频率。

特别地，定义当前 *Saved_Docu* 中每个 D_S 的前 *M* 个词汇的并集 *TUnion*，一般来说，*TUnion* 中词汇的个数 *M'*≥*M*(但由于 *W* 的大小确定，因此 *M'*相对于整个词汇表是一个可以接受的不太大的值)。对于 *TUnion* 中的每一个词汇，记录其 *Term_id*, *Term_name* 和 *df*，分别表示该词汇在词汇表中的标识号和名称，*df* 则表示流中包含该词汇的文档个数。类似地，*TUnion* 的每一项指向一个 *Doc_id* 链表，标识所有在 *Saved_Docu* 中出现过该词汇的文档。

概括地说，系统的工作首先是通过 *M* 参数进行词汇交叉扫描，再利用 *K* 进行相似度的计算，最后通过 *T* 进行是否相似的判断。具体流程如下：

(1)当一个新的文档 D_{new} 到达时，词汇交叉扫描服务启动，将 D_{new} 的 *top*-M 词汇去匹配 *TUnion* 的 *Top-M'*词汇，(前面提到，*TUnion* 的大小 *M'*不是很大，它是保存在内存中的所有 *W* 个文档的 *Top-M* 词汇的并集)，由此找出所有保存在内存中的跟 D_{new} 提及了相同事件的文档的集合 *Flitered_Docu*，显然，*Flitered_Docu* 是 *Saved_Docu* 的一个有限的子集。

(2)在相似度比较计算中，只要将 D_{new} 和筛选后的 *Flitered_Docu* 的文档 *Top-K* 个词汇依次进行比较，计算相似度就可以了。

(3)如果计算出的相似度值小于预定义的相似度的阈值 *T*，则认为 D_{new} 是一个第 1 次报道某个事件的文档，也就是说被识别成为一个“first story”。反之，则认为 D_{new} 是一个与过去某些报道有着相当相似程度(由阈值 *T* 定义)的旧报道，被 ONED 系统丢弃。

3.3　算法描述及分析

```
输入：在线流环境中的一个新到达的文档 D_new
      常数 W,T,K //用于预定义常数
      常数 M //用来定义贡献最大相似度的 Top-M 个词汇
输出：fs_Flag 置为 True 并保存 D_new；
生成 D_new 的 Top-M 词汇集 TMSet；
比较 TMSet 和 TUnion；
If 无交叉 then //表明 D_new 是一个新事件
  fs_Flag=True;
  Save (D_new);
else
for TUnion 中的每个词汇 t' do
    if t' in TMSet then
        Add(t'->Doc_id, Flitered_Docu)
end for
for Flitered_Docu 中的每个 Doc_id do
if SimVal(Doc_id，D_new)< T then
                //表明 D_new 是一个新事件
fs_Flag=True;
                Save (D_new);
                exit loop
  end for
```

上述算法的主要计算开销是相似度的计算 *SimVal*()，其算法的复杂度约为 $O(n \times K)$，其中，n 为筛选后的 *Flitered_Docu* 集合大小；K为参与相似度计算的词汇个数；

虽然在词汇交叉扫描和列表合并方面也有一定开销，但可以忽略不计。相似度计算过程位于一个循环体内，显然通过减少循环次数有效地降低整个新事件识别系统的算法复杂度。

4 实验

实验使用国内较有影响的新浪、雅虎、中国新闻网、新华社网站为测试语料，分别抽取 2006 年 5 月～2006 年 12 月中不同时间段的新闻进行分析。这些语料库的大小分别约为 40 MB, 250 MB, 20 MB, 164 MB。本实验以 TDT 的 5 种中文语料作为训练语料，主要考察改进后的 ONED 检测算法是否可以在不降低识别准确率的基础上大幅度地提高系统的工作性能，实验中的技术参数取缺省值。实验还测试了 M 在取值范围变化时试验数据的变化。

(1)实验采用在 TDT 评测领域常用的归一化识别代价 C_{Det} 作为评价指标，该值越小说明识别效果越好。归一化识别代价由系统的识别漏报率和误报率计算得到，文档的漏报率是指系统没有识别出来的关于某话题的新闻报道的数目与语料库中描述该话题的新闻报道的总数之比，而误报率是指对某一话题来说判断错误的新闻报道数目与语料库中所有没有描述该话题的新闻报道的总数之比。

(2)实验测试的结果如表 1 所示，第 1 个数值为识别代价成本 C_{Det}，第 2 个数值为耗费的时间。

表 1 实验语料相关统计数据

M		新浪(40 MB)	新华社(250 MB)	中国新闻网(20 MB)	雅虎(164 MB)
10	原系统	23 128, 0.721 1	122 120, 0.7081	9 865, 0.711 9	68128, 0.718 1
	改进后	78, 0.732 4	341, 0.718 2	42, 0.723 0	195, 0.743 2
15	原系统	25 899, 0.801 2	124 570, 0.837 0	12 018, 0.820 1	72 102, 0.813 2
	改进后	90, 0.809 0	366, 0.841 1	68, 0.829 9	301, 0.837 8
20	原系统	36 031, 0.843 2	150 891, 0.890 1	15 292, 0.834 3	128 010, 0.853 2
	改进后	182, 0.844 2	569, 0.887 1	132, 0.899 0	441, 0.887 1

从表 1 可以看出，在采用了改进后的系统中，计算时间比原有系统缩短了几个数量级，以新浪网站新闻的预料测试结果为例，原来的计算时间为 23 128 s，相当于 6 h，而改进后的系统的计算时间缩短为 78 s，它们的归一化代价成本分别为 0.721 1 和 0.732 4，可以看出新系统的识别成功率略有下降，但是幅度很小，几乎可以忽略不计。可见，这种计算开销的大幅度降低并不以降低识别成功率为代价。另外，不同的 M 值对计算时间和归一化成本值也有影响，当 M 取值为 10 左右时效果最好。这是因为 M 越大，参与计算的旧文档就越多，计算时间也越长。

5 结束语

ONED 系统一直侧重于研究如何提高系统的查准率和降低漏检率，而很少考虑系统性能的问题。本文提出了一个改进的 ONED 系统框架，可以在不降低识别准确率的基础上，通过合理设定和选择存储在内存中的文档，用链表索引机制进行预筛选，使得文档比较过程中的存储和计算开销大大降低。从而提升系统的整体性能。最后通过实验证明了结论的正确性。下一步将围绕如何进一步提高 ONED 系统性能和资源利用效率进行研究，如系统如何利用 ONED 系统的输出结果进行文档分级，以及词汇表中同义词的语义分析与处理等。

参考文献

[1] Allan J, Papka, R Lavrenko V. On-line New Event Detection and Tracking[C]//Proc. of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. [S. l.]: ACM Press, 1998: 37-45.

[2] Allan J, Lavrenko V, Jin H. First Story Detection in TDT is Hard[C]//Proc. of the 9th International Conference on Information and Knowledge Management. New York, USA: ACM Press, 2000: 3-5.

[3] Braun R, Kaneshiro R. Exploiting Topic Pragmatics for New Event Detection in TDT-2004[C]//Proc. of Topic Detection and Tracking Workshop. [S. l.]: ACM Press, 2004.

[4] Luo Gang, Tang Chunqiang, Philip S Y. Resource-adaptive Real-time New Event Detection[C]//Proc. of ACM SIGMOD-PODS'07. Beijing, China: ACM Press, 2007: 3-4.

（上接第 71 页）

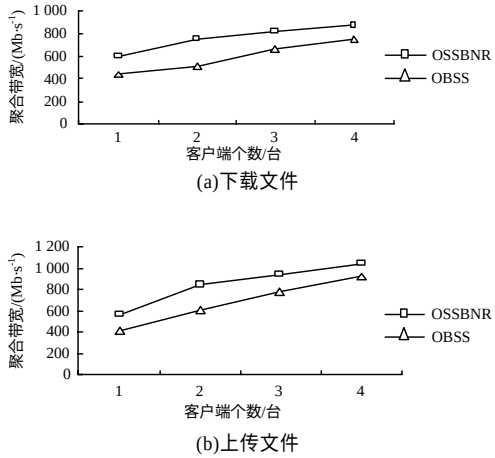

图 3 OSSBNR 中 2 个 SO 与 OBSS 中 2 个 OSD 的性能比较

由测试结果可以看出，OSSBNR 的吞吐率曲线明显高于 OBSS，这是 OSSBNR 中 SO 的 2 个 Net-RAID 直接连网的结构特点和对象控制与数据传输分离的处理方式所决定的。可见，随着系统容量的扩充和客户端的增加，OSSBNR 的性能更佳。

5 结束语

本文分析并比较 OSSBNR 与 OBSS 的体系结构和性能，测试结果表明，OSSBNR 具有 OBSS 的优点且充分利用了 Net-RAID 的特点，实现了元数据流、控制流、数据流的分离，在系统容量和系统性能的同步扩展方面表现更佳。

参考文献

[1] Mesnier M, Ganger G R, Riedel E. Object-based Storage[J]. IEEE Communications Magazine, 2003, 41(8): 84-90.

[2] Garth A G, David F N. NASD Scalable Storage Systems[C]//Proc. of the USENIX Conference on Linux Workshop. Monterey, USA: USENIX Press, 1999.

[3] Weber R O. ANSI/INCITS400－2004 SCSI Object-based Storage Device Commands[S]. 2004.

[4] Braam P J. The Lustre Storage Architecture[EB/OL]. (2002-03-22). http://www.lustre.org/docs/lustre.pdf.

[5] Panasas White Paper. Shared Storage Cluster Computing[EB/OL]. (2003-10-09). http://www.panasas.com.

[6] Brandt S A. Efficient Metadata Management in Large Distributed Storage Systems[C]//Proc. of the 20th IEEE Conference on Mass Storage Systems and Technologies. California, USA: IEEE Press, 2003.

[7] 王 芳. 网络磁盘阵列系统的研究[D]. 武汉: 华中科技大学, 2001.