基于 AC-Trie 的在线社交网络文本流热点短语挖掘

黄九鸣,吴泉源,张圣栋,贾 焰,刘 东,周 斌
(国防科学技术大学计算机学院,湖南长沙 410073)

摘 要: 在线社交网络文本流中的热点短语能反映文本流中隐含的热点话题和突发事件.本文提出了一种无需分词并能支持多种热度度量函数的热点短语挖掘技术.首先用文本流的某个典型时段采样得到候选短语,构建 AC-Trie 前缀树.然后,基于该前缀树,单遍扫描后续的文本流,将候选短语的历史出现频率记录在 Trie 相应节点上,从而支持多种基于历史频率的热度计算方法.此外,为及时发现新的热点短语并减少 AC-Trie 的构建次数,本文通过分析 Trie 树各节点上的遗漏短语频率,动态确定候选短语的更新时机.新浪微博数据集上的实验验证了本文方法的有效性(准确率达 89%)和高效性(时空开销仅为基准算法的 2%).

关键词: 文本流;热点短语;AC-Trie;文本挖掘;在线社交网络
中图分类号: TP391 **文献标识码:** A **文章编号:** 0372-2112 (2016)10-2466-05
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2016.10.026

Mining Hot Phrases on Social Network Text Streams Based on AC-Trie

HUANG Jiu-ming, WU Quan-yuan, ZHANG Sheng-dong, JIA Yan, LIU Dong, ZHOU Bin
(*School of Computer, National University of Defense Technology, Changsha, Hunan* 410073, *China*)

Abstract: The hot phrases in the social network text streams can reflect the hidden hot topics and sudden events. This paper proposes a hot phrase mining technology which can support various hot degree measures without word segmentation. We first construct an AC-Trie using the candidate phrases gathered from text streams. Based on such AC-Trie, we record the historical occurrence frequency of phrases on the Trie by scanning the following streams in single-pass. Furthermore, the AC-Trie needs to be reconstructed using the new samples in the text stream because of the evolution of hot phrases. Thus, we start the reconstruction dynamically according to estimating the occurrence frequency of the missed phrases. The experiments on the Sina micro-blog show that our approach is effective (precision of 89%) and efficient (overhead is 2% of naïve approach).

Key words: text stream; hot phrase; AC-Trie; text mining; micro-blog

1 引言

微博、即时通信、BBS 等在线社交网络应用的用户通过文本消息来表达和传递自己的思想.这些带有时间属性的文本消息构成了网络文本流或网络文本流.挖掘网络文本流中被广泛讨论和关注的热点短语,可有效地应用于舆情分析、股市预测以及商业智能等领域.

本文称文本流中能体现当前热门话题或突发事件的短语为热点短语."热点"短语不同于"频繁"出现的短语.图 1 展示了 2015 年 4 月 20 日至 5 月 19 日期间国内各主要微博平台(新浪微博、腾讯微博、搜狐微博)上含有"尼泊尔地震"、"谢谢"、"中央巡视组"、"复仇者联盟 2"四个短语的微博数量.可见,日常用语"谢谢"的出现频率较高且相对稳定,每天的微博数在 10 万左右;突发事件"尼泊尔地震"在 4 月 25 日数量剧增,日发微博量高达 12 万条,但随后迅速降落;持续热议话题"中央巡视组"每天的微博量不多,但月底由于中央巡视组公布名单会突增;新上线的影片"复仇者联盟 2"从早期便有相关讨论,并随着影片上映日期接近微博数量持续增长.由此可见,并非出现频率高的短语都是热门短语.另外,从短语的出现频率变化情况能够发现不同类型的热门短语,根据需求采用不同的出现频率统计模型可有效发现特定热门短语.

收稿日期:2015-02-15;修回日期:2015-08-14;责任编辑:李勇锋
基金项目:国家 973 重点基础研究发展计划(No. 2013CB329601);国家自然科学基金(No. 61502517)

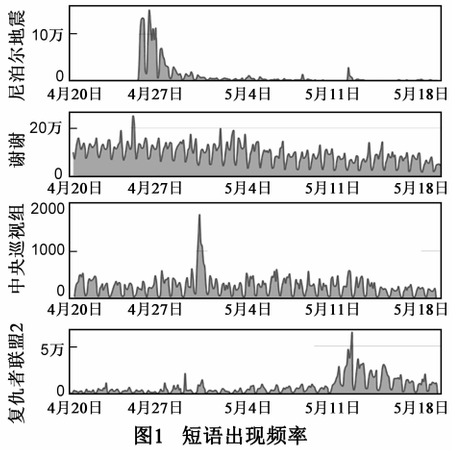

图1　短语出现频率

已有的热点词挖掘技术面临以下挑战:①在高速到达的海量网络文本流上进行统计挖掘,计算和存储开销巨大;②热点短语的鉴别与用户需求相关,单一的统计方法适应性差. 针对上述问题,本文提出一种网络热点短语挖掘技术 AC-Hot. 该技术与最大频繁项挖掘技术[1]相比,具有以下显著特点:①只需单遍扫描文本流. ②作为一种热点短语挖掘框架,通过详细记录了候选短语的历史状态,可支持各种自定义的热度度量方法. ③内存空间占用量可控,通过从文本流中采样构建候选短语集合,有效控制内存占用量. ④无需预先分词,可自动发现热点新词和短语.

2　相关研究

有一些研究与本文方法间接相关,它们致力于挖掘突发词语或检测给定词语的爆发时间点. 例如,文献[2]用一个突发属性集合来表示一个突发事件. 文献[3]针对博客和论坛文本流的特点对 Kleinberg 的算法[4]进行扩展. 这类方法只考察单个词语,没有考虑词语的组合,无法发现由多个词语组成的热点短语. 给定文档集中的频繁短语挖掘有较多研究成果,如文献[5,6]. 但是,这类方法无法处理流数据. 数据流中挖掘最大频繁项集方面有三类方法,分别是滑动窗口模型[7],时间消逝模型[8]和界标模型[9]. 但它们都是在特定前提下给出了频繁项集的定义,不保存各个(项集)子串的所有历史状态,因此无法支持多种热度度量模型.

社会媒体热点话题检测与突发事件检测方面的研究[10~12]与本文目的相似,旨在发现能代表热点话题或突发事件的文档集或关键词. 文档集方面,主要采用基于文本聚类的方法,通过比较文档的相似性并采用 Single-Pass 或 K-Means 等聚类方法实现聚类[11],进而通过各类别的文档数量和点击率等指标计算话题热度. 这类方法的计算量巨大,可理解性差. 另一方面,基于关键词的热点话题或突发事件检测方面,比较有代表性的是基于 LDA 模型的各种改良方法[12]. 这类方法的计算量同样比较大,并且需事先对文档进行分词,无法自动发现网络新词.

3　问题定义

本文将微博、BBS、即时通信等在线社交网络应用的文本消息数据抽象为文本流,简称文本流. 称字母表中的一系列字符组成的字符串为一个短语. 如果短语 q 是消息 m 的子串,称消息 m 包含短语 q,又称短语 q 在 m 中出现,记为 $m \supseteq q$. 一个文本流截至时间 t 时包含短语 q 的消息条数,称为截至 t 时 q 在该文本流的出现次数. 短语的出现频率指其在一定时间窗口内该短语出现次数与消息条数的比值,用函数 θ 表示,如定义 1 所示.

定义 1(出现频率)　令 q 为一短语,S 为文本流,频率统计时间窗口为 Δt,函数 $\tau(m)$ 表示消息 m 的产生时刻,则 t 时 q 在 S 中的出现频率为:

$$\theta(q,t,S)=\frac{|\{m|m\supseteq q,m\in S,t-\Delta t<\tau(m)\leqslant t\}|}{|\{m|m\in S,t-\Delta t<\tau(m)\leqslant t\}|} \tag{1}$$

热点短语挖掘任务为在指定时间点,查找出热度值排名在前 k 位的短语,如定义 2 所示.

定义 2(热点短语挖掘)　给定的文本流 S,时刻 t,用 Q 表示所有短语的集合,d 为用户指定的热度度量函数,称热度值排名前 k 位的短语为 t 时刻文本流 S 的 TopK 热门短语,如式(2)所示:

$$Hot(t,k,S)=\{q_x|q_x\in Q,1\leqslant x\leqslant k,\forall_{q\in Q}d(q_x,t)\geqslant d(q,t)\} \tag{2}$$

4　基于 Trie 的朴素算法

根据定义 2,热点短语挖掘直观的解决方案是在内存中保存所有短语,将短语出现次数的变化情况保存在历史频率表中. 为压缩数据存储,本文的朴素方法基于 Trie 实现. 在 Trie 树的每个节点上增加一个历史频率表. 历史频率表中的元素为时间与出现频率组成的二元组. 该方法分为两个步骤:首先将短语及其出现频率保存在 Trie 树上,然后在 Trie 树上查找最热门的 k 个短语.

第一个步骤的过程如下:(1)为文本流创建一个 Trie 树;(2)当新消息 m 到达时,将 m 包含的所有短语放在集合 E 中;(3)对每个 $q\in E$,在 Trie 树上查找是否存在 q,如果不存在则将 q 加入 Trie 树;(4)对每个 $q\in E$,设其历史频率表表尾的元素为$\langle t,x\rangle$,若 $\tau(m)<t$ 则更新表尾的值为$\langle t,x+1\rangle$,否则更新表尾为$\langle t,x/|S[t-\Delta t,t]|\rangle$,并新增二元组$\langle(\tau(m)/\Delta t+1)\times\Delta t,1\rangle$. 其中,$S[t-\Delta t,t]$表示文本流中时间期间$[t-\Delta t,t)$

内的消息集合.

第二个步骤中,当用户需要获取文本流的热点短语时,遍历 Trie 树的所有短语并用指定的热度公式来计算短语的热度,然后,按热度值对所有短语进行排序,挑选出最热的 k 个短语.

这个朴素算法的缺点是时空开销巨大. 对于一个文本流,设消息集合为 S,短语集合为 E,第一个步骤的时间复杂度为 $O(\sum m \in S2|m|)$,内存存储的开销是 $O(\sum q \in E|q|)$. 第二个步骤每次查找最热的 k 个短语的时间复杂度为 $O(|E|\log(|E|))$.

5 基于 AC-Trie 的热点短语挖掘技术 AC-Hot

由于 AC-Trie 只需单遍扫描文本流便可同时匹配出多个模式串,本节提出基于 AC-Trie 的热点短语挖掘框架 AC-Hot. 只要能及时地从文本流中截取一个片段作为样本,将样本消息中的所有短语加入 AC-Trie 中进行监视,便可高效发现新出现的热点短语. 因此,AC-Hot 是短语采样和文本流扫描监视两个状态交替运行的过程. 由于 AC-Trie 的构建开销巨大,因此为提高运行效率应尽可能减少短语采样次数. 同时,为保证及时发现新热点短语,应动态确定短语采样时机. AC-Hot 通过估计扫描过程遗漏掉热点短语的概率,动态确定短语采样的时机.

定义 3(遗漏短语) 设文本流 S,短语出现频率统计处于扫描阶段,监视 S 的 AC-Trie 树为 T,有短语 $q \notin T \wedge m \supseteq q$,$m$ 为 S 中新产生的消息,则在该扫描阶段称 q 为遗漏短语.

出现频率统计由扫描状态转入采样状态的时机,将根据遗漏短语是热点短语的可能性大小来动态确定. 为估计遗漏短语是热点短语的可能性大小,我们用遗漏短语的频率值(简称遗漏频率)来估计遗漏短语是热点短语的概率. 遗漏频率记录在 AC-Trie 遗漏短语的父节点上.

定义 4(遗漏频率) 设统计时间窗口为 Δt,时间段$[t-\Delta t, t)$内文本流 S 中有遗漏短语 $q_1, q_2, \cdots, q_n$ 在 AC-Trie 树上的最长前缀都为 q,则 q 对应节点 $v(q)$ 在 t 时的遗漏频率为:

$$miss(v(q)) = \sum_{k=1}^{n} \theta(q_k, t, S) \tag{3}$$

记录在每个节点上的遗漏频率,是以节点对应短语为前缀的所有遗漏短语的出现频率之和,不能直接用于热度计算,应根据遗漏频率估算出每条遗漏短语的出现频率范围. 由于扫描状态下没有为新短语新增子节点,因此各节点应新增的子节点数量,等于以节点短语为前缀的遗漏短语个数. 给定一个文本流 S,对于任意短语(字符串)"$c1c2\cdots ck$"($k \geq 1$),相应后继字符集合 $C(t) = \{c|\text{“}c1c2\cdots ckc\text{”} \subseteq m, \tau(m) \leq t, m \in S\}$,则集合 $C(t)$的大小随 t 增长递增,但增长速度逐渐变慢. 本文假设短语后继字符的数量关于时间呈指数分布. 在 AC-Trie 树上,对于任一节点(短语),其指向子节点的边上的字符即为该节点的后继字符. 后继字符数量关于时间的分布情况,等价于潜在子节点数量关于时间的分布情况. 因此,潜在子节点数量关于时间的分布函数如下式所示:

$$f(x,t) = \alpha_x \ln(\beta_x + \frac{t}{\Delta t}) \tag{4}$$

其中,x 为一节点,t 为时间,α_x 和 β_x 为待定参数. 为估计各个节点的 α_x 和 β_x 参数,首先记录每个节点在采样阶段的每个统计时间窗口内新增子节点的数量,再用最小二乘法进行估计.

TopK 查找过程与短语出现频率监视过程并行运行,基于 AC-Trie 中各候选短语的历史频率表,用具体的热度计算公式计算并查找出热度排名在前 k 位的短语,同时根据各节点的遗漏历史表估计遗漏短语出现频率的取值,以判断是否需要重新进行短语采样. TopK 查找过程首先采用自底向上宽度优先遍历 Trie 树的策略,将子节点上历史频率表的值汇总到父节点和 fail 指针指向的节点(后缀)上. 对每个节点,计算其热度,并估算以该节点为前缀的各遗漏短语中的最大热度值,然后将这两个值分别用两个格式为 < 热度,节点,类型 > 的三元组表示. 执行完上述步骤后,检查遗漏历史表,若相应节点下的遗漏短语中可能含有热度在前 k 位的短语,文本流中可能有 AC-Trie 上不存在的热点短语,监视状态转入短语采样状态.

6 实验验证

为验证本文方法的有效性,我们从新浪微博、腾讯微博和 Twitter 三个社交网络平台上采集 2015 年 5 月 1 日至 5 月 30 日一个月内关于"四川"的 2661 万条微博,构建实验数据集. 本实验以"自动发现每天舆情热点"为需求背景,设置 TopK 查找的运行周期(简称 TopK 周期)为 1 天. 由于人的关注范围有限,TopK 查找输出的热点短语数目 k 设为 20.

由于本文方法 AC-Hot 与基于话题模型的热点话题发现在表现形式、计算性能上存在显著差异(见本文相关研究),因此本实验不同这类方法进行对比. 另一方面,已有的基于关键词的方法,都需要事先分词,不能发现新词,更不能灵活支持多种统计方法,难以同 AC-Hot 进行实验对比. 为此,本实验以基于 Trie 的朴素算法为基准算法,对比分析 AC-Hot 的准确性和处理速度.

我们在数据集上充分测试了 AC-Hot. 各个 TopK 周期上的准确率如图 2 所示,平均准确率为 0.89,总体上

比较稳定. 表 1 列出了 AC-Hot 在数据集上运行时,短语采样被触发的 TopK 周期编号. 可见,几处波动较大的地方,恰为 AC-Hot 判断需重新进行短语采样的时刻. 并且,重新采样完毕后,准确率马上又回到较高的水平.

表 1　短语采样时刻

TopK 周期	4	11	15	18	21	26	27
准确率	0.7	1	0.85	0.5	0.9	0.7	0.6

图 3(*a*)与图 3(*b*)分别展示了 AC-Hot 与基准算法在数据集上的时间开销对比和内存空间开销对比. 两

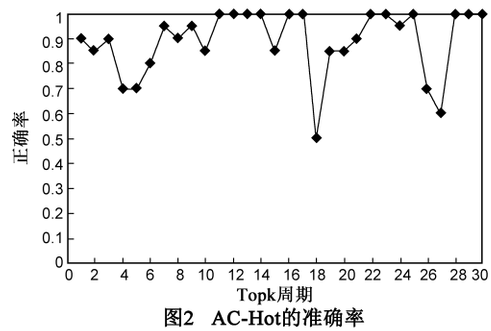

图2　AC-Hot的准确率

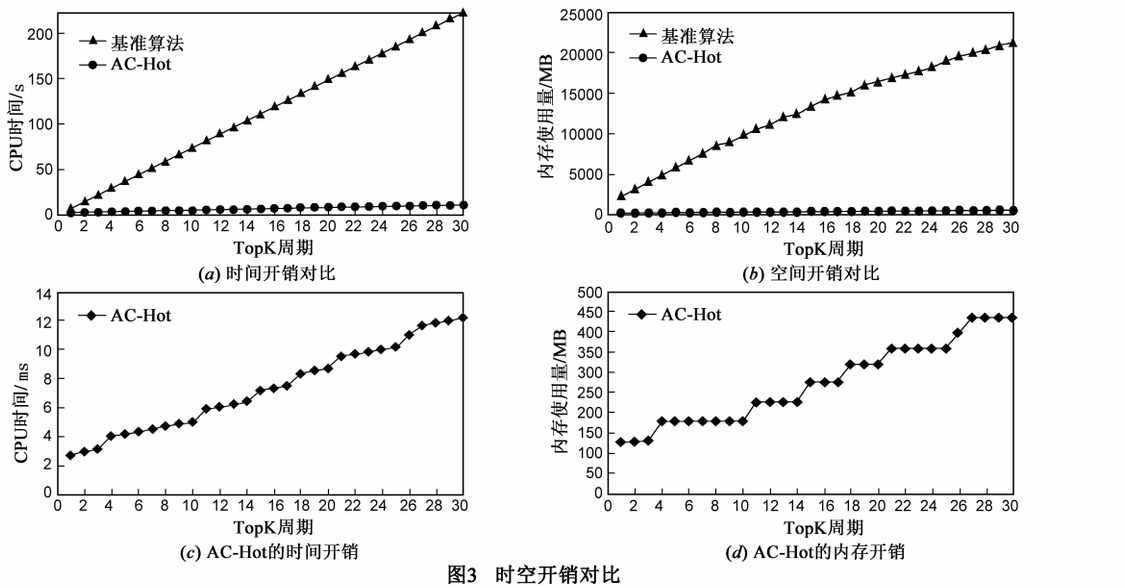

图3　时空开销对比

张图的横轴均为 TopK 周期编号,纵轴分别为所需 CPU 时间与内存使用量. 可见,AC-Hot 的时间开销和内存开销都远小于基准算法(朴素算法).

7　结束语

文本流中的热点短语能反映文本流中隐含的热点话题和突发事件. 本文分析了热点短语的形成规律,针对热度度量方法多样、文本消息数量巨大等挑战,提出了具有极高性能的近似方法 AC-Hot. 该方法能支持多种热度度量方法,平均准确率达 89%,时空开销仅为基准算法的 2%.

参考文献

[1] Calders T, Dexters N, Goethals B. Mining frequent itemsets in a stream[A]. Seventh IEEE International Conference on Data Mining[C]. Omaha, Nebraska: IEEE, 2007. 83 – 92.

[2] Yuan Z, Jia Y, Yang S. Online burst detection over high speed short text streams[A]. Computational Science-ICCS 2007[C]. Heidelberg, Berlin: Springer, 2007. 717 – 725.

[3] Fujiki T, Nanno T, Suzuki Y, Okumura M. Identification of bursts in a document stream[A]. First International Workshop on Knowledge Discovery in Data Streams (in conjunction with ECML/PKDD 2004)[C]. Pisa, Italy, 2004. 55 – 64.

[4] Kleinberg J. Bursty and hierarchical structure in streams [J]. Data Mining and Knowledge Discovery, 2003, 7(4): 373 – 397.

[5] Ahonen-Myka H. Discovery of frequent word sequences in text[A]. Pattern Detection and Discovery[M]. Berlin Heidelberg: Springer, 2002. 180 – 189.

[6] Han J, Pei J, Yin Y. Mining frequent patterns without candidate generation[A]. ACM SIGMOD Record[C]. Dallas, Texas: ACM, 2000. 29(2): 1 – 12.

[7] Wong R C W, Fu A W C. Mining top-K frequent itemsets from data streams[J]. Data Mining and Knowledge Discovery, 2006, 13(2): 193 – 217.

[8] Lee D, Lee W. Finding maximal frequent itemsets over online data streams adaptively[A]. Fifth IEEE International

Conference on Data Mining[C]. Houston, Texas: IEEE, 2005. 8.

[9] Yu J X, Chong Z, Lu H, et al. False positive or false negative: mining frequent itemsets from high speed transactional data streams[A]. Proceedings of the Thirtieth International Conference on Very Large Data Bases (VLDB Endowment)[C]. Toronto, 2004. Volume 30: 204 – 215.

[10] Thanh Lam H, Calders T. Mining top-k frequent items in a data stream with flexible sliding windows[A]. Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining[C]. Washington, DC: ACM, 2010. 283 – 292.

[11] Zhou G, Zou H C, Xiong X B, et al. MB-singlepass: microblog topic detection based on combined similarity[J]. Computer Science, 2012, 39(10): 198 – 202.

[12] Liu G, Xu X, Zhu Y, et al. An improved latent dirichlet allocation model for hot topic extraction[A]. IEEE Fourth International Conference on Big Data and Cloud Computing (BdCloud)[C]. Sydney: IEEE, 2014. 470 – 476.

作者简介

黄九鸣 男, 1981 年生于福建安溪. 博士、中国人民解放军国防科学技术大学助理研究员. 研究方向为 Web 挖掘、大数据、分布式计算和社交网络分析.

E-mail: jiuming. huang@ qq. com

吴泉源 男, 1942 年生于上海. 中国人民解放军国防科学技术大学教授、博士生导师. 研究方向为人工智能和分布式计算.