第 49 卷 第 4 期
2013 年 7 月

南京大学学报(自然科学)
JOURNAL OF NANJING UNIVERSITY
(NATURAL SCIENCES)

Vol. 49, No. 4
July, 2013

基于重用检测的微博垃圾用户过滤算法*

赵 斌**,吉根林,曲维光,顾彦慧
(南京师范大学计算机科学与技术学院,南京,210023)

摘 要: 针对微博中的反垃圾处理问题,本文提出了基于重用检测模型的垃圾用户检测算法,该方法综合考虑了消息序列中文本相关性和时间相关性,对垃圾用户的发布行为进行建模.按照文本粒度不同,基于重用检测模型的检测算法分为语句级检测(SRD)和词项级检测(TRD).SRD 算法侧重于用户行为方式,而 TRD 算法侧重于垃圾消息的主题.基于真实数据集的实验表明,SRD 算法在整体性能上优于 TRD 算法,但 TRD 算法具有更高的运行效率,并且检测针对性强,可发现指定类型的垃圾用户.最后,本文运用重用检测算法在垃圾用户群体检测方面做了初步尝试,实验表明基于转发关系的重用检测算法可以发现真实有效的垃圾群体用户.

关键词: 垃圾消息,微 博,重用检测

Detecting microblog spammers based on reuse detection

Zhao Bin, Ji Gen-Lin, Qu Wei-Guang, Gu Yan-Hui
(School of Computer Science and Technology, Nanjing Normal University, Nanjing, 210023, China)

Abstract: Tremendous increase of spam has become a serious problem. In this paper, we aim to detect microblog spammers by means of retweeting relationship. We introduce a new reuse detection model, which simultaneously incorporates text content and temporal information, to rate the intensity of spamming behaviours. We then present two spam detection algorithms based on such model. One is sentence-level detection algorithm, the other is term-level one. The sentence-level detection algorithm prefers the behaviour pattern of spammers and ignores the topic of spam messages. The term-level detection algorithm focuses the topic of spam messages and compensates for lack of sentence-level one. Finally, we evaluate our approaches on a real dataset collected from Sina microblog, the largest microblog in China. Extensive experiments show the effectiveness and efficiency of our algorithms.

Key words: spam, microblog, reuse detection

* 基金项目:江苏省自然科学基金重点项目(BK2011005),国家自然科学基金(61272221),江苏省社科基金(12YYA002)
收稿日期:2013－04－15

** 通讯联系人,E-mail:zhaobin@njnu.edu.cn

微博作为目前流行的网络营销媒体,为商业用户拓展市场空间、树立品牌形象和开展网络营销提供了巨大的帮助.但同时也成为了垃圾用户(Spammer)发布非法广告和垃圾消息的平台.垃圾消息的大量出现不仅影响了用户的使用体验,还妨碍了商业营销活动在微博平台上的健康发展.所以,研究微博中的反垃圾技术是一个具有理论意义和应用价值的研究课题.

在反垃圾方面,学术界已经开展了广泛的研究工作,如垃圾网页检测[1,2]、垃圾邮件过滤[3,4]、虚假的在线评论[5~8]和社交网络中的垃圾过滤[9,10]等.由于微博的反垃圾研究起步较晚,因而研究成果不多.文献[11]对 Twitter 上的网络犯罪生态系统开展研究,提出基于社交网络关系的检测算法,采用基于图模型的方法识别非法账户.文献[12]提出了一种度量用户行为相似性的 URL 预测方法,并用于检测 Twitter 中的垃圾广告和促销微博.

微博上的垃圾检测也可以采用重复检测(Duplicate Detection)的方法来实现.文献[13]介绍了多种基于聚类的重复检测算法,通过度量文本相似性实现聚类计算.文献[14]利用 SimHash 的低时间复杂度和低空间复杂度对大规模的网页数据进行重复检测.文献[15]采用时间窗口模型检测文本序列中的重用性.上述方法都可以通过度量微博中消息文本的相似性,达到检测垃圾消息的目的.

但是,借鉴已有的研究成果用于微博的反垃圾研究,存在以下几个问题:

(1)已有的研究工作大多针对文本特征进行垃圾检测,忽略了用户的行为特征.值得注意的是,垃圾用户的危害性不仅表现在消息内容上,还包括发布方式上.本文从用户发布消息的行为方式上对微博进行反垃圾研究.

(2)已有的微博反垃圾研究通常采用"关注关系"(Following Relationships)表示用户间的社交关系.但是,这样的关注关系与用户现实中的社交关系并不一致,因而不够准确和可信.本文采用用户的转发消息建模社交网络关系,准确识别垃圾用户之间的交互行为,实现垃圾用户的群体检测.

(3)重复检测模型的计算代价太高,并且不是针对微博文本而专门设计的,所以不适用于微博中的垃圾检测.本文提出的检测算法时间复杂度为 $O(n)$,适合处理类似微博这样的文本序列.

针对上述问题,本文的反垃圾研究对象针对垃圾用户,而不是垃圾消息本身,通过对垃圾用户发布消息的行为进行建模来设计检测算法.在考虑微博时序特性的情况下,结合消息序列中的文本相关性和时间相关性,提出了基于重用检测模型的微博垃圾用户检测算法.按照文本粒度的不同,检测算法分为语句级检测(Sentence-level Reuse Detection,SRD)和词项级检测(Term-level Reuse Detection,TRD).SRD 算法侧重于用户行为方面的检测,而 TRD 算法着重于指定主题的垃圾消息检测.采用上述检测算法,基于用户转发关系,本文还实现了垃圾用户的群体检测.需要说明的是,本文的微博反垃圾研究主要针对垃圾用户的转发消息.原因是转发消息传播性强、危害性大,理论上传播范围没有限制.所以转发消息值得被重点关注.当然,本文的研究工作也同样可以用于非转发的微博数据集.

1　垃圾用户检测的问题定义

1.1　问题定义　垃圾用户检测问题定义为,给定用户 u 及其转发的消息序列 $S_u=\{w_1,w_2,\cdots,w_n\}$,用户 u 在转发关系上的用户集合为 $V=\{v_1,v_2,\cdots,v_m\}$.按照转发关系将 S_u 划分成 $|V|$ 个子序列 S_{uv},即 $S_u=\bigcup_{v\in V}S_{uv}$.垃圾用户的判别方法是:(1)如果用户转发行为的评估函数 $F_{rt}(S_{uv})$ 的值大于阈值 θ_{rt},则此转发属于垃圾消息的发布行为;(2)如果用户 u 的评估函数 $F(S_u)$ 的值大于阈值 θ_{spam},则此用户为垃圾用户.

通常,垃圾用户发布行为的危害性包括两个方面:(1)内容方面,如非法消息、垃圾广告和虚假信息;(2)发布行为方面,短时间内发布大

量重复或者近似消息.无论是转发行为的评估函数 $F_{rt}(S_{uv})$,还是用户的评估函数 $F(S_u)$,都必须涵盖上述危害性的两个标准.

1.2 垃圾用户的行为建模 采用上节介绍的评估函数可以对垃圾用户行为的危害性进行评估,从而为构建垃圾用户的检测模型打下基础.

用户的评估函数 $F(S_u)$定义如下,

$$F(S_u)=\frac{\sum_{v\in V}F_{rt}(S_{uv})}{|V|} \tag{1}$$

函数 F_{rt}称为度量转发行为强度的评估函数,用于度量用户 u 和 v 之间转发行为的危害性,该函数需要同时考虑第 1.1 节中行为危害性的两个标准,即文本内容和行为方式.根据文本粒度的不同,将采用基于语句和词项两种不同的实现方法.基于语句的方法关注用户转发行为本身,而基于词项的方法更关注用户转发的内容.

2 重用检测模型

为了综合考虑文本相关性和时间相关性,本文参考"重用距离"模型,提出基于重用距离的重用检测模型,用以实现度量转发行为的评估函数 F_{rt}.

定义 1 重用距离(Reuse Distance)是一种用于局部性度量的计算模型[16],主要对序列中相同元素重复出现的情况进行量化处理,一般使用相同元素间的不同元素个数来表示.

例如,给定一个序列 $S=(a,b,c,c,d,a,e)$,则序列中两个 a 元素之间的重用距离为 3.

定义 2 重用检测(Reuse Detection)序列中元素的重用距离计算可基于队列结构实现,对序列中元素进行重用性度量的过程,称为重用检测.具体描述为:

给定序列 $S=\{e_1,e_2,\cdots,e_n\}$和检测窗口序列 Q,$Q\subseteq S$,Q 的长度阈值为 θ_q,即 $|Q|=\theta_q$.将 S 中元素依次加入 Q 中进行重用检测,得到命中列表 $H=\{\langle e,l\rangle|e\in S,l\in[0,\theta_q-1)\}$.当添加元素 $e_i\in S$ 时,进行如下操作:

(1)如果 e_i 和 Q 中已有的元素 e_j 恰好相同,则称为命中(Hit),此时计算队列 Q 中元素 e_i 和元素 e_j 之间的重用距离 l,从 Q 中删除 e_j 并将$\langle e,l\rangle$插入命中列表 H 中.

(2)如果没有命中任何元素,则插入 e_i 后检查队列 Q 元素个数是否超过阈值 θ_q,如果超过,则删除 Q 中首元素.

从上述检测模型可以发现:(1)序列 Q 中的元素永远保持唯一,一旦插入的元素命中队列中已有的元素,则重复元素被删除.(2)阈值 θ_q 限定了局部性检测的范围.越大,保留在队列中的元素个数就越多,否则只有频繁出现的元素才有可能保留下来.(3)经过重用检测处理后,得到命中列表 H,其中的元素具有局部邻近特性,正好对应于垃圾消息在微博时间线上聚集的特点.

3 基于重用检测模型的过滤算法

本节将采用重用检测模型度量微博用户行为的危害性.因为处理对象为微博消息,它不仅含有文本内容,还带有时戳属性.所以本节将在"命中规则"和"重用距离计算"这两方面对重用检测模型进行改造.

为了有效检测微博垃圾用户,基于上节的重用检测模型设计了两种不同文本粒度的检测算法,分别为语句级(Sentence-level)和词项级(Term-level)检测算法.语句级检测算法侧重于用户发布消息的行为方式,例如,垃圾用户通常采用批处理的方式短时间高频率发布大量重复或近似消息.而词项级检测算法不仅考虑时间因素,还考虑了微博消息的主题,可以弥补上述的不足.

3.1 基于语句级别的检测算法 基于语句级别的检测算法(Sentence-level Reuse Detection,SRD)主要偏重于用户的行为模式,典型的垃圾用户为了提高垃圾消息传播的效率,在短时间内发布大量重复消息或者近似消息.所以采用语句级别的重用检测算法可以有效发现他们.

但是,SRD 算法所采用的检测方法和第 2 节中的重用检测模型有几点不同:

(1)检测对象为序列 S_{uv},即用户 u 转发用户 v 的微博消息序列.序列 S_{uv} 按照发布时间顺序

排列,$S_{uw}=\{w_1,w_2,\cdots,w_n,w_i$ 为用户在 t_i 时刻发布的消息,t_i 为 w_i 的时戳属性,记作 $t_i=ts(w_i)$.

(2)为了度量发布行为的强度,采用时间作为重用距离更合适,相对应的重用队列阈值 θ_q 不再使用元素个数表示,而是时间间隔.

(3)命中列表 H 也需要相应调整,H 由相似文本序列 H_i 构成,即 $H=U_iH_i$,序列 $H_i=\{w_{i1},w_{i2},\cdots,w_{in_i}\}$,$w_{ij}\in S_{uw}$,元素按照消息文本发布时间排列.

在 SRD 算法中,经过重用检测处理后,式(1)中 $F_{rt}(S_{uw})$ 的函数可采用以下公式进行计算:

$$F_{rt}(S_{uw})=\frac{\sum_i f(H_i)}{|H|} \tag{2}$$

$$f(H_i)=\frac{\sum_{j=1}^{n_{i-1}} H(w_{ij},w_{i,j+1}}{|H_i|-1} \tag{3}$$

其中,w_{ij} 和 $w_{i,j+1}$ 为 H_i 中相邻的消息,函数 H 称为命中函数,用于度量序列中相邻消息的相关性,定义为:

$$H(w_i,w_{i+1})=R(w_i,w_{i+1})\cdot T(t_i,t_{i+1}) \tag{4}$$

其中,$R(w_i,w_{i+1})$ 用于度量消息文本相关性,而 $T(t_i,t_{i+1})$ 用来度量消息发布时间的相关性. 两种函数都要求实现归一化.

由于面对文本,因而命中规则同样需要调整,不能直接采用"完全相等"的比较方法,而应该使用更适合文本比较的相似性度量. 式(4)中函数 R 定义为:

$$R(w_i,w_j)=\mathrm{Sim}(\mathrm{Sig}(w_i),\mathrm{sig}(w_j)) \tag{5}$$

其中 w_i 和 w_j 是待比较的消息文本,Sim 函数采用 Jaccard 相似性度量计算文本的相似性,而 Sig 函数可以采用 Bloom-filter 签名技术实现,可以对文本进行高效地度量计算.

语句级重用检测过程如图 1 所示. 给定消息序列 $S_u=\{w_1,w_2,w_3,w_4,w_5\}$ 和重用检测队列 $Q=\{w_1,w_2,w_3,w_4\}$,且以 Bloom-filter 签名形式保存,重用队列阈值 θ_q 为 1 min,相似性阈值设定为 0.8. 当新元素 w_5 加入队列 Q,扫描 Q 中元素找出与之签名相似性最高且超过阈值的元素 $\mathrm{Sim}(\mathrm{Sig}(w_1),\mathrm{Sig}(w_5))=0.5$,$\mathrm{Sim}(\mathrm{Sig}(w_2),\mathrm{Sig}(w_5))=0.88$,$\mathrm{Sim}(\mathrm{Sig}(w_2),\mathrm{Sig}(w_5))=0.44$,$\mathrm{Sim}(\mathrm{Sig}(w_4),\mathrm{Sig}(w_5))=0.5$. 显然,$w_5$ 命中 w_2,重用距离 $D_{rs}(w_2,w_5)=25$ s. 命中后,将 w_5 和 w_2 插入命中列表中. 最后,将得到的命中列表代入式(2),用以评估转发行为的危害性.

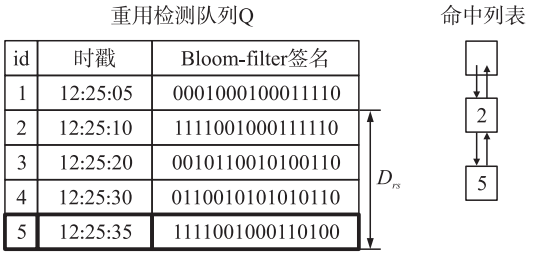

图 1　基于语句重用检测的示例

Fig. 1　Example of sentence-level reuse detection

3.2　基于词项级别的检测算法　语句级检测算法对于短期内连续出现的近似文本序列比较有效. 但是,如果垃圾用户改变发布行为的节奏,或者邻近文本相似性低于阈值,那么它的检测性能就会明显下降. 而且语句级检测算法忽略消息的主题信息,偏重于用户的行为模式,所以检测的针对性差. 本节介绍词项级检测算法(Term-level Reuse Detection, TRD)可以解决上述问题.

TRD 检测算法的检测对象不再是语句,而是词项. 给定消息序列 $S_u=\{w_1,w_2,\cdots,w_n\}$ 和词项集合 E,消息 w_i 由词项构成,即 $w_i\in S_u$,$w_i\subseteq E$,. 与 SRD 检测算法相比较,有以下几点不同:

(1)TRD 中的词项集合 E 是垃圾消息常用的代表性词集,如产品类别名称、品牌和营销词语等.

(2)原先的重用检测针对检测窗口队列 Q 中所有的元素进行命中操作,而 TRD 算法只针对相邻消息进行重用检测.

(3)命中列表 H 由序列 H_i 构成,即 $H=U_iH_i$. 每个 H_i 对应一个词项 $e_i\in E$. 序列 $H_i=\{t_{i1},t_{i2},\cdots,t_{in_i}\}$ 按照发布词项 e_i 的时间排列.

在 TRD 算法中,经过重用检测处理后,式(1)中的 $F_{rt}(S_{uw})$ 函数可采用以下公式进行计算:

$$F_{rt}(S_{uw})=\sum_i f(H_i) \tag{6}$$

$$f(H_i)=\frac{\sum_{j=1}^{n_i-1}H(t_{i,j},t_{i,j+1}}{|H_i|-1} \tag{7}$$

其中,函数 H 称为命中函数,用于度量序列中相邻词项的时间相关性,定义为:

$$H(t_i,t_{i+1})=T(t_i,t_{i+1}) \tag{8}$$

其中,t_i 和 t_{i+1} 为 H_i 中相邻的时戳,$T(t_i,t_{i+1}$ 用于度量词项在时间维度上的相关性,并且要求实现归一化.

值得说明的是,式(6)并没有归一化,这是因为 TRD 算法的检测依赖于垃圾代表性词集 E,如果 $F_{rt}(S_{uv})$ 值越高,则说明用户在转发中使用的垃圾代表性词项越频繁,因而该消息成为垃圾的可能性也越大.

词项级重用检测过程如图 2 所示. 给定用户消息序列 $S_u=\{w_1,w_2,w_3,w_4\}$,词项集合 $E=\{a,b,c,d\}$ 和重用检测队列 Q,重用队列阈值 θ_q 为 1 min. $w_1=\{b,d\}$,$w_2=\{a,b,c\}$,$w_3=\{b,c,d\}$,$w_4=\{c,d\}$. 最后,将得到的命中列表代入式(6),用以评估转发行为的危害性.

图 2 基于词项重用检测的示例

Fig. 2 Example of term-level reuse detection

3.3 基于转发关系的垃圾用户群体发现 文献[10]发现垃圾用户之间存在明显的社交网络关系,可是"关注关系"不能作为真实社交关系的依据. 为了实现垃圾用户的群体检测,本文采用基于转发行为的群体检测方法.

具体方法为,采用本文提出的重用检测算法为用户间的转发行为进行评估,如果用户间频繁发生转发行为,并且超过阈值. 则判定相关用户同为垃圾用户,且属于同一群体. 最后,采用连通分支生成算法就可以找出相互关联的垃圾用户群体. 第 4.6 节的实验证明了该方法的有效性.

3.4 算法复杂度分析 重用检测模型的时间复杂度为 $O(n\cdot\theta_q)$,$n=|S_u|$,为序列中的元素个数,θ_q 通常为固定值. 所以,重用检测算法的时间复杂度为 $O(n)$.

而空间复杂度方面,重用检测模型的存储空间包括重用检测队列和命中列表两个部分,而重用检测队列的空间并不是固定的,由时间间隔决定. 时间间隔越大,重用检测队列内空间占用越大. 而命中列表的空间大小由用户消息序列中的数据分布决定,对于 SRD 算法,由重复性或者近似重复文本在序列中的分布情况决定;对于 TRD 算法,由词项的分布决定. 总而言之,因为重用检测模型主要面对如非法信息、营销广告和垃圾消息这样的偏斜数据,因而在消息内容上和词项分布上都集中于少数数据. 所以,检测模型所占存储空间非常有限.

4 验结果与分析

4.1 实验数据集 本文实验采用的数据来自于新浪微博(www. weibo. com),通过新浪微博提供的 API 程序获得实验所需的真实数据集. 具体步骤为,挑选种子用户,沿着关注关系收集用户的微博消息序列,从中提取出转发消息作为实验数据.

表 1 微博测试数据集的统计情况

Table 1 Experiment dataset description

数据集	时间跨度	转发微博数	用户数
DSApril	2011 年 4 月 1 日	4474120	88308
DSSpam	至 5 月 1 日	33742	1000

本文实验有两个数据集,分别为 DSApril 数据集和 DSSpam 数据集. 表 1 列出了实验所用数据集的基本情况,包括微博消息总数、参与

的用户数和时间跨度. DSApril 数据集比较大,主要用于测试算法的检测效率,而 DSSpam 数据集比较小,通过对 DSApril 数据集随机抽取得到(为了保证测试效果,挑选总消息量超过 30 条的用户),主要用于算法检测效果的测试. 该数据集中垃圾用户占 19.3%.

4.2　垃圾用户的评判标准　本文实验判断用户为垃圾用户的标准为:

(1)在内容方面,用户消息序列中是否包含非法消息、垃圾广告和虚假信息等;

(2)在发布频率方面,短时间高频率重复发布(新浪规定正常用户 10 min 内不能发布相同内容);

为了测试垃圾检测算法的效果,采用 DSSpam 数据集,使用信息检索中的准确率、召回率和 F1 三种指标进行挖掘效果的测试.

4.3　基准测试算法　为了验证基于重用检测模型的两种不同算法的有效性和可行性,本次实验采用基于 Locality Sensitive Hashing[17] 的重复性检测算法(LSH)作为基准测试算法,该算法可以用来发现用户消息序列中重复或者近似的微博消息,结合时间间隔可以判断用户是否为垃圾用户,设定消息相似性阈值为 0.8. 本文提出的算法分别为语句级重用检测算法(SRD)和词项级重用检测算法(TRD). 通过测试检测算法的检测性能和检测效率,将验证基于重用检测模型的检测算法是有效的和高效的.

另外,TRD 算法需要采用外部的主题字典捕获指定主题的垃圾消息,所以本次实验从著名电商网站京东商城(www.jd.com)上搜集了商品的目录信息(1131 个代表性词项),用于垃圾消息主题字典的构建.

4.4　检测算法的性能比较　本文实验采用准确率、召回率和 F1 值三个指标对 LSH、SRD 和 TRD 三种算法的检测性能进行比较. 其中,LSH 算法在原理上采用重复性检测方法,和重用检测时间间隔无关,因而在所有性能比较图上都表现为直线,如图 3 所示.

在准确率方面,图 3a 表明随着重用时间间隔的拉长,SRD 算法和 TRD 算法的准确率逐渐降低,且 SRD 算法和 LSH 算法明显优于 TRD. 主要原因是 SRD 算法和 LSH 算法比 TRD 算法更关注垃圾用户的行为方式,垃圾用户发布的主要方式仍然是大规模发布重复的或高相似性的消息内容. 而 SRD 算法的重用时间间隔越短,检测效果越好,越容易发现高频率和高强度的垃圾用户,而这些用户对于整个微博平台的危害性最大.

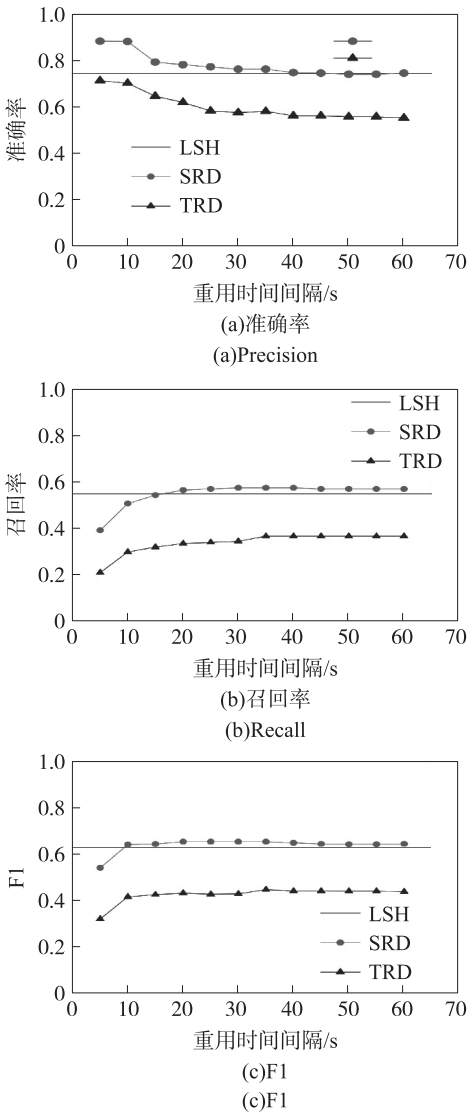

图 3　LSH、SRD 和 TRD 算法在准确率、召回率和 F1 值上的比较

Fig. 3　Performance on spam detection under LSH, SRD and TRD algorithms

在召回率方面,图 3b 显示随着重用时间间隔的增长,算法 SRD 和 TRD 的召回率逐渐提高. 因为时间间隔的拉长可以发现一些中等发布强度的垃圾用户. 三种算法对比发现,算法 SRD 和 LSH 明显优于 TRD. TRD 召回率低的原因和它采用的主题字典有关.

在 F1 值比较方面,图 3c 仍然表明了算法 SRD 和 LSH 在检测性能上整体优于 TRD 算法.

除此以外,在实验中发现 TRD 算法可以检测出一些文本重复性低的垃圾用户. 表 2 列出在不同重用时间粒度下,TRD 算法可以发现而 SRD 算法没有发现的垃圾用户比率. 通过分析发现,这些用户在消息序列上整体文本相似性不高,但大多围绕特定主题. 由于本次实验所用的主题字典由商品类的代表性词项构成,所以他们是一群比较潜在的以商品为主题的垃圾用户.

表 2 SRD 算法对比 TRD 算法漏检的垃圾消息比率

Table 2 Ratio between spamming messages neglected by SRD and that of TRD

重用时间间隔/s	漏检率
10	7.2%
20	5.6%
30	5.7%
40	5.7%
50	5.7%
60	5.7%

4.5 检测算法的执行效率比较 本次实验运行在一个单机平台上,采用 Intel Core i7 2.93 GHz CPU 和 16 GB 内存,操作系统为 Ubuntu Linux 12.04. 在运行时间方面,表 3 清楚表明 TRD 的运行时间最短,LSH 运行时间最长,并且可以发现,由于重用检测模型的时间复杂度为 O(n),采用单遍扫描方式处理消息序列,因而基于此模型的算法对于重用时间间隔的变化不敏感.

表 3 LSH、SRD 和 TRD 算法的运行时间比较(分钟)

Table 3 Runtime under LSH, SRD and TRD algorithms (minutes)

重用时间间隔/s	LSH	SRD	TRD
10	300	77	10
20	300	75	10
30	300	74	10
40	300	74	10
50	300	74	10
60	300	75	10

4.6 垃圾用户群体发现 以 DSApril 数据集为实验对象,设置重用时间间隔为 10 s,命中的文本相似度阈值为 0.8. 采用 SRD 算法共发现垃圾用户群体 31 个,共 147 个用户. 表 4 列出了排名前 5 的用户集合. 需要说明的是,为了保证集合质量,选择用户数大于 4 的集合作为结果返回.

表 4 垃圾用户群体检测的实验结果

Table 4 Spammer group detection

排名	用户数	描述
1	24	汽车主题,包括车展、汽车保养、试驾等
2	11	演唱会门票
3	8	演艺人员
4	7	3G 产品推广
5	6	自驾游

5 结束语

针对微博垃圾用户的检测问题,本文提出了基于重用检测模型的过滤算法. 按照文本粒度的不同,分为语句级检测 SRD 算法和词项级检测 TRD 算法. SRD 算法侧重于用户的行为方式,检测准确率比较高;TRD 算法侧重于检测指定主题的垃圾消息,算法性能受主题字典影响较大,但算法执行效率高于 SRD 算法. 最后,通过基于真实数据集的实验验证了上述两种检测算法的有效性和可行性.

References

[1] Wang Y, Ma M, Niu Y, *et al*. Spam double-funnel: Connecting web spammers with advertisers. Williamson C L, Zurko M E, Patel-Schneider P F, *et al*. Proceedings of the 16th International Conference on World Wide Web, Banff, Alberta, Canada, May 8－12, 2007, 291～300.

[2] Cheng Z, Gao B, Sun C, *et al*. Let web spammers expose themselves. King I, Nejdl W, Li H. Proceedings of the 4th International Conference on Web Search and Web Data Mining, Hong Kong, China, February 9－12, 2011, 525～534.

[3] Dasgupta A, Gurevich M, Punera K. Enhanced email spam filtering through combining similarity graphs. King I, Nejdl W, Li H. Proceedings of the 4th International Conference on Web Search and Web Data Mining, Hong Kong, China, February 9－12, 2011, 785～794.

[4] Fette I, Sadeh N, Tomasic A. Learning to detect phishing emails. Williamson C L, Zurko M E, Patel-Schneider P F, *et al*. Proceedings of the 16th International Conference on World Wide Web, Banff, Alberta, Canada, May 8－12, 2007, 649～656.

[5] Jindal N, Liu B. Opinion spam and analysis. Najork M, Broder A Z, Chakrabarti S. Proceedings of the International Conference on Web Search and Web Data Mining, Palo Alto, California, USA, February 11－12, 2008, 219～230.

[6] Ott M, Choi Y, Cardie C, *et al*. Finding deceptive opinion spam by any stretch of the imagination. Lin D, Matsumoto Y, Mihalcea R. The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19－24 June, 2011, Portland, Oregon, USA. The Association for Computer Linguistics 2011, 309～319.

[7] Kant R, Sengamedu S H, Kumar K S. Comment spam detection by sequence mining. Adar E, Teevan J, Agichtein E, *et al*. Proceedings of the 5th International Conference on Web Search and Web Data Mining, Seattle, WA, USA, February 8－12, 2012, 183～192.

[8] Mukherjee A, Liu B, Glance N S. Spotting fake reviewer groups in consumer reviews. Mille A, Gandon P L, Misselis J, *et al*. Proceedings of the 21st World Wide Web Conference 2012, Lyon, France, April 16－20, 2012, 191～200.

[9] Benevenuto F, Rodrigues T, Almeida V A F, *et al*. Detecting spammers and content promoters in online video social networks. Allan J, Aslam J A, Sanderson M, et al. Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Boston, MA, USA, July 19－23, 2009, 620～627.

[10] Lee K, Caverlee J, Webb S. Uncovering social spammers: social honeypots + machine learning. Crestani F, Marchand-Maillet S, Chen H H, *et al*. Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Geneva, Switzerland, July 19－23, 2010, 435～442.

[11] Yang C, Harkreader R C, Zhang J L, *et al*. Analyzing spammers' social networks for fun and profit: A case study of cyber criminal ecosystem on twitter. Mille A, Gandon P L, Misselis J, *et al*. Proceedings of the 21st World Wide Web Conference 2012, Lyon, France, April 16－20, 2012, 71～80.

[12] Zhang X C, Zhu S P, Liang W X. Detecting spam and promoting campaigns in the twitter social network. Zaki M J, Siebes A, Yu X, *et al*. 12th IEEE International Conference on Data Mining, Brussels, Belgium, December 10－13, 2012, 1194～1199.

[13] Hassanzadeh O, Chiang F, Miller R J, *et al*. Framework for evaluating clustering algorithms in duplicate detection. Abiteboul S, Markl V, Milo T, *et al*. Proceedings of the VLDB Endowment, 2009, 2 (1), Lyon, France, 1282～1293.

[14] Sood S, Loguinov D. Probabilistic near-duplicate detection using simhash. Macdonald C, Ounis I, Ruthven I. Proceedings of the 20th ACM Conference on Information and Knowledge Management, Glasgow, United Kingdom, October 24－28, 2011, 1117～1126.

[15] Draisbach U, Naumann F, Szott S, *et al*. Adaptive

windows for duplicate detection. Kementsietsidis A, Salles M A V. IEEE 28th International Conference on Data Engineering, Washington, DC, USA, 1－5 April, 2012, 1073～1083.

[16] Ding C, Zhong Y T. Predicting whole-program locality through reuse distance analysis. Cytron R, Gupta R. Proceedings of the ACM SIGPLAN 2003 Conference on Programming Language Design and Implementation 2003, San Diego, California, USA, June 9－11, 2003, 245～257.

[17] Indyk P, Motwani R. Approximate nearest neighbors: Towards removing the curse of dimensionality. Vitter J S. Proceedings of the 30th Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23－26, 1998, 604～613.