第 35 卷　第 1 期　　计　算　机　工　程　　2009 年 1 月
Vol.35　No.1　　Computer Engineering　　January 2009
·开发研究与设计技术·　文章编号：1000—3428(2009)01—0283—03　文献标识码：A　中图分类号：TP393

基于 Struts 框架的垃圾短信过滤模块的设计

杨妮娜[1]，王　庆[2]
(1. 西北工业大学软件与微电子学院，西安 710072；2. 西北工业大学计算机学院，西安 710072)

摘　要：为了提高对垃圾短信的拦截效率，提出一种同时从发送号码、发送频率以及短信内容 3 个方面对垃圾短信进行过滤的方法。通过黑白名单从发送号码进行一次过滤，对群发短信进行内容分析过滤，发送频率的引入可以实现黑名单的自动生成。以内容过滤为核心，并对其进行了阐述，基于 Struts 框架进行设计与实现了一个垃圾短信拦截模块。实验结果表明，查准率达到了 90.69%。

关键词：垃圾短信；Struts 框架；模糊匹配；过滤机制

Design of Junk Short Message Filter Module Based on Struts Framework

YANG Ni-na[1], WANG Qing[2]
(1. College of Software and Micro-electronics, Northwestern Polytechnical University, Xi'an 710072;
2. School of Computer Science, Northwestern Polytechnical University, Xi'an 710072)

【Abstract】 In order to improve junk message intercepting efficiency, one scheme is adopted by filtering junk message according to sending numbers, sending frequencies and content synchronously, which filters sending numbers based on the black-and-white list first and analyzes the spam content afterwards. Sending frequencies is applied to realize the blacklist generation. The core of this method is content filtering which is expounded. This paper designs a spam intercepting module just based on the Struts framework. Experimental result shows that the system's succee can reach 90.69%.

【Key words】 junk short message; Struts framework; approximate matching; filter system

近年来，依托通信网络和互联网的手机短信已成为重要的人际交往和沟通手段。从用户角度来看，短信之所以深受人们青睐，最大的特点就在于它的费用低、实时性高。但同时，短信也成为散布广告、诈骗、色情信息的工具。垃圾短信的泛滥已经成为继垃圾邮件之后，困扰人们生活的又一大安全隐患。目前的短信治理方式[1]大多都是在用户收到短信后进行，特别是有些用户在陷入骗局、陷阱之后才寻求解决办法，这种事后监管多少有些被动，而且监管成本也比较高，所以，要尽可能地将垃圾短信拒在用户之外。

1　过滤系统的设计思想

随着 Internet 和 IT 行业技术的发展，人们已经可以通过 PC 发送短信。在网络上，向手机发送短信的逻辑流程如图 1 所示。从图中可以看出，垃圾信息不仅占用了用户的存储空间，而且还浪费了大量的网络资源。在网络传输正常的情况下，如何快速有效地阻止垃圾短信在网络中的蔓延，成为服务提供商(SP)越来越关注的问题。本文就如何利用黑白名单、发送频率和短信内容 3 方面过滤垃圾短信进行讨论。

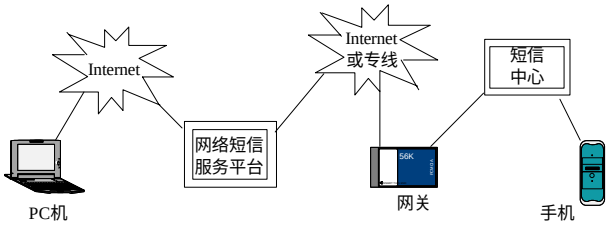

图 1　短信发送流程

在这里通过Web页面发送短信的服务平台采用基于Struts[2]的MVC框架。Struts是基于MVC模式设计的Web框架：M(Model)：模型，V(View)：视图，C(Controller)：控制器。在本文设计中，模型由实现业务逻辑的Java Bean组件构成，控制器由ActionServlet和Action来实现，视图由一组JSP文件构成。图 2 显示了在本设计中采用的Struts所实现的MVC框架。采用Struts框架可将Web应用中的视图和模型分离，使得应用结构更加清楚，同时提高代码的重复性、维护性和扩展性，这也是Struts框架区别与其他框架的重要标志。

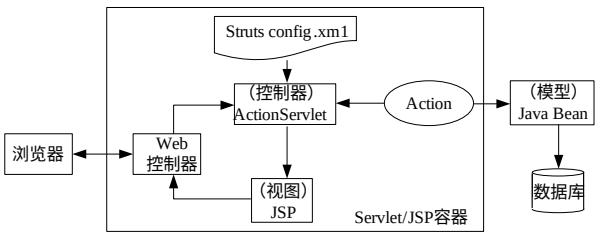

图 2　基于 Struts 的 MVC 框架

根据图 2，搭建短信服务平台，并在此基础上扩充功能，增加了过滤模块。在过滤模块中对短信进行过滤的流程如图 3 所示。

作者简介：杨妮娜(1982－)，女，硕士研究生，主研方向：网络技术；王　庆，教授、博士生导师

收稿日期：2008-05-08　**E-mail：**ynn-1982@163.com

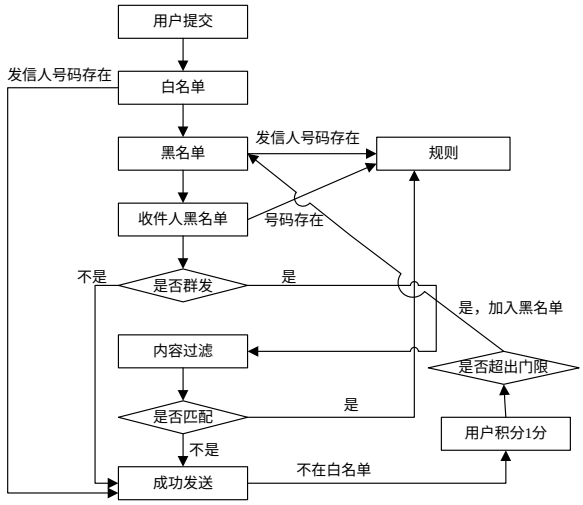

图 3　短信过滤流程

为实现图 3 所描述的流程，在执行发送短信的 Action 和 JavaBean 中共增加了 4 个类：CheckBlackNum, CheckWhiteNum, CheckJunk 和 filter。CheckBlackNum 是检查黑名单的程序，CheckWhiteNum 是检查白名单的程序，CheckJunk 是进行短信内容检查的程序，filter 则是用来为用户的发送数量记分的程序。程序中还调用 log4j 接口，对所执行的过滤动作进行详细的日志记录。

黑白名单规则具有较高的执行效率，内容过滤具有较高的判别精度。在过滤机制中本文将讨论内容过滤的算法。

2　黑白名单过滤的设计

用户登录系统时，UserAction.java 将用户登录信息添加到 session 中。当用户执行短信发送功能时，相应的 Action 类调用黑白名单对用户进行检查，将非法短信制造者封杀。

黑白名单中的信息都是保存在数据库中的，执行过滤规则时必须从数据库中读取数据。在实际应用系统，同一时段会有大量用户同时访问数据库，频繁的访问势必造成数据库的瘫痪，消耗对象资源。因此，程序中采用 JCS 来缓存黑白名单信息(以及关键字信息)以提高性能，每隔一阶段自动对缓存中的数据进行刷新。

JCS 是 Jakarta Turbine 项目的一部分，除了简单地将对象缓冲在内存中以外，还为企业级的缓冲系统提供了许多重要功能。如内存管理、数据期限管理、并行式的分布缓冲等。

采用 JCS 开发的时候不用考虑底层的缓冲配置构架。同一个应用，只需要修改 Web-INF/classes 目录下配置文件 cache.ccf，就可以改变缓冲构架。

鉴于短信发送号码难以伪造，在短信服务系统中使用黑白名单方法是很有效的。在实际工作中，也常遇到某个部门因为工作的需要，需群发大量短信，这时只要向管理员申请，加入白名单，无论何时都能正常发送短信。有些短信，对于一部分用户来说是垃圾信息，而恰恰是另一部分用户希望收到的信息，比如某个商场在做打折活动，反对的用户只要向管理员投诉，将自己的号码列入收信人黑名单，以后就会屏蔽来自这个发送打折信息号码的短信。还可以利用用户发送频率的大小来判定其是否被列入“黑名单”，这将在下节中进行具体的阐述。

3　频率过滤设计

频率过滤主要是基于对某设定的时间段内用户发送的短信数目进行监督来实现的。垃圾信息制造者一般是集中某一时刻，连续不断地向外群发大量信息，利用这一特征，通过为用户进行打分，可以统计出用户在一定时间内发送相同信息的总数目。为此，设定一个门限时间，假定是 5 min。用户登录系统首次进行发送时，启动计时器，用户每成功发送一条短信，为用户记一分，当定时器到时时，如果用户的发送量超过在该时间间隔内允许的最大量时，自动将用户加入黑名单。如果没有，则重新启动计时器。这一功能由 Struts 的 filter 过滤组件来实现。

4　内容过滤设计

4.1　相关知识

短信发送的时间消耗是必须考虑的问题。通过黑白名单的检查后，根据每一次发送接收号码的数量，来判断是否是短信群发，如果是才进行内容检查过滤。

短信的内容过滤简单地说，也就是通过字符串匹配来进行过滤。将网络传送中的每条短信与事先设定好的词库中的关键词进行匹配，如果匹配成功即认为该条短信为垃圾短信，执行拦截规则。

字符串匹配模式在信息检索、生物学、模式识别等领域都占有重要地位，是计算机学科算法设计的一大热点。早期的研究多限于精确字符串的匹配，前人提出了经典的 RK(Rabin-Karp),KMP(Knuth-Morris-Pratt),BM(Boyer-Moore) 利用有限自动机进行字符串匹配[3]等算法。

精确匹配算法的应用也在网络平台系统中发挥过优势，起到了一定的过滤效果。但垃圾短信制造者们通过对所发送的信息进行伪装变换，躲避传统精确关键字匹配过滤机制的封堵，令目前的监管控制系统防不胜防，为此有必要研究字符串的模糊匹配算法，以便将伪装后的短信及时给予封杀。

所谓模糊匹配可以描述为：已知长度为n的字符串t，长度为m的模式串p以及一个正整数$k<m$，找出t中所有满足$ed(s, p)<=k$的字串s。其中，$ed(s, p)$是指把p转变成s所需要的修改次数[4]。

短信服务标准规定：一条短信最多只有 70 个汉字。垃圾信息制造者要想在如此短的字数内表达清楚其内容，对短信的伪装变化有以下 2 种情况：

(1)用同音字替换

如：将“幸运者”写成“信用者”。

(2)增加特殊符号

如：将“卫星电视”写成“卫星***电#视”、“&r 卫星电视”等。

计算机对汉字的存储不同于一般字符的存储，每个汉字占 2 个字节，有时候前一汉字的第 2 个字节和后一个汉字的前一个字节组成新的字符，导致一些算法出现异常，不该出现匹配的地方出现匹配。本文受NEW-BYG[5]的启发，对KMP算法进行了改进，使之在适应汉字与字符混合的情况下，能进行近似匹配。

4.2　内容过滤算法

算法实现思想：用 $T[i]$和 $P[j]$分别表示当前正在接受比较的一对字符，wrong 为允许产生误差的个数。当比较发生失配后，首先判断 $T[i]$是否是为汉字，如果不是，就用下一个字符 $T[i+1]$和 $P[j]$进行匹配，否则分成是否找到第 1 个匹配位置 2 种情况讨论。重复此过程直到 $i=m$ 或 $j=n$ 结束。

改进的算法伪代码如下：

输入：文本串 $T[1, m]$和模式串 $P[1, n]$，num 为找到匹配的个数，count 用来记录不匹配的个数

```
while i<m and j<n
do
    if  T[i] = = P[j] then
        num 加 1，继续比较 T 和 P 的下一个字符
    else
        if T[i]不是汉字，而 P[j]是汉字，then
            用 T 的下一字符和 P 比较
        else
            if  num＝＝0 then
                if  T[i+1] = = P[j+1] then
                    num 加 1 ，继续比较 T 和 P 的后 2 个字符
                else 用 T 的下一字符和 P 比较
            else count 加 1，继续比较 T 和 P 的下一个字符
                if  count 的数目超过允许的最大误差数目 wrong then
                    count 清零 ，num 清零，j 回到第 1 次出先不匹配的位置，即 j=next[j-count+1]准备下一轮的匹配。
```

算法分析：KMP 算法是经典的精确匹配算法，为了实现模糊匹配，在改进的算法中，当 2 个字符失配时，并不是立即回溯，而是作进一步判断，若文本串相对于模式串出现了特殊符号，则直接跳过该符号；没有特殊符号时，若文本中已经找到了部分匹配，且不匹配个数在允许的范围内，继续后面字符的比较，直至误差超出范围且循环没有结束，才进行回退。为避免只对文本串中的首字符做替换，在没有找到匹配前，会根据文本串和模式串中下一字符是否匹配决定 2 个字串的移动情况。最后，根据返回的结果求解相似度，即查找到的匹配字符的个数占模式串总长度的百分比。相似度的定义如下：

```
quote = num/n;
```

如果 quote>阈值，则 return 是垃圾短信。

显然，理论上该算法在最坏情况下的比较次数没有超过文本串的长度，与 KMP 算法相比，也没有增加时间复杂度，而且简单易于实现。

4.3 内容规则的形成

一条短信，除了收发双方的号码和区区几十个文本文字，可用的信息相对较少，传统生成关键词的简单规则，很容易造成误判。

在该过滤模型中添加 log4j 日志器，定期收集系统内一定时间段内出现的短信，通过检查短信之间的相似性，将具有一定相似程度而且数量超过一定限制的短信分离出来，发邮件给管理员，管理员可以手工对短信进行判断，适当地修改规则。

垃圾短信有生存周期，同一内容的垃圾短信只会在同一时期内被重复发送，过旧的规则没有什么价值，反而占用数据库空间，降低系统的性能，所以，对于每一条规则设定一个时间期限，当该规则到达所设期限，自动被清除。

5 实验结果

为了验证该过滤机制的有效性，笔者做了以下测试工作：

(1)测试环境

客户端采用 Windows 操作系统、IE6.0。

服务器采用 Linux 操作系统。

数据库采用 mysql5.0。

(2)验证内容

(1)KMP算法不断将模式串和文本串比较，一旦局部失配，则利用此前比较所给出的信息，尽可能长距离地移动模式串，它的复杂度与待匹配的文本串和模式串的因素无关，又可以避免很多不必要的比较操作[3]。整个过程的时间复杂度为$O(n+m)$(n和m是待匹配的文本串和模式串的长度)。但是KMP原本是精确算法，新算法在KMP的基础上做了改进，使之可以进行简单的字串模糊匹配。收集到 1 000 条内容规则，对比新内容过滤算法和KMP算法查询的效率。实验结果显示，在相同的实验环境下，新算法查询时间约为 0.95 s，与KMP查询时间 0.93 s没有太大差异。这表明新算法并没有增加算法的时间复杂度，既可以满足系统对垃圾短信内容进行模糊匹配，又可以满足短信传输过程中要求的及时性。

(2)将该过滤模块部署到一通过 Web 页面可以发送短信的实际运行环境中，经一个月的观察，对采集到的 1 800 个短信发送号码分析如表 1 所示，其中，发送垃圾短信和发送正常短信的用户各占一半。

表 1 垃圾短信数据统计表

	系统判定垃圾短信	系统判定正常短信
人工判定垃圾短信/条	838	62
人工判定正常短信/条	86	814
查准率/(%)	90.69	
查全率/(%)	93.11	

(3)根据该模块中日志的统计分析，得出从发送号码、发送频度以及短信内容 3 方面对垃圾短信过滤的比例如图 4 所示。从图中可以看出，黑白名单规则和频度积分规则引入后，能更加有效地将垃圾短信阻拦在源端，减少垃圾短信对网络负载的冲击。

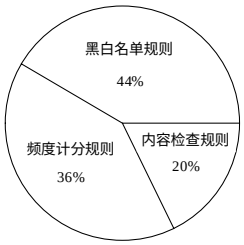

图 4 各个单元拦截垃圾短信的比例

6 结束语

本文提出了一种完整的基于 Struts 框架的短信过滤机制，将黑白名单、发送频率和内容审查 3 种过滤方法结合使用，首先根据实际需要制定“黑白名单”，然后对用户的发送短信内容进行内容过滤，符合要求的方允许发送，最后对用户在设定的时间段内的发送短信数量予以监督，超过门限值且不在白名单中的实行频率过滤。实验结果表明，该方法的查准率达到了 90.69%，在对垃圾短信实际拦截工程中具有很好的应用前景。但是垃圾短信的防范治理需要社会各界的共同关注。除了运营商要在技术上不断更新升级和客户积极参与举报，还应有一套合理健全的法律法规，从严惩治那些散布垃圾信息的散发者。

参考文献

[1] 严奇春. 短信治理呼唤“黑名单”功能[Z]. (2007-05-20). http://www.cnii.com.cn/20070520/ca412587.

[2] 闻 涛. Struts 网络编程例学与实践[M]. 北京: 清华大学出版社, 2006-04.

[3] 邓俊辉. 数据结构与算法——Java 语言描述[M]. 1 版. 北京: 机械工业出版社, 2006-02.

[4] 陈开渠, 赵 洁, 彭志威. 快速中文字符串模糊匹配算法[J]. 中文信息学报, 2004, 18(2): 58-65.

[5] 安世虎, 刘淑辉. 模式匹配问题的进一步研究[J]. 计算机应用研究, 1998, 15(4): 13-15.