代　号 10701　　学　号 1021221345
分类号 TP311.5　　密　级 公开

题（中、英文）目 邮件过滤系统的研究与实现

Research and Implementation of Email Filtering System

作 者 姓 名 袁晓容　**指导教师姓名、职务** 郑有才　副教授

学 科 门 类 工 学　**学 科 专 业** 计算机软件与理论

提交论文日期 二〇一三年一月

西安电子科技大学
学位论文独创性（或创新性）声明

秉承学校严谨的学分和优良的科学道德，本人声明所呈交的论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果；也不包含为获得西安电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中做了明确的说明并表示了谢意。

申请学位论文与资料若有不实之处，本人承担一切的法律责任。

本人签名：_____ 日期_____

西安电子科技大学
关于论文使用授权的说明

本人完全了解西安电子科技大学有关保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属西安电子科技大学。学校有权保留送交论文的复印件，允许查阅和借阅论文；学校可以公布论文的全部或部分内容，可以允许采用影印、缩印或其它复制手段保存论文。同时本人保证，毕业后结合学位论文研究课题再撰写的文章一律署名单位为西安电子科技大学。（保密的论文在解密后遵守此规定）

本学位论文属于保密，在___年解密后适用本授权书。

本人签名：_____ 日期_____

导师签名：_____ 日期_____

摘要

日益泛滥的垃圾邮件给普通百姓的生活带来了诸多不便，也给某特定应用领域带来了麻烦。因此设计一种高效的和广泛应用的邮件过滤系统是一件很有意义的事情。

本文在分析了传统邮件过滤方法的基础上，针对传统过滤方法容易漏判垃圾邮件和误判正常邮件，设计和实现了一个邮件过滤系统。该系统采用对过滤 URL 模块的 URL 比较方法和贝叶斯分类模块的朴素贝叶斯分类方法进行了算法改进，使邮件过滤效率有较大的提高，而且在哈希表查找模块应用哈希表查找关键词，在过滤系统重要用户模块建立重要用户数据库和去重复模块中建立重复邮件标准，提高了查找速率和过滤准确性。

最后搭建测试平台对每个模块进行单元功能测试，再对过滤系统进行性能测试。通过对测试的结果的分析，表明该邮件过滤系统过滤垃圾邮件具有较好的功能和较高的性能。

关键词：垃圾邮件 贝叶斯算法 重复邮件 URL 比较 哈希表

Abstract

The increasing number of junk email has not only brought a lot of inconvenience to people's life, but also brought lots of trouble to a certain special field, and therefore, it is very meaningful to design an efficient and widely-used junk email filtering.

This paper, based on the analysis of the traditional e-mail filtering methods, has designed and implemented an e-mail filtering system to deal with the problems of easily missing junk mails and misjudging normal mails by traditional e-mail filtering methods. This system has improved algorithm in the URL comparison method of the URL Filtering Module and the Naïve Bayesian classification of Bayesian Classification Module, and greatly improved the mail filtering efficiency. What's more, this system has applied hash table to search keywords in hash table module, established the important user database in important user module, created a duplicate mail standard in reduplication module, and thus improved the Find rate and the filtering accuracy.

Finally, this paper has built a test platform for each module functional testing and performance testing of the filtering system. The analysis of the testing results shows that the e-mail filtering system has better functions and a higher performance for filtering junk e-mails.

Keywords: Junk email Bayes algorithm Repeat email URL Algorithm Hash tale

目录

第一章 绪论

1.1 研究背景和意义

随着互联网技术的普及，电子邮件日益成为人们生活中的重要部分。传统的书信和贺卡逐渐地被电子邮件所取代。电子邮件凭借其方便、快捷和环保的优点给人们的生活带来了便利，越来越多的人从邮件中获取商业信息以及个人信息。然而垃圾邮件的产生也带给了人们很多的麻烦。大量的垃圾邮件侵占了存储空间，浪费了人们的时间、精力和上网流量费用。因此过滤垃圾邮件可以大大节约社会资源，提高工作效率，减少个人和单位损失，营造一个良好的公共网络环境。

网络在给现代社会普通百姓带来方便的同时也给某特定应用领域带来了生活上的便捷。通过邮件进行信息交流也成为了人们信息共享的主要方式，而垃圾邮件也给大家的生活带来了很大的麻烦，因此对垃圾邮件过滤系统的研究具有重要的意义。

虽然目前有很多邮件过滤系统的软件，但是这样的过滤系统一般是只针对普通的个人用户或者地方单位，没有涉及到某特定应用领域。本文设计一个邮件过滤系统，既可以适用于普通百姓，也可以适用于某特定应用领域的需要。

1.2 邮件过滤研究现状

垃圾邮件[1]是指未经收件人的许可，发件人把一些对收件人无用的邮件发送到收件人的信箱里面的邮件。垃圾邮件的特征：(1)邮件头的信息很不全或者隐藏全部邮件头信息；(2)伪造发件人信息，含有虚假的路由信息；(3)用户没有订阅的纯商业广告；(4)不健康的内容宣传图片和超链接。(5)邮件的日期显示是几个月前的邮件或者是几年前的老邮件。

一般的垃圾邮件的来源包含以下几种情况：(1)商业广告，图片和链接。(2)隐藏 IP 的反动邮件。(3)收件人重复收取而且无法拒绝的邮件。(4)虚假信息、骗人的钓鱼链接和广告。（5）反党反人民的恶意宣传邮件。

根据中国互联网协会反垃圾邮件中心(www.anti-spam.cn)的统计显示[2]，2007 年中国网民收到的垃圾邮件总量为 694 亿个，相比 2006 年的 500 亿个。 增长率达到 38.8%。相应的评估显示，2007 年垃圾邮件给中国造成的经济损失高达 188.4 亿元，与 2006 年 104.315 亿元的经济损失相比较，其损失增长高达 80.6%。

据有关报道显示，国外一些间谍分子，利用垃圾邮件策反了某特定领域的个别人员为他们服务，给国家安全带来很大的危害。

可见随着互联网应用规模的扩展，垃圾邮件的数量也在增大，对社会和特定

领域造成的危害也越来越大。垃圾邮件的危害主要表现在以下几个具体的方面：

(1)占用网络带宽，浪费网络资源，造成邮件服务器拥塞，进而降低整个网络的运行效率。当网路上充斥了大量的垃圾邮件，就会占用网路带宽。大量垃圾邮件，也占用了服务器存储。垃圾邮件也损害了 ISP 的市场形象,扰乱了 ISP 市场，造成无形资产流失。

(2)浪费用户的宝贵时间和上网费用，占用了邮件空间。如果我们每天都要花费时间处理垃圾邮件，就会降低了个人工作效率。对个人来说，是浪费了时间和上网流量，对整个社会来说，就是浪费了社会资源,减少了人们创造的社会财富。

(3)对网络安全形成威胁。一些不法之徒通过垃圾邮件散布谣言迷糊大众,扰乱社会治安。尤其是那些妖言惑众，骗人钱财，传播色情等方面的垃圾邮件，将严重危害社会。

（4）对某特定应用领域人的思想侵蚀。一些国内外反革命分子，通过垃圾邮件毒害人的思想，宣传反革命言论，给该领域的人思想稳定造成负面影响。有些变节的分子通过邮件和国外反革命分子勾结在一个给该领域安全造成危害。

随着垃圾邮件的日益泛滥，反垃圾邮件的技术也日益提高。目前，常用的邮件过滤技术有以下几个方面：

(1)基于邮件主题、邮件正文的关键词过滤[3]。通过判读邮件主题或者邮件的内容，是否包含某些关键字符来判读邮件是不是垃圾邮件。关键词既可以是字符，也可以是字符串，还可以是特定的符合。一般采用正则表达式对关键词进行匹配。该技术简单易行。该方法的优点是很好地过滤了特定的邮件，缺点是需要定期更新关键词，维护比较麻烦。

(2)基于黑白名单的过滤[3]。黑名单过滤，就是先检查邮件头的发件人和服务器信息，信息包含黑名单内的反党反人民分子，该邮件就是垃圾邮件。基于白名单过滤是一个相反的。白名单的内容包含了许可的内容，比如发件人是某部的司令部参谋。如果，邮件的发件人和服务器存在白名单中，该邮件就是一封正常邮件。否则，是一份垃圾邮件。黑名单的优点是可以快速过滤已知的黑名单用户，缺点是漏判了很多其他的垃圾邮件。白名单的优点是可以百分之百过滤垃圾邮件，缺点是会把一些正常邮件当作垃圾邮件给过滤掉。

(3)基于规则的过滤[4]。依据邮件的某些特征整理出规则。当新的邮件来的时候，就依据这些规则来判定邮件的类别。该技术，需要维护一个庞大的规则库来判读邮件的类别。该方法，也是使用比较广泛一种方法，能够过滤同一规则的邮件，缺点是制定和维护规格是一件很麻烦的事情。

(4)基于意图的检测过滤。常用的一个意图检测过滤就过滤含非法 URL 邮

件,URL(Universal Resource)中文名翻译是统一资源定位符，也就称作网页地址或者叫网页链接,，是描写互联网上的网页地址一种标志方法。该检测方法是检测该 URL 指向的内容，来判断该邮件是否为垃圾邮件。

(5)基于统计算法的过滤[4]。在这个方法中，使用的最多的是贝叶斯统计方法，贝叶斯分类的原理是，将邮件中分词当作特征项，先进行大量的训练获取这些特征在邮件中的先验概率，利用统计贝叶斯公式计算概率，算出需要分类的邮件属于垃圾邮件后验概率。统计方法广泛应用于邮件过滤，能够过滤大部分的垃圾邮件，不过需要训练大量的样本邮件及时反馈更新特征库。

1.3 本文的工作

本文设计的邮件过滤系统主要过滤的垃圾邮件是无用的多媒体邮件以及重复的邮件。需要保留的邮件主要是生活中重要的邮件以及有职业特征特定内容的邮件。本文研究的对象是日常公开生活网络邮件管理状况。研究的内容不涉及到保密内容，只涉及到一些某特定领域职业字样特征的内容。研究的目的是方便大家管理日常邮件，提高工作效率。本文主要工作如下：

（1） 对传统的邮件过滤方法进行了简单的介绍并分析了传统方法的优缺点。

（2） 根据邮件过滤系统的特性，对系统需要使用的 URL 过滤算法[5]。朴素贝叶斯算法、哈希查找法的优缺点以及在邮件过滤系统的适用性进行了分析比较。

（3） 本文重点介绍两个算法改进和一个应用。在 URL 过滤算法中，一是散列值比较：将邮件中大部分的 URL 换算成固定长度的数字与数据库中普通非法 URL 比较；二是相似度比较：如果邮件中某个 URL 与数据库中重点非法 URL 的某个 URL 有 95%的相似度，也认为是一个非法 URL。在最小风险贝叶斯算法中，采用概率值阈值[6]，当邮件判为垃圾的概率是判为正常的概率的倍数大于一个阈值时，才判为垃圾邮件。应用哈希表快速查找邮件关键词。

（4） 针对单一传统过滤技术容易漏判和误判垃圾邮件，设计了邮件多种方法过滤系统。该系统使用四大方法过滤邮件。一是利用重要用户名数据库确保重要发件人的邮件都为正常邮件，剩余的邮件再去重复，二是采用改进后的 URL 过滤算法过滤含有非法链接的垃圾邮件，三是基于最小风险贝叶斯方法过滤剩余的邮件，只有当邮件被判为垃圾邮件的概率是被判为正常邮件概率的 8 倍时，才将邮件判为垃圾邮件，

从而降低邮件误判为垃圾邮件的风险。四是用快速哈希[7]表在垃圾邮件中查找关键词，确保有职业特征的垃圾邮件重新判为正常邮件。

（5） 最后，对系统进行功能和性能测试，证明过滤系统达到设计要求，既能准确有效过滤垃圾邮件，又能降低正常邮件误判为垃圾邮件。

1.4 本文章节安排

本文的内容围绕邮件过滤系统的研究和实现展开。全文共分六章，具体内容安排如下：

第一章 介绍了本文的研究的背景和意义，最后说明各个章节的安排。

第二章 首先介绍电子邮件，然后介绍传统的过滤邮件的方法的优缺点。

第三章 首先邮件过滤系统的评判标准、重要用户数据库的建立过程以及重复邮件的判断条件。接着介绍了 URL 链接算法的改进，然后介绍了朴素贝叶斯算法和基于最小风险的贝叶斯算法，最后是哈希算法的应用。

第四章 设计了一个邮件过滤系统。先是系统介绍，然后是系统的五大模块设计，最后系统整体实现。

第五章 首先对邮件过滤系统的每个模块进行单元功能测试和系统整体功能测试，然后选取邮件过滤系统的一个模块进行性能测试。

第六章 结束语主要对全文的工作进行一个总结，并对后期工作进行展望。

第二章 邮件过滤方法的研究基础

在第一章介绍了邮件过滤的技术研究现状以及本文的内容安排。本章首先介绍邮件的一些理论基础，邮件的工作原理，还有邮件的协议。重点介绍了传统的邮件过滤方法以及这些方法的优点和缺点。

2.1 电子邮件简介

2.1.1 电子邮件的发送和接收原理

电子邮件在 Internet 上发送和接收的原理[8]。如同日常生活中邮寄普通包裹一样。当要寄一个普通包裹时，首先要找到任何一个有这项业务的邮局，在填写完邮编、地址、收件人名、发件人信息后包裹就被邮寄了出来。包裹送到了收件邮局，收件人必须到这个指定的邮件去取包裹。同样的道理，发送电子邮件是可以从任何一个服务器发送邮件，然而收件人只能到该指定的服务器取该封邮件。过程如图 2.1 所示。

普通邮件与电子邮件的传递过程

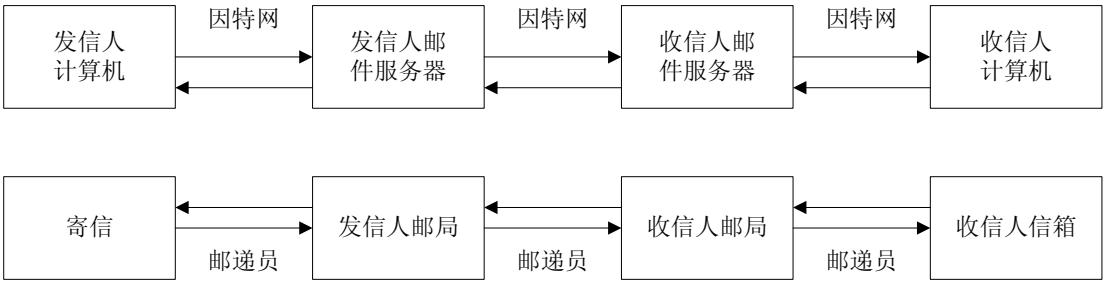


图 2.1 邮件传递过程

2.1.2 电子邮件系统组成

电子邮件地址的格式由三部分组成。第一部分“user”代表用户信箱的用户名，对于同一个邮件接收服务器来说，该用户名必须是唯一的；第二部分“@”是分隔符；第三部分是用户信箱的邮件接收服务器域名，用以标志其所在的位置。普通的域名后缀分为：com(商业) 、gov(政府)、mil(军事)、edu(教育)。

电子邮件系统通常由两个系统组成[9]：用户代理(User Agent)和消息传输代理

(Message Transfer Agent)。用户代理是让用户能阅读和发送电子邮件，而消息代理是将消息从发信方送到收信方。用户代理就是一个电子邮件阅读器，安装客户端。通过该阅读器，用户可以接受消息、撰写消息、发送消息。消息传输代理是一个在后台运行的程序进程，作用是准确地把消息从发送方传递到接受方。电子邮件系统项包含五个基本功能：撰写(write)是指创建消息和回信的过程；传输(transmission)指的是把消息从发信人处传递到收信人处；报告(report)必须告诉发信人该消息怎么样了；显示(display)收到的消息是必需的，这样人们才可以阅读他们的电子邮件；处理(process)是最后一步，它关心的是收件人在收到消息后如何处理消息，可能的处理方式包括：不阅读而直接丢弃消息、阅读后再丢弃、保存消息等等。

2.2 传统的邮件过滤方法

2.2.1 关键字符过滤法

基于关键字符的过滤，属于一种基于邮件内容的过滤。有些垃圾邮件，存在有一些特定词，来显示该邮件是垃圾邮件。因此，先可以从邮件的主题词和邮件的主体内容来查找这些特定词。为了提高查找速度，人们先把这些特定词，换算成哈希值存放在哈希表里。当一封新的邮件来的时候，查找邮件的特征词，将特征换成哈希值。然后，遍历邮件的所有哈希值，一旦邮件中有一个哈希值和哈希表里的哈希值相匹配时，该封邮件就当作垃圾邮件。使用这样的哈希表查找，提高查找速度，但是也增加了计算机的负担，浪费了一些存储。还有一种直接查找方式，设置一些关键字符，例如“军队国家化”为垃圾字符。读取邮件的内容，匹配邮件中的字符，凡是含有“军队国家化”的字样邮件就认为是垃圾邮件。这样方法可以过滤百分之六十的垃圾邮件，同时也会错误判断。有的时候还设置一个阈值，该关键词在邮件中出现的次数大于阈值时，才能把这封邮件当作垃圾邮件。基于关键字符的缺点是，寻找关键字是一件很费力的事情，关键词的规律也很不明显。有的邮件含某个分词是正常邮件，有的邮件含有相同的分词却是垃圾邮件。

2.2.2 基于白名单过滤

基于白名单的过滤，是一种基于邮件来源的过滤。建立好 IP 地址表，当发件人的 IP 地址存在白名单表里面的时候，判定邮件就是一封合法邮件。当邮件的 IP 地址不在表里面的时候，系统先给邮件发送方发送一个询问。合法的发送者能给出一个正确的回复，这样就把这个新的发送者的 IP 地址列入白名单里。垃圾邮件的发送者给不出正确的回复，该邮件就会被系统认定为垃圾邮件。除了

IP 地址，还可以建立别的白名单表，例如包含了一些发件人的邮箱地址。要是邮件头的发件人，能够和名单表的发送方匹配的话，该邮件就当作正常邮件,否则，判为垃圾邮件。例如：邮件发件人名称是“解放军某部政治部”，该邮件就是正常邮件。白名单过滤的优点是可以百分之百过滤垃圾邮件。它的缺点是给计算机带来额外的负担，当 IP 地址没有在名单的时候，要询问发送方，反馈一个正确的答复。有的时候，会把第一次通信的正常邮件，发送者给不了正确的答复，该邮件当作垃圾邮件处理了。另外，依据个人主观建立的白名单表，有时会造成大量合法邮件误判为垃圾邮件。

2.2.3 基于黑名单的过滤

黑名单过滤也是基于来源的过滤。先通过分析和收集当前的垃圾邮件源。如果该地址属于恶意或无意的垃圾邮件来源，就把这样的 IP 地址列入黑名单里。黑名单过滤过程是首先检查邮件头的发件人或者 IP 地址，如果属于黑名单的发件人或者 IP 地址，像包含“法轮功分子”就当垃圾邮件处理。

中国反垃圾邮件联盟[10] 也采用黑名单方式：CBL(中国垃圾邮件黑名单)主要面向中国国内的垃圾邮件情况，所甄选的黑名单地址也以国内的垃圾邮件反馈情况为主。但是有的垃圾邮件发送者，会仿照正常邮件发送的邮箱地址和 IP 地址，欺骗过滤系统，逃避检查，因此，这样技术会漏判很多垃圾邮件。

2.2.4 基于规则的过滤

经过大量训练垃圾邮件统计某一类邮件的特有的规律，制定规则来判读邮件的类别。当新的邮件来的时候，按照制定的规则匹配。当匹配一致的时候，该封邮件判为垃圾邮件，否则是正常邮件。寻找规则的过程也是一个归纳与总结的过程。发现一个规则，在一个垃圾邮件中，就推广到所有的邮件中。优点是，可以迅速过滤大量的同一规则的垃圾邮件。缺点是，规则不好找，要阅读大量的邮件才能总结规律。另外规则容易变化，人容易判定的规则，电脑有的时候不好识别这个规则。一般来说，可以从邮件头信息、特定的关键词、邮件的内容特征来分析。常用的基于规则过滤有以下几个方法：

1、Rough Set方法

Rough Set理论是由波兰数学家Pawley，在1982年提出来的。它是一种对决策支持进行近似性推理,用来处理含糊与不确定性问题的数据工具。它重新定义了知识的含义，将知识构成为某个领域的一个族集，从新的视角出发对知识进行了定义。知识被认为是一种将具体的或抽象的对象进行分类的能力, 知识构成了某一感兴趣领域中各种分类模式的一个族集。该理论是在机器学习理论的基础上，

采用一种不完整与不确定的知识，来学习归纳的理论方法。刘洋等[11]将Rough Set引入来进行邮件分类,在小规模的一个邮件样本上实验，达到80％左右的正确率。

2、Ripper方法

Ripper是William W.Cohen[13]提出的一种基于规则的方法。它比传统的规则方法速度更快、性能更高。Ripper算法通过学习训练集中的全部正例，逐渐地向一切初始值是空集的规则集中加入新规则，形成一个新的正向的规则集，再利用全部的反例不断地对规则集中。那些关键字加入约束条件，这样一个新生成的规则集，就可以做出决策。这样方法对基于邮件内容一般分类，效果还可以，但是仍然有许多缺陷，尤其过滤在线邮件。需要人工收集大量的规则，同时，垃圾邮件变化很快，这样需要花费很多的时间和精力在训练样本的建立规则集上。

2.2.5 基于 URL 意图检测过滤

垃圾邮件发送者总是想方设法逃避邮件过滤技术的检测。有些邮件，无论内容还是格式都和正常的邮件是一模一样的。从标题与正文的内容是无法断定是否为垃圾邮件。但是，这样的邮件的正文包含了一个 URL 地址，也就是网页链接地址，正是这个 URL 地址就指向链接了一个垃圾网站。基于 URL 意图的检测就是来检测，该邮件网页链接地址所链接的内容是否为垃圾网页，来判断该邮件是否为垃圾邮件。常用的方法是收集大量的非法 URL 存于数据库，把新的邮件中 URL 提取出来与数据库里的非法 URL 比较，如果是非法 URL，该邮件就是垃圾邮件。关于过滤非法 URL 邮件的算法改进，会在第三章 3.2.3 节有详细的研究。

2.2.6 基于统计的过滤

这种方式先要求对大量的垃圾邮件，经过阅读、统计、找出垃圾邮件明显特征。例如，查找邮件，是不是包含一些反动的、不健康的内容，就要找“台独”、“色情”这样的词。通过训练，不断自我学习来提高垃圾邮件的识别能力。常用的方法是基于贝叶斯的过滤:贝叶斯分类是一种基于概率统计的机器学习方法，把邮件看成是—种分类(正常邮件和垃圾邮件)问题。该方法认为正常邮件和垃圾邮件都有各自的特征词集，每一封邮件是其中特征词的某种组合。经过大量学习，找到垃圾邮件的特征词集，以及正常邮件的特征词集。再利用 Bayes 公式可以求得邮件属于垃圾邮件的概率。

1、 KNN 方法

该算法是一个理论上比较成熟的算法[14]，也是最简单的机器学习算法之一。该方法的思路是，如果一个样本在特征空间中的 K 个最相似的样本中大多数属于某一个类别，在该样本也属于这个类别。KNN 方法在文本分类中得到了较为

广泛的应用。并且将邻近该样本的各个邻居的属性平均值赋给该样本，让不同距离代表相应的权值。给定两点 X=(x_1,x_2,…x_m)和 Y=(y_1,y_2,…y_m)的欧几里德距离。

$$D(x,y)=\sqrt{\sum_{i=1}^{m}(x_i-y_i)^2} \qquad 式(2\text{-}1)$$

KNN 没有训练过程，分类时直接将待分类文本与训练集合中的每个文本进行比较，然后根据前 k 篇相似的文本得到新文本的类别。

2、Winnow方法

Winnow分类算法[15]是一种典型的线性分类器。线性分类器是透过线性组合来分类的。对象的特征表示为特征值，而在向量中则表示为特征向量。

输入的特征向量：实数向量$\vec{x}$，则输出的分数为：

$$y = f(\vec{w}\cdot\vec{x}) = f\left(\sum_j w_j x_j\right) \qquad 式(2\text{-}2)$$

$\vec{w}$是一个权重向量。它通过训练找到某个类别所有特征的权重，一个阈值θ。给定一个待分类的文本

$$x = (x_1, x_2, \dots, x_n) \qquad 式(2\text{-}3)$$

如果

$$w_1x_1 + w_2x_2 + \cdots + w_nx_n > \theta \qquad 式(2\text{-}4)$$

则属于该第一类，否则，属于第二类。

Winnow 算法不需要独立性假设这样的前提。Winnow 非常适合于含有大量无关特征的分类问题。Winnow采用在线学习的方法，每出现一个样本就反馈调整一次,随着训练集合的增大，能够不停更新分类器。

3、Bayes方法

贝叶斯定理[16](Bayes' theorem)，是研究一个概率论的问题。研究先验概率和条件概率的关系。具体来说，在B发生的前提下A发生的概率，与在A发生的前提下B发生的概率是不同的。贝叶斯定理就是讲叙这两者之间关系的。

作为一个规范的原理，贝叶斯定理对于所有概率的解释是有效的；然而，频率主义者和贝叶斯主义者，对于在应用中概率如何被赋值，有着不同的看法： 频率主义者根据随机事件发生的频率，或者总体样本里面的个数来赋值概率；贝叶斯主义者要根据未知的命题来赋值概率。一个结果就是，贝叶斯主义者有更多的机会使用贝叶斯定理。贝叶斯定理公式如下：

其中 $P(\frac{A}{B})$是　A发生在B发生前提下的概率。

$$p(\frac{A}{B}) = \frac{p(\frac{B}{A})\bullet p(A)}{p(B)} \qquad 式(2\text{-}5)$$

P(A)是A的发生概率，成为先验概率,计算这个概率不用考虑B发生的情况。

$P(\frac{A}{B})$是已知在B发生条件下，A发生的条件概率,是一个后验概率。

$P(\frac{B}{A})$是已知在A发生后，B发生的条件概率，在A发生的前提下B发生的一个后验概率。

P(B)是B的一个先验概率或边缘概率，也作标准化常量(normalized constant)。

在贝叶斯的基础上，人们提出了朴素贝叶斯。朴素贝叶斯的是贝叶斯的一个简化方式，假设条件就是，文本的所有特征属性之间是独立的，这样的话，计算的复杂度和和难度就降低了很多。朴素贝叶斯方法，被广泛用于垃圾邮件的识别。前提条件，是文本邮件中的出现特征单词，是一个个独立的。训练邮件样本，找出垃圾邮件的特征，利用这些特征，就可以区分垃圾邮件和正常邮件。

朴素贝叶斯应用到文本分类，是要做一些假设条件。要把每一篇文章，看成一个“袋子”的词。每一个词，都是独立于语境，也就是说每个词的产生与别的词是独立的。首先根据类别先验概率，选择一个类别，然后根据对应类别的词分布的条件概率，再推导出未知文本所属于该类别的概率。类别分类的前提，需要收集大量的垃圾邮件和合法邮件来进行样本训练，将训练的结果作为判读正常邮件和垃圾邮件的根据。文本可以分成两个类别C_{k}(k=1,2)。每一个文本 可以被一个n维单词集标注，而且这些词集是独立分布，例如文本d$(\omega_1,\omega_2,\omega_3,\ldots,\omega_{n,})$.对于一个给定的类别C，文本d的概率表示为：

$$P(\frac{d}{c})=\prod_{j=1}^{n}P(\frac{\omega_j}{C}) \qquad \text{式(2-6)}$$

计算文本属于某个类别的概率$P(\frac{C_k}{d})$时，利用贝叶斯概率公式，对于给定的d，属于第C_K(k=1，2，…，m)类的概率为：

$$\mathrm{P}(\frac{\mathrm{C_K}}{\mathrm{d}})=\frac{\mathrm{p(C_K)\times P(\frac{d}{C_K})}}{\mathrm{p(d)}} \qquad \text{式(2-7)}$$

在式(2-7)中$P(C_k)$是先验概率：

$$\mathrm{P(C_K)}=\frac{\mathrm{N_K}}{\mathrm{N}} \qquad \text{式(2-8)}$$

N表示为总的训练文本数量，N_k表示属于C_k类别的文本数量。

在式(2-7)中$\mathrm{P}(\frac{\mathrm{d}}{\mathrm{C_k}})$是条件概率：

$$P(\frac{d}{C_k})=\prod_{j=1}^{n} p(\frac{\omega_j}{C_k}) \quad \text{式(2-9)}$$

对于同一个文本$P(d)=P(\omega_1)\times P(\omega_2)\times ...\times p(\omega_n)$,假定正常邮件是类别$C_1$，垃圾邮件是类别$C_2$,如何区分文本是属于垃圾邮件还是正常邮件。根据贝叶斯分类法则，文本将归属概率最大那个类别。也就是说当$P(\frac{C_1}{d})>P(\frac{C_2}{d})$时，该文本属于正常邮件，属于$C_1$类，否则属于垃圾邮件，属于$C_2$类。

关于朴素贝叶斯方法模型的建立，在后面的第三章的3.3.2节会有详细的分析。

2.3 本章总结

本章主要介绍了邮件过滤方法的研究基础。文章首先介绍邮件的工作原理。然后就介绍了传统的邮件过滤方法。在几种邮件过滤方法中，最简单的垃圾邮件过滤方法是关键词过滤方法，但是只能过滤有限邮件，没有自我学习的能力。白名单和黑名单过滤方法，基于一张名片的过滤，每次过滤都要和名单比较。基于规则的过滤，需要大量的学习，自我训练和完善。基于 URL 意图检测，需要到链接的网页进行内容判断。基于统计的过滤，也是要训练大量的邮件样本，还要自我调整，才能达到一个好一点效果。

第三章 邮件过滤算法的研究

本章重点分析与改进两种算法：过滤 URL 算法、最小风险贝叶斯算法以及分析与应用哈希表在邮件中查找关键词。首先介绍了邮件过滤体统评价标准、重要用户数据库标准和去重复邮件标准。其次，分别重点研究与改进两个算法研究。在过滤 URL 算法中，一是散列值比较：将邮件中大部分的 URL 换算成固定长度的数字与数据库中普通非法 URL 比较；二是相似度比较：如果邮件中某个 URL 与数据库中重点非法 URL 的某个 URL 有 95%的相似度，也认为该 URL 是一个非法 URL。在最小风险贝叶斯算法中，采用概率值和阈值，当邮件判为垃圾的概率是判为正常的概率的倍数大于一个阈值时，才判为垃圾邮件，提高了过滤邮件准确率。接着是应用哈希表查找连续 5 个字符为一个关键词的邮件。最后，对本章总结。

3.1 邮件过滤的三个标准

本章中邮件过滤系统的三个标准是包含：邮件过滤评价标准、重要用户标准和去重复邮件标准。下面分别介绍。

3.1.1 邮件过滤的评价标准

用三个标准来评价邮件过滤模块的性能：准确率、查全率、F_1测试值。

通过这三个指标分析与设计出最优的贝叶斯过滤器。假定有n份邮件要经过过滤器进行判定分类。这n份邮件包含合法的邮件和垃圾邮件。

通过过滤器，就得出了四个结果：$n_{s\to s}$表示垃圾邮件被判为垃圾邮件的数量; $n_{s\to L}$表示垃圾邮件被评判为合法邮件的数量; $n_{L\to s}$为合法邮件被判为垃圾邮件的数量; $n_{L\to L}$表示合法邮件被判为合法邮件的数量。具体如下表3.1所示：

表3.1 邮件判断表

	系统判为垃圾	系统判为合法邮件
垃圾邮件	$n_{s\to s}$	$n_{s\to L}$
合法邮件	$n_{L\to s}$	$n_{L\to L}$

准确率[17]:系统正确判为垃圾的邮件数与被系统判为垃圾邮件数量的比值

数学公式表示如下:

$$\text{准确率(precision)}=\frac{n_{s\to s}}{n_{s\to s}+n_{L\to s}} \qquad \text{式(3-1)}$$

查全率[18]:系统正确判为垃圾的邮件数，与实际的垃圾邮件数量之间的比值,数学公式表示如下:

$$\text{查全率(recall)}=\frac{n_{s\to s}}{n_{s\to s}+n_{s\to L}} \qquad \text{式(3-2)}$$

简而言之，准确率为本身是垃圾邮件并且也被系统评判为垃圾的邮件数与被系统评判为垃圾邮件数的比值；查全率为本身是垃圾邮件并且也被系统评判为垃圾的邮件数与实际真正垃圾邮件数的比值。最理想的系统是$n_{s\to L}$和 $n_{L\to s}$使都为零，即准确率与查全率都为1；

理想的情况是准确率和查全率都为1，但是在实际中这种情况是不好实现的。假如调高判断为垃圾邮件的标准，$n_{s\to s}$变小，$n_{L\to s}$变小，是可以调高邮件的准确度，于此同时，$n_{s\to L}$变大，也就是更多的垃圾邮件会漏判为正常邮件，这样查全率会变小。

反过来，降低垃圾邮件的判断标准，$n_{s\to s}$变大 ，$n_{L\to s}$变大，也是说会有更有合法邮件会误判为垃圾邮件，准确率会下降。与此同时，$n_{s\to L}$变小，垃圾邮件判为合法邮件的数量会减小，查全率会提高。因此准确率和查全率是从两个不同方面来考虑这个问题，需要综合两者特点，提出了一个新的系统评估标准。用数学公式[18]表示为:

$$F_1\text{测试值}=\frac{\text{准确率}\times\text{查全率}\times 2}{\text{准确率}+\text{查全率}} \qquad \text{式(3-2)}$$

统计大部分邮件过滤系统的性能测试数据结果，当F_1测试值大于80%时，就能对邮件进行比较准确的过滤了。

3.1.2 邮件过滤重要用户标准

本文的重要用户指的是重要的发件人的邮箱地址。单位相关负责人有的时候要给每个成员群发一份邮件。为了防止邮件过滤系统，把这样的邮件识别为垃圾邮件，需要先把这样的邮件提前作为正常邮件处理。因此，规定只要是重要的发件人发过来的邮件，一律是作为正常邮件来处理。例如一份邮件，在邮件头包含“From: =?GBK?B?1u7Hws/y?= <qazniuhaoyi@163.com>”。该行里面都包含了一个发件人的邮件地址，先提取该份邮件的发件人的邮箱地址 qazniuhaoyi@163.com”。再核实该地址是否为重要的发件人。

例如“某部队首长的参谋”给全体人员发来的邮件，该封邮件直接当作正常邮件，不再作别的处理了。为了存储重要的发件人邮箱地址，需要建立一个数据库来存放重要的发件人邮箱地址。先建一个数据库里，再要建立一张表，表的内容包含重要的邮箱名。凡是重要的用户名发过来的邮件，直接作为正常邮件，不再进行垃圾邮件过滤处理。首先新建数据库名Mail.mdb，库内表名为：MailTable，表内字段名为：mailname(邮箱名)一个字段。然后在该字段下面添加重要的发件

人的邮箱地址了。 访问数据库使用ADO技术。先在程序中用#import语句，插入支持ADO的组件类型库(*.tlb)。该类型库可以作为执行程序或者动态链接的一部分被嵌程序本身中来实现。这种方法在结束时要关闭初始化的COM，可以用语句CoUnInitialize()来实现。最后就可以直接使用ADO的操作了，对数据库连接、访问、删除、插入、查询。也可以直接打开数据库里面的表格，手动添加、删除重要的邮箱地址。

3.1.3 邮件过滤去除重复标准

判读邮件重复的最简单方法是判读两份邮件是不是一模一样的。传统的方法是：采用哈希表把邮件通过哈希算法转换为哈希值，与前面的邮件的哈希值比较。如果与前面的邮件的哈希值比较是一样的，该邮件就重复邮件。要是与前面的所有邮件的哈希值比较是不一样的，把该封邮件的哈希值添加到哈希表。这样的做法是可以去除一部分重复邮件。但效果不明显。有些重复邮件，表面呈现出来的内容是一模一样的。但是里面的字符，可能是前一份邮件比后一份邮件多一个无关紧要字符。肉眼识别这两份邮件是重复邮件。但是换算为哈希值，两份邮件的哈希值是不一样的，他们被判为不同的邮件。

本文所使用的判读邮件重复的新标准。要判读两份邮件是不是重复邮件，要符合下面四个标准。发件人，邮件主题要求一致。两份邮件大小相差在3K以内。如果该邮件有附加的话，附加名要一致。满足这个四个条件，该封邮件就是一个重复邮件。为了迅速比较，建立一个结构体，包含四项：发件人，主题，邮件大小，附加名。具体实现，如下图3.1 所示：

```
1 struct Mail
2        {
3          CString   From;          // 发件人
4          CString   Subject;       // 主题
5          int       mailsize;      // 邮件大小
6          CString   attachment;    // 附件名
7          } mail;
```

图3.1 邮件结构体

先比较发件人，要是发件人一致的话，再比较主题，主题一致再比较邮件大小，大小之差要在3K以内的时候，再比较附件名，没有附件名的话，就把附件名这项值为空字符。四项都一致的时候，才判断新的邮件是重复邮件。

3.2 邮件过滤 URL 算法研究

垃圾邮件泛滥成灾，传统的反垃圾过滤技术是黑白名单过滤，基于关键字符过滤，基于规则过滤，都有各自的优缺点。

黑白名单过滤，是将已经知道的垃圾邮件的邮箱地址、IP 地址，存放在一个黑名单里面。把与邮件通信过的正常邮件的邮箱地址、IP 地址，存放在一个白名单里面。当新的邮件过来的时候，就进行判读，符合黑名单的要求的邮件，就是垃圾。对于白名单，符合要求白名单要求的判为正常邮件。这样方法需要不断维护更新黑白名单列表。优点是：对已经知道的黑白名单效果明显。同时这样的判断也存在缺点：垃圾发送者可以伪装发件人来逃避黑名单过滤；白名单有时也会把新发的邮件当垃圾邮件处理。

基于关键字符的邮件过滤，是查找该邮件里面有没有关键字符。要是有这些关键字符，就把该封邮件当作垃圾。优点是能够快速过滤含有关键字符的邮件。该方法的缺点是要预先准备大量的关键字符。垃圾发送者篡改了邮件里面关键字符了，该方法就没有效果了。

基于规则的过滤，先有一个学习的过程，首先进行大量的学习，找出正常邮件和垃圾邮件的不同规则。然后建立相应的函数。该方法的缺点是，需要通过训练大量的邮件，来找出其中的规则。邮件的规则是很难找的，垃圾邮件发送者可以用小伎俩，逃避规则过滤。

垃圾邮件发送者会想出各种各样的伎俩来逃避传统的一些方法的过滤，很大一部分网页格式的邮件是垃圾邮件。在垃圾邮件中有 URL 的邮件占了邮件的大部分。尤其是那些邮件大小比 10K 要小的邮件，有百分之七十多包含 URL。虽然一些合法邮件，也存在 URL，但是垃圾邮件存在 URL 比例是远远高于正常邮件比例的。

3.2.1 邮件具有 URL 的 HTML 格式特点

HTML(Hypertext Markup Language)格式，是用来描叙网页的一种语言。网页的本质就是 HTML，通过结合使用其他的 Web 技术(如：脚本语言、CGI、组件等)，可以创造出功能强大的网页。因而，HTML 是 Web 编程的基础，也就是说万维网是建立在超文本基础之上的。HTML 之所以称为超文本标记语言，是因为文本中包含了所谓“超级链接”URL。

利用这个特性大量的垃圾邮件采用 HTML 格式。垃圾邮件只所以大量选择选择 HTML 格式，有以下三个原因：

(1)HTML格式的邮件包含丰富的内容。垃圾邮件发送方，可以在HTM格式

的邮件插入大量的图片，背景音乐，影视文件，动画文件以及链接，让邮件的内容更加绚丽多彩。让基于特征的过滤防不慎防。

(2)HTML格式的邮件有反馈的功能和欺骗性。有些垃圾发送者在邮件中安装一个反馈的源代码，假如收件人去网上下载该邮件的时候，该反馈码被激活，发给垃圾邮件发送这一个信息，用来判断这个邮箱是否是一个有效的邮箱，这样垃圾邮件就能继续发送给该邮箱。还有的网页上面，显示的是一个安全正常的链接，只要用户点击该链接后，就会自动链接到别的非法的垃圾网站。

(3)HTML格式的邮件具有多样性、随机性，可以在邮件插入无关的字符，让邮件呈现出不同的邮件。例如在HTML格式的邮件中，添加一些无效字符，插入白色的文字在白色的背景下，虽然改变了邮件内容，但是呈现给用户看到邮件的文本内容是一样的。还有的一些邮件表面看起来的内容是不一样的，但是那个关键的URL是一样的，都链接到到了相同一个垃圾网站。这样基于内容的垃圾邮件过滤就无能为力。

在HTML垃圾邮件中，经常采用的一种形式是在邮件中加入URL的链接。这样的URL链接到的网页，才是垃圾发送者真正想让用户看的网页。

3.2.2 邮件过滤 URL 垃圾邮件普通方法

垃圾邮件通常会改变内容来逃避垃圾过滤软件。但是邮件中 URL 一般是不容易更改的。URL 是要通过授权才能获取的，同时 URL 指向的链接的网页，也需要成本和时间建立。因此查找含有 URL 的垃圾邮件，通过查找该类邮件的 URL，就可以找出这一类邮件。

虽然认为含有这样的 URL 就认为这封邮件是垃圾有时不一定完全准确，但是对于过滤同一类的垃圾邮件是很有效果的。可以通过收集大量垃圾邮件，从邮件中提取了大量的垃圾 URL 链接，存放在黑名单列表中。通过匹配、判断，该 URL 指向的内容是否垃圾邮件链接。该方法的优点是可以过滤大量的非法链接。但是也存在缺点：让邮件中的 URL 与数据库里面的 URL 逐个比较，运算是比较大的。另外，一些非法 URL 会变着花样来逃避检查，给比较增加难度。

3.2.3 邮件过滤 URL 比较算法改进

URL 比较算法的改进有两个优点：一是相似度比较：如果邮件中某个 URL 与数据库中重点非法 URL 的某个 URL 有一定的相似度，也认为该 URL 是一个非法 URL；二是散列值比较，提高比较速度，降低比较复杂度。将邮件中大部分的 URL 换算成固定长度的数字与数据库中普通非法 URL 比较。

1、相似度比较：从黑名单中查找 URL 是可以提高垃圾邮件识别率，但是需

要经常维护非法的 URL 数据库。为了躲避黑名单的检查，有的时候垃圾邮件发送者对一些出现率很高超链接，会随机改名 URL 中的某些不重要的字符，来逃避检查。例如从邮件 A 中，提取了一个非法的 URL ,判读邮件 A 为垃圾邮件。也从邮件 B 中也提取了一个 URL，但是该 URL 的内容和从邮件 A 中提取 URL 的末尾相差了一个空格字符。打开链接，发现两个链接指向同一个非法网页。但是机器识别的话，就不能判读邮件 B 的链接与邮件 A 的链接是同一个非法链接，因为两个链接相差了一个空格字符。

因此，普通的 URL 比较存在的缺点：是会漏掉一些非法的 URL,特别是一些出现率很高的 URL,会经常改变链接中的个别不重要的字符来逃避追查。针对那些出现率很高的非法链接,需要通过字符串相似度匹配来实现。一般是采用编辑距离表示两个字符串的差异。所谓的编辑距离是一个字符串变化多少次之后，才变成另外一个字符串。变化的最小次数就是编辑距离。字符串变化过程，通过删除、添加、替换字符来完成。例如:将字符“qszxdw”变成“qazxsw”字符串“qszxdw”需要变化两次，首先第二个字符从 s 变成 a，变成“qazxdw”。

其次是第五个字符从 d 变成 s，变成“qazxsw”。因此编辑距离是 2。也就说一个字符串变成另外一个字符，需要最小的变化操作数，就是编辑距离。一般是用动态规划技术[19]计算两个字符串的编辑距离。

采用矩阵的方法存储两个字符串的编辑距离 $d(s_1, s_2)$。

$$m[0,0] = 0 \quad \text{式(3-4)}$$

$$m[i,0] = i, i = 1,2,\dots,|s_1| \quad \text{式(3-5)}$$

$$m[0,j] = j, j = 1,2,\dots,|s_2| \quad \text{式(3-6)}$$

$$m[i,j] = \min(m[i-1,j-1] + p(s_1[i], s_2[j]), m[i-1,j] + 1, m[i,j-1] + 1) \quad \text{式(3-7)}$$

如果$s_1[i] = s_2[j]$，那么 p(s_1[i], s_2 [j])=0，否则$p(s_1[i], s_2[j]) = 1$。

这个算法的时间复杂度为$O(|s_1||s_2|)$。缺点是：时间复杂度大。

本文在编辑距离的基础上，基于 URL 字符串长度不长，进行了算法改进，提高了比较速度，降低运算的复杂度。

改进的方法思想如下：

假如有两个字符串L[x]和N[y]，分别只比较两个字符串相对应位置的字符。

当$L[i] \neq N[i]$是 1，否则是 0。然后统计相应字符之间的差异。改进后比较运算的复杂度是$O(\min(|L|,|N|))$。

为了得出两个不同的 URL 的差异，计算出相似度，本文采用一个新的比较法,建立一种基于 URL 距离的聚类模型。该模型具体思想如下：

首先定义两个 URL(x y)之间的距离，设 URL_x 的字符长度为L_x ， URL_y 的

字符长度为L_y，则URL_x 和 URL_y 的 URL 距离[20]为：

$$D(URL_x,URL_y)=scale\times\frac{(\sum_{i=0}^{L_x} URL_x[i]\neq URL_y[i])+|L_x-L_y|}{2\max(L_x,L_y)} \quad 式(3\text{-}8)$$

URL[i]是 URL 的第 i 个字符，$URL_x[i] \neq URL_y[i]$是一个布尔值，不同的时候为真值(1)，相同的时候为假值(0)。公式(3-8)得到的距离值是一个从 0 到 scale 之间的值。距离值越小，表示两个 URL 之间的关系越密切。距离小于一个阈值，就认为他们是同一个链接。

2、散列值比较：把出现率的很高的个别 URL,按照原始字符存于黑名单中。把其他的非法的 URL 字符串转换为数字比较大小。该散列函数把不相同长度的 URL 地址散列成固定长度的数字，这样比较就快速多了。对每一份新邮件先分析头部，看看是否是 HTML 格式，再提取 URL 地址，最后和黑名单的列表来比较。如果在黑名单里能找到的话，认为是垃圾邮件，就作为垃圾邮件处理。

如果没有找到，再与出现率很高的非法链接比较字符。当互相比较的结果小于一个阈值，该邮件就是垃圾邮件，当它们之间比较的结果大于一个阈值就当作正常邮件。没有找到的话，再换算成散列值与名单中的散列值比较改进算法后的垃圾邮件检查示如图3.2所示。

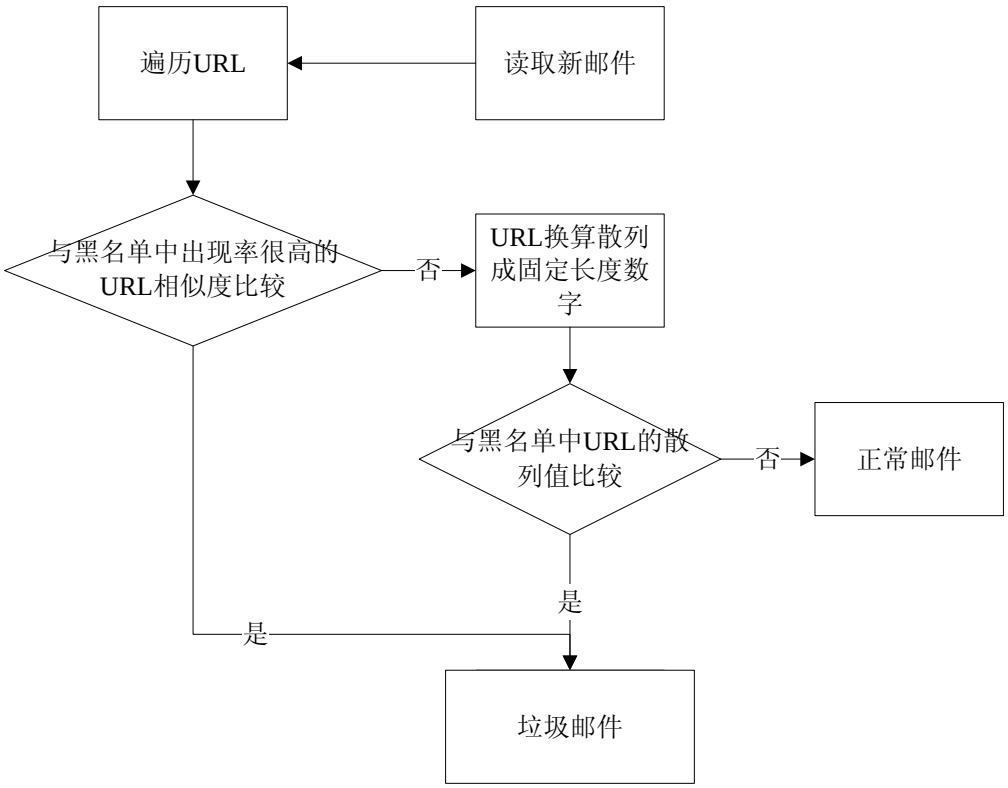


图3.2 改进后的URL示意图

3.3 邮件过滤贝叶斯算法研究

3.3.1 邮件贝叶斯的理论基础

前面的第二章的第四节，介绍了贝叶斯算法，现在，简单地回顾一下具体的内容。贝叶斯分类原理最初源自于概率论中的贝叶斯定理．该定理表示对未来某件事情发生的概率可以通过计算它已经发生过的频率来估计。朴素贝叶斯分类就是简化了各个样本直接的联系，假定每个样本都是独立的。当假设条件成立的时候，朴素贝叶斯的分辨率是很高的。举个例子，如果一种水果其具有红，圆，直径大于8厘米等特征，该水果可以被判定为是苹果。虽然，关于苹果的常识(先验概率))告诉大家，颜色和形状，没有什么关系。因而在统计上，可以认为是独立的。在邮件分类中，先训练一定数量的正常邮件和垃圾邮件形成两个分类样本，作为未知邮件的主要判读依据，运用到相应分类算法中去。

3.3.2 邮件过滤朴素贝叶斯算法研究

一般贝叶斯模型：假如一个文本有 n 个特性，那么每一个特性，就表示为：$(\omega_1, \omega_2, \ldots, \omega_n)$,那么一个父节点 C 类就可以表示为如图 3.3 所示。

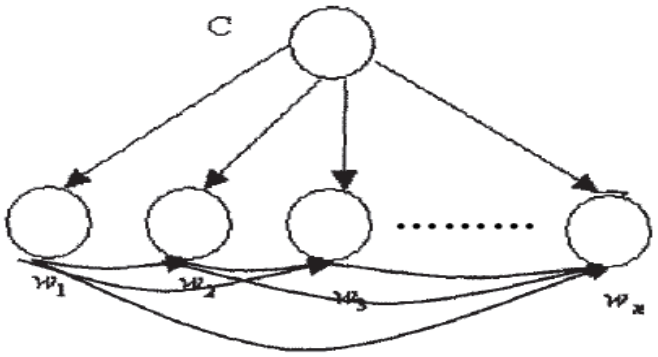


图 3.3 一般贝叶斯

在这个类中特征量之间是互相联系的。建立模型的话，运算量是很大的。因此人们提出朴素贝叶斯算法[21]。朴素贝叶斯算法的基本思想是每一个特征量之间，不存在任何关系，是一个个独立的部分。

朴素贝叶斯模型是基于贝叶斯模型的基础上建立起来。首先假定各个因素之间不存在任何关联，是一个完全独立的模型。经过这样简化之后，该模型具有独立性和较高的精确度，同时计算的复杂度，也大大减低了。在朴素贝叶斯模型中,文本中用 ω_i 表示第 i 个特征项，总共 n 个 ω 值。因此，对于一个给定的类变量 c，所有的属性 $\omega_1, \omega_2, \ldots, \omega_n$ 都要求条件独立类变量 C。也就是说，每一个属性变量之间关系相互独立，他们共同有唯一的一个父结点——类变量 C。结构如图 3.4

所示。

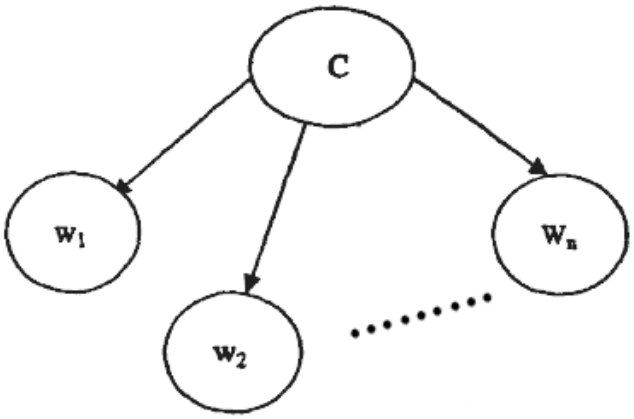


图 3.4 朴素贝叶斯

因为假设了所有属性变量，分别独立地作用于一个类变量，所以这样会大大降低运算的复杂度。朴素贝叶斯算法广泛应用于邮件过滤、数据挖掘中以及文本分类。

第二章第二节介绍了朴素贝叶斯的一些公式。可以把朴素贝叶斯算法应用到垃圾邮件的分类中去。在朴素贝叶斯模型中，对于一个文本 d 属于某个类别的概率，也就是计算条件概率 $P(\frac{C_k}{d})$，C_k 表示一个给定类变量。假定w_i表示第 i 个特征项，其特征属性之间相互独立。如上文所述，用C_k(k=1,2……m)表示类变量。d 表示一个任意 n 维向量的文档。基于文档中单词出现的概率相对独立的假设，例如文本d$(\omega_1,\omega_2,\omega_3,\ldots,\omega_{n,})$，对于一个给定的类别 C，文本 d 的概率表示为：类条件概率：

$$p(\frac{d}{C})=\prod_{j=1}^{n}P(\frac{\omega_j}{C}) \qquad \text{式(3-9)}$$

同样的文本 n 维向量的d$(\omega_1,\omega_2,\omega_3,\ldots,\omega_{n,})$，文本 d 的概率表示为：

$$P(d)=\prod_{j=1}^{n}P(\omega_j)\prod_{j=1}^{n}P(\frac{\omega_j}{C}) \qquad \text{式(3-10)}$$

对于同一片文档P(d)的值是表示不变的。

$P(C_k)$是先验概率P$(C_k)=\frac{N_k}{N}$， N 表示所有训练文本总的数量，N_k表示属于C_k类别的文本数量。常用的计算条件概率的算法，有两种，一种做多变量伯努利事件模型，另一种叫多项式事件模型。两者都是在在朴素贝叶斯假定的基础上，利用不同方法计算特征项权值，实现对文本邮件进行分类。

多变伯努利事件模型[22](Multi-variate Bernoull Event Model)，也是属于朴素

贝叶斯过滤模型，应用了布尔空间向量 1 和 0，就是判定特征项的权值为 0 或 1。例如，对于一个给定的文本向量d($\omega_1,\omega_2,\omega_3,\dots,\omega_{n,}$)。$\omega_j$ (j=1,2,…n)是向量 d 的第 j 个特征项。当该特征项在文档向量中出现，那么该特征项的权值为 1，如果特征项在文档向量中没有出现时，该特征项的权值就为 0。该方法的缺点是并没有统计这个特征项在文档向量中的出现总次数。

多项式事件模型(Multinomial Event Model)，也是一种朴素贝叶斯过滤器，基于一种数值型向量空间模型。统计文本向量 d 中的特征项的权值为特征项，以及出现在该文档向量中的次数。

然而该方法没有按照文档中的特征项的出现顺序来进行统计。当计算文档属于某一类的条件概率P($\frac{C_k}{d}$) 时，需要累计相乘所有出现在该文档中的特征项的概率值。因此假如特征项在文档中的出现次数称为事件的话，那么文档就是特征项的事件的集合。这种方法应用在传统语音识别的统计模型中，也常常用于文本分类。

相比多变量伯努利事件模型， 多项式事件模型，统计了在文档中所出现的单词次数。对于一些特定文章，某些单词会出现很多次数。例如部队的文本，就会更多出现某个部队单词，医学文本，会出现更多某个医学单词。为了更好地对文本分类，需要统计某个单词在文章中总共出现的次数。可以做一个朴素贝叶斯假设：每个单词在文档中出现的概率，与单词在文档中所处的位置，以及与别的特征项单词的是否出现无关，也与该文档的内容无关。

实践证明：当等待分类的文本比较长，训练词库比较大时，利用多项式事件模型来进行文本分类，就能够得到更高的精确度。

假定文本向量 d，共有 n 个数特征项d($\omega_1,\omega_2,\omega_3,\dots,\omega_{n,}$)。所以邮件分为正常邮件$C_1$类和垃圾邮件$C_2$类;假定$N_k$表示属于$C_{\mathrm{k}}$类的样本邮件数量，N 表示所有类别的邮件总数。属于正常邮件类和垃圾邮件类贝叶斯概率公式分别如下:

$$P(\frac{C_1}{d})=\frac{p(C_1)\times p(\frac{d}{C_1})}{p(d)} \qquad \text{式(3-11)}$$

$$P(\frac{C_2}{d})=\frac{p(C_2)\times p(\frac{d}{C_2})}{p(d)} \qquad \text{式(3-12)}$$

在同一个文本中 P(d)代表一个常量，先验概率P(C_1)和P(C_2)的值可下面公式就出来计算出:

$$\mathrm{P}(C_{\mathrm{k}})=(\frac{N_k}{N}) \qquad \text{式(3-13)}$$

由于特征向量之间是各自独立的：

$$P(\frac{d}{C_1}) = \prod_{j=1}^{n} p(\frac{\omega_j}{C_1}) \quad \text{式(3-14)}$$

$$P(\frac{d}{C_2}) = \prod_{j=1}^{n} p(\frac{\omega_j}{C_2}) \quad \text{式(3-15)}$$

利用公式 3-14，可以求出给定类C_k((k= 1 或 2) ,任意特征项w_j的条件概率，为了避免P($C_{k)}$等于零的情况，采用拉普阿斯概率估计。

$$P(\frac{\omega_i}{C_k}) = \frac{1+\sum_{j=1}^{|D|} N(\omega_i, d_j) \times P(\frac{C_k}{d_j})}{|V| + \sum_{j=1}^{|D|} N(\omega_i, d_j) \times P(\frac{C_k}{d_j})} \quad \text{式(3-16)}$$

公式中|D|表示在训练样本集中,训练样本的总数量，|v|是在特征向量空间中，特征项单词的总数。通过计算可以算出，文本 d 是属于垃圾邮件和正常邮件的概率。根据朴素贝叶斯分类规则，要是：P($\frac{C_1}{d}$) > P($\frac{C_2}{d}$),文本属于正常邮件，否则属于垃圾邮件。

在朴素贝叶斯算法分类的基础，使用 EM 算法朴素贝叶斯算法分类。EM 算法是用于一个最大似然估计，目标是找到某个模型参数 Θ 似的观测数据D_0的似然函数 P(D_0; Θ)达到最大值。

定义朴素贝叶斯模型中 EM 算法的迭代公式是

$$p(c_j; \Theta^T) = \frac{\sum_{i=1}^{|D|} P(\frac{c_j}{d_j}; \Theta^{T-1})}{|D|} \quad \text{式(3-17)}$$

迭代公式推导如下：

给定文档类别时文档的条件概率是

$$P(\frac{d_i}{c_j}; \Theta) = p(|d_i|)|d_i|! \prod_{i=1}^{|V|} \frac{p(\frac{\omega_t}{c_j}; \Theta)^{N_{ti}}}{N_{ti}!} \quad \text{式(3-18)}$$

按照朴素贝叶斯的条件，每一个文档和类别都是独立的，设$c_{(i)}$是文档 i 的类别标识。似然度函数写成

$$\prod_{i=1}^{|D|} P(\frac{d_i}{c_{(i)}}; \Theta) P(c_{(i)}; \Theta) = \prod_{i=1}^{|D|} P(|d_i|) d_i! \prod_{i=1}^{|V|} \frac{P(\frac{\omega_t}{c_{(i)}}; \Theta)^{N_{ti}}}{N_{ti}!} P(c_{(i)}; \Theta) \quad \text{式(3-19)}$$

式(3-19)两边同时取对数，得到完全似然度函数

$$\sum_{i=1}^{|D|}\sum_{t=1}^{|V|}N_{ti}\log P(\frac{\omega_t}{c_{(i)}};\Theta)+\sum_{i=1}^{|D|}\log P(c_{(i)};\Theta)+\Phi$$

$$=\sum_{i=1}^{|D|}\sum_{t=1}^{|V|}\sum_{k=1}^{|C|}h_{ik}N_{ti}\log P(\frac{\omega_t}{c_{(k)}};\Theta)+\sum_{i=1}^{|D|}\sum_{k=1}^{|C|}h_{ik}\log P(c_k;\Theta)+\Phi \qquad \text{式(3-20)}$$

当存在不可见类别标识，变量h_{ik}的值在Θ^{T-1}情况下条件期望，由式(3-20)得到期望完全对数似然度为

$$\sum_{i=1}^{|D|}\sum_{t=1}^{|V|}\sum_{k=1}^{|C|}P(\frac{c_k}{d_i};\Theta^{T-1})N_{ti}\log P(\frac{\omega_t}{c_k};\Theta)+\sum_{i=1}^{|D|}\sum_{k=1}^{|C|}P(\frac{c_k}{d_i};\Theta^{T-1})\log P(c_k;\Theta)+\Phi$$

$$=\sum_{i=1}^{|D|}\sum_{t=1}^{|V|}\sum_{k=1}^{|C|}P(\frac{c_k}{d_i};\Theta^{T-1})N_{ti}\log P(\frac{\omega_t}{c_k};\Theta)+\sum_{i=1}^{|D|}\sum_{k=1}^{|C|}P(\frac{c_k}{d_i};\Theta^{T-1})\log P(c_k;\Theta)+\Phi$$

$$+\lambda(1-\sum_{k=1}^{|c|}P(c_k;\Theta))+\sum_{t=1}^{|V|}\sum_{k=1}^{|C|}\lambda_{tk}(1-p(\frac{\omega_t}{c_j};\Theta)+\phi) \qquad \text{式(3-21)}$$

式(3-21)两边对 λ 求导，得到

$$\sum_{k=1}^{|c|}P(c_k;\Theta)=1 \qquad \text{式(3-22)}$$

式(3-21)两边对$P(c_k;\Theta)$求导，得到

$$\sum_{k=1}^{|D|}P\left(\frac{c_k}{d_i};\Theta^{T-1}\right)=\lambda P(c_k;\Theta) \qquad \text{式(3-23)}$$

对 k 求和并且由式(3-22)得到

$$\lambda=\sum_{i=1}^{|D|}\sum_{k=1}^{|C|}P\left(\frac{c_k}{d_i};\Theta\right)=|D| \qquad \text{式(3-24)}$$

将式(3-24)代入到式(3-23)中去，就得到迭代公式

$$\mathrm{p}\left(c_j;\Theta^{T}\right)=\frac{\sum_{i=1}^{|D|}P\left(\frac{c_j}{d_i};\Theta^{T-1}\right)}{|D|} \qquad \text{式(3-25)}$$

3.3.3 邮件过滤最小风险贝叶斯算法的改进

朴素贝叶斯算法是把文本中的每一个特征属性看成一个独立体，计算邮件属于垃圾邮件的概率和正常邮件的概率，当邮件属于垃圾概率大于属于正常概率，就判为垃圾邮件。在邮件的分类与过滤中，垃圾邮件与合法邮件具有不同的特性。一般情况下，不希望把垃圾邮件漏判为合法邮件，更不希望把合法邮件误判为垃圾邮件。因为，合法邮件被判为垃圾邮件会给用户带来更大的损失。为了减小风险，在原来的基础上，提出了基于最小风险的贝叶斯过滤方法，提高邮件过滤的

准确率。从两个方面，可以证明该算法可行性。

1、阈值方法

首先，引入了一个阈值[23]的概念，当邮件属于垃圾邮件的概率大于一个阈值时，才将该邮件判为垃圾邮件，提高邮件过滤的准确率。

一个邮件属于合法邮件和垃圾邮件的概率分别是$P\left(\frac{C_1}{d}\right)$和 $P(\frac{C_2}{d})$。

$$P(\frac{C_2}{d}) \geq M \qquad \text{式(3-26)}$$

当邮件属于垃圾邮件概率大于等于 M 时候，判该邮件为垃圾邮件。推导如下：

$$\frac{p(\frac{C_2}{d})}{p(\frac{C_1}{d})} \geq \lambda \qquad \text{式(3-27)}$$

只有该封邮件被判垃圾邮件的概率是是判为正常邮件的概率的λ倍时，才把这份邮件当作垃圾邮件，否则，还是把邮件当作正常邮件。把正常的邮件判为垃圾邮件，称作误判 L→S。垃圾邮件判为正常邮件，称作漏判 S→L。上面的公式，也可以说成是：S→L 是 L→S 的λ 倍时，才把邮件判为垃圾邮件。

一个邮件要么属于正常邮件要么属于垃圾邮件，因此属于正常邮件概率加上属于垃圾邮件的概率等于 1。

$$P\left(\frac{C_1}{d}\right) + P\left(\frac{C_2}{d}\right) = 1 \qquad \text{式(3-28)}$$

$$\frac{p(\frac{C_2}{d})}{p(\frac{C_1}{d})} = \frac{p(\frac{C_2}{d})}{1 - p(\frac{C_2}{d})} \geq \lambda \qquad \text{式(3-29)}$$

换算后就是：

$$P(\frac{C_2}{d}) \geq \frac{1}{1+\lambda} \qquad \text{式(3-30)}$$

$$M = \frac{1}{1+\lambda} \qquad \text{式(3-31)}$$

$$\lambda = \frac{1-M}{M} \qquad \text{式(3-32)}$$

换句话说：$P(\frac{C_2}{d}) \geq M$，把邮件 d 判为垃圾邮件。

当$P(\frac{C_2}{d}) \leq M$，把邮件 d 判为正常邮件，可以通过改变阈值 M 的大小，来取得最好的判断结果。

2、损失因子方法

先引入一个损失因子[24]。垃圾邮件判为正常的损失因子设为：1，那么正常邮件判为垃圾的邮件的损失因子就设为：λ，垃圾邮件判为垃圾邮件损失是 0，正常邮件判为正常邮件损失也是 0。

从最小风险的角度考虑，显然$1 \leq \lambda$，正常邮件判为垃圾的损失是要大。

表 3.2 邮件损失因子

实际情况	决策	损失
正常邮件	正常邮件	0
正常邮件	垃圾邮件	λ
垃圾邮件	正常邮件	1
垃圾邮件	垃圾邮件	0

从上面表 3.2 分析，可以看出正常邮件判为正常邮件和垃圾邮件判为垃圾是没有损失的。为了减少风险，要让正常邮件判为垃圾邮件带来的损失，要比垃圾邮件判为正常邮件带来的损失大。

一个邮件属于合法邮件和垃圾邮件的概率分别是$P(\frac{C_1}{d})$和$P(\frac{C_2}{d})$，

邮件 d 决策判为垃圾邮件带来的损失为：

$$P\left(\frac{C_2}{d}\right) \times \lambda + P\left(\frac{C_2}{d}\right) \times 0 = P\left(\frac{C_2}{d}\right) \times \lambda \quad \text{式(3-33)}$$

邮件 d 决策判为正常带来的损失为：

$$P\left(\frac{C_1}{d}\right) \times 0 + P\left(\frac{C_1}{d}\right) \times 1 = 1 - P\left(\frac{C_2}{d}\right) \quad \text{式(3-34)}$$

推导如下：因为邮件 d 判为垃圾邮件带来的损失，要比判为正常邮件的带来的损失要大，所以：

$$P(\frac{C_2}{d}) \times \lambda \geq P\left(\frac{C_1}{d}\right) \times 1 = 1 - P(\frac{C_2}{d}) \quad \text{式(3-35)}$$

也就是：

$$P\left(\frac{C_2}{d}\right) \geq \frac{1}{1+\lambda} \quad \text{式(3-36)}$$

$$N = \frac{1}{1+\lambda} \quad \text{式(3-37)}$$

$$\lambda = \frac{1-N}{N} \quad \text{式(3-38)}$$

就是说当$P(\frac{C_2}{d}) \geq N$时，邮件 d 判为垃圾邮件。

当$P(\frac{C_2}{d}) \leq N$时，邮件 d 判为正常邮件时。

从式(3-31)和式(3-37)可以得出结论 M=N,因此只要选择合适的λ,就能够得到一个好的结果。因此从损失因子的角度考虑，得出了和阈值法一样的结果。把这样的算法称为最小风险贝叶斯的邮件过滤算法。改进后最小风险贝叶斯算法在朴素贝叶斯的基础上，增加了邮件过滤的性能，同时减少了合法邮件被误判为垃圾邮件的风险。要过滤垃圾邮件，也会把合法邮件误判为垃圾邮件。人们一般倾向要尽量减少损失、减少风险。调整λ的大小，来得到一个最合理的值，减低损失与降低风险。

在第五章的性能测试中，试验证明当阈值$\lambda = 8$时，基于最小风险的贝叶斯算法在邮件过滤方面的准确率和F_1测试值是要比朴素贝叶斯算法要好。具体试验数据会在第五章详细介绍。

3.4 邮件过滤哈希算法研究

3.4.1 邮件过滤哈希算法可行性分析

哈希算法就是把任意长度的输入变成固定长度的输出，该过程是一个数字散列过程。输入的文本那有一点字符变化，得到的输出也会变化。把一些长字符换算成哈希值比较，可以提高比较速度。根据哈希函数换算成哈希值，在吧哈希值映射到一段存储地区间，记录关键词在表中映射像的存储位置，这样的表称为哈希表或散列，所得存储位置称为哈希地址或散列地址。有一些正常的邮件通过贝叶斯分类器，经常被误判为垃圾邮件。通过，大量邮件分析，发现这类邮件有一些特有的特征。这样，就需要从邮件内容，找到一些新的特征，来进一步判读，这封邮件是不是垃圾。譬如，发往营房科的一封邮件被误判为垃圾邮件，邮件的内容就涉及到一些特有特征“营房科助理员”。如果贝叶斯分类系统把它识别为垃圾，人工识别它的话又是一封正常合法邮件。为了减少这样的邮件，被误判为垃圾，需要再重新判定一次。

通过对正常邮件误判为垃圾邮件的再学习，发现这些邮件的内容有一些明显的标识。文件的开头，会体现发件人或者收件人的身份情况，像“汽车连司机班”，“营房科助理员”，“财务科助理员”，“干部科干事”，“司令部参谋”。有的时候文件开头没有这些内容，但是文件内容部分，也会体现这些关键词。这些单词的邮件通常是熟人之间的邮件。需要重新要把这些邮件重新当作正常邮件来对待。为了提高搜索效率，需要建立关键词哈希表，提高查找速度。

下面将对随机探测[25]的一组公式推导证明哈希查找能够提高速度。

先分析长度 m 的哈希表，包含有 n 个记录时的查找不成功的平均查找长度。也就是在这张表里填入第 n+1 个记录时所需要的比较次数的期望。有两个假定条件：(1)哈希函数是均匀的。 (2)处理冲突后产生的地址也是随机的。

假设p_i 表示前 i 个哈希地址都发生冲突概率； q_i表示要进行 i 次比较才找到一个“空位”的哈希地址的概率。则有：

$$\begin{aligned} p_1 &= \frac{n}{m} \\ p_2 &= \frac{n}{m} \times \frac{n-1}{m-1} \\ p_i &= \frac{n}{m} \times \frac{n-1}{m-1} \cdots \frac{n-i+1}{m-i+1} \\ p_n &= \frac{n}{m} \times \frac{n-1}{m-1} \cdots \frac{1}{m-n+1} \\ p_{n+1} &= 0 \end{aligned} \qquad \text{式(3-39)}$$

相对应的

$$\begin{aligned} q_1 &= 1 - \frac{n}{m} \\ q_2 &= \frac{n}{m} \times (1 - \frac{n-1}{m-1}) \\ q_i &= \frac{n}{m} \times \frac{n-1}{m-1} \cdots \frac{n-i+2}{m-i+2} (1 - \frac{n-i+1}{m-i+1}) \\ q_n &= \frac{n}{m} \times \frac{n-1}{m-1} \cdots \frac{2}{m-n+2} (1 - \frac{1}{m-n+1}) \\ q_{n+1} &= \frac{n}{m} \cdots \frac{1}{m-n+1} \end{aligned} \qquad \text{式(3-40)}$$

可见，在p_i和q_i之间存在关系式:

$$q_i = p_{i-1} - p_i \qquad \text{式(3-41)}$$

因此，长度为 m 的哈希表中已填有 n 个记录时，查找不成功的平均查找长度为：

$$\begin{aligned} U_n &= \sum_{i=1}^{n+1} q_i C_I = \sum_{I=1}^{n+1} (p_{i-1} - p_i) i \\ &= 1 + p_1 + p_2 + \cdots + p_n - (n+1) p_{n+1} \\ &= \frac{1}{1 - \frac{n}{m+1}} \end{aligned}$$

$$\approx \frac{1}{1-\alpha} \qquad \text{式(3-42)}$$

由于哈希表中所有的记录是按照先后顺序添加进去的，每查找一个记录所需要的比较次数期望值，刚好与添加该记录需要找到哈希地址所进行比较次数的期望值。因此，表长为 m,记录为数为 n 的哈希表，查找成功的平均查找长度为：

$$S_n = \sum_{i=1}^{n-1} P_i C_i = \sum_{i=0}^{n-1} P_i U_i \qquad \text{式(3-44)}$$

假设 n 个记录的查找概率相等。即$p_i = \frac{1}{n}$则有：

$$S_n = \sum_{i=1}^{n-1} U_i = \frac{1}{n}\sum_{i=0}^{n-1} \frac{1}{1-\frac{i}{m}} \qquad \text{式(3-44)}$$

$$\approx \frac{m}{n}\int_0^{\alpha} \frac{d_x}{1-x} \approx -\frac{1}{\alpha}\ln(1-\alpha)$$

从上面可以分析可以看出查找的平均长度与 n 无关，只与α 有关的函数。因此，无论 n 有多大，使用哈希查找，找到一个合适的因子总能将查找长度限定在一个范围内，从而能够提高查找速率。

3.4.2 邮件过滤哈希表的建立

针对部队单位的特点，设定的关键词都是五个连在一起的词，这样就可以大大提高查找速度。为了便于计算部队的关键词，像“财务科助理”，“司令部参谋”，“政治部干事”等等。需要建立一个哈希表，包含关键词如:“后勤部车队”、“后勤部助理”、 “干部科干事”、“军需部参谋”等等。因为都是 5 个连着词来做关键，先要把这些关键词哈希运算，再把运算的结果生产的哈希值，存放在哈希表里面。

具体的哈希表的建立过程：先将汉字用第一拼音字母表示，像“财务科助理”表示为“CWKZL”,政治部干事表示为“ZZBGS”。,再求出每一个字母的 ASCII 码之和。例如：“CWKZL”的的 ASCII 码之和为 395，395 除以 13 得余数为 5。具体的表示如表 3.3 所示。

表 3.3 简单哈希函数

key	CWKZL (财务科助理)	SLBCM (司令部参谋)	ZZBGS (政治部干事)	HQBCD (后勤部车队)	HQBZL (后勤部助理)	GBKGS (干部科干事)	ZXBCM (军需部参谋)
ASCII 码	395	369	400	354	385	366	388
余数	5	5	10	3	8	2	11

本文采用连地址法[25],将所有的关键词的为同义词的记录，存储在一个线性链表中。设定一个哈希函数产出的哈希地址在区间[0,m-1]上 ，那么就设立一个指针型向量 Chain ChainHash[m]。

每一个的分量的初始状态都是一个空指针，如果哈希地址为 i 的记录就插入到 ChainHash[m]的链表中。可以插在链表的头部、中部或者尾部，保持同义词在同一线性表中的按照关键字有序。本文中 m 去 13，关键词去各自模 13 的余数。

关键字为（财务科助理，司令部参谋，政治部干事，后勤部车队，后勤部助理， 干部科干事，军需部参谋）根据 ASII 码和可以转换为一组关键数组（395,369,400,354,385,366,388）。

按照哈希函数(H(key)=keyMOD13 和链地址法处理冲突构造所得哈希表如图 3.5 所示。

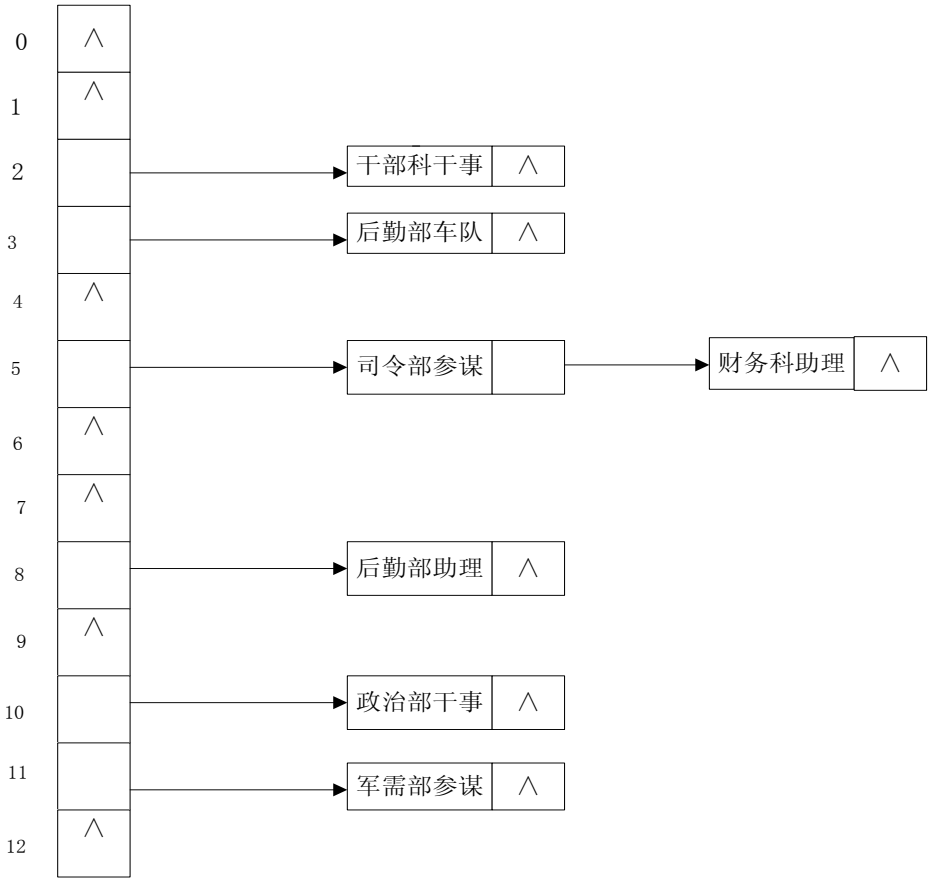


图 3.5 哈希生成表

面对一封新的垃圾邮件，先把邮件的内容按照五个词来计算哈希值。生产的哈希值，与哈希表里面的哈希值比较。如果哈希表里面，有该份邮件生产的哈希值，就认为这封邮件是正常。再把这份原来判为垃圾的邮件，重新划为正常邮件。

通过哈希值计算，可以进一步降低正常邮件被误判为垃圾邮件，进一步降低垃圾邮件的风险。

3.5 本章小结

本章对邮件过滤方法进行了研究和改进。首先，介绍了过滤系统的三个标准，评判标准、重要用户标准、去重复标准。其次，研究与改进过滤 URL 和朴素贝叶斯分类算法，在过滤非法的 URL 邮件中使用相似度比较和散列值比较，在朴素贝叶斯分类中使用 EM 算法以及阈值法和最小损失因子法，提高了过滤效果，将哈希查找应用于邮件关键词查找，确保有职业特征的邮件为正常邮件。最后对本章做了一个小结。

第四章 邮件过滤系统的设计与实现

邮件过滤系统设计的思想是既要过滤掉垃圾邮件，又要最大限度保护重要用户邮件。因此先建立重要用户数据库，确保重要发件人的邮件都为合法邮件。接着去除大量重复邮件。然后分两个层级过滤垃圾邮件，首先是基于 URL 链接，去除大量图片，然后通过贝叶斯算法分离器，按照内容特征过滤垃圾邮件。剩余的垃圾邮件，哈希表查看是否还有关键词，如果有的话就把该邮件当作正常邮件。

4.1 项目介绍

4.1.1 主要功能模块简介

实际情况是仅仅依靠某一项过滤技术，是很难有效的过滤全部的垃圾邮件。为了提高垃圾邮件过滤效果，需要采取多种过滤技术结合的方法，来过滤垃圾邮件。本课题主要是结合最新的垃圾邮件过滤技术，改进新的过滤算法，快速高效过滤垃圾邮件，同时要降低正常邮件误判为垃圾邮件的风险。该邮件过滤系统对进出服务器的邮件进行垃圾邮件过滤。

因为在过滤垃圾邮件的过程中，既要去除垃圾邮件又要最大限度地保留正常邮件。所以系统需要运用几个不同的过滤的手段。首先是要确保重要的用户发来的邮件收取，再去重复的邮件。处理垃圾邮件的时候，先是按照规则，基于 URL 垃圾邮件的过滤，先分出垃圾。剩下的邮件再按内容分类。首先在基于文本内容的贝叶斯过滤方法，进一步确定是否为垃圾邮件。对判为垃圾的邮件，还要看看是否含有哈希表里的关键词，如果有就把该邮件当作正常邮件。按照邮件的处理先后顺序，有以下几大模块：重要用户模块、去重复邮件模块、URL 模块、贝叶斯模块、哈希表查找模块 。按照处理顺序，分别介绍每个模块的功能。重要用户和去重复模块，是判读邮件是否为重要邮件和重复邮件。URL 模块是检查邮件中是否包含非法的超链接。贝叶斯模块，根据统计规则对邮件进行判断是否为垃圾邮件。哈希表查找模块是查找判定的垃圾邮件是否具有单位特征词。

该项目主要功能模块分为以下几个部分：

（1） 重要用户模块

检查邮件的发件人，再与数据库里的重要用户名比较。如果是重要用户发来的邮件就保留，全部当作正常邮件处理。

（2） 去重复邮件模块

与前面的邮件按照重复邮件标准四个条件比较，发件人、主题、附件

名要一致，邮件大小要在 3K 之内，如果满足条件就是重复邮件。

（3） 过滤 URL 模块

过滤含有非法 URL 的邮件。收集非法 URL，存入数据库里，对一些重要的经常出现的非法 URL 进行相似度比较，对其他的 URL 先换算成散列值再比较。

（4） 贝叶斯分类模块

该模块包括贝叶斯训练部分、维护部分、过滤部分以及维护部分，找出垃圾邮件和正常邮件的特征向量的概率值，计算新邮件的垃圾概率值，从而判断新邮件属于垃圾邮件的概率。

（5） 哈希表查找模块

该模块包括建立哈希表、哈希表查找。让有职业特征的垃圾邮件重新判为正常邮件。

（6） 其他功能模块

其他功能模块包括海量存储、邮件日志管理、邮件备份。限于篇幅不在介绍了。

邮件过滤系统的功能结构如图 4.1 所示。

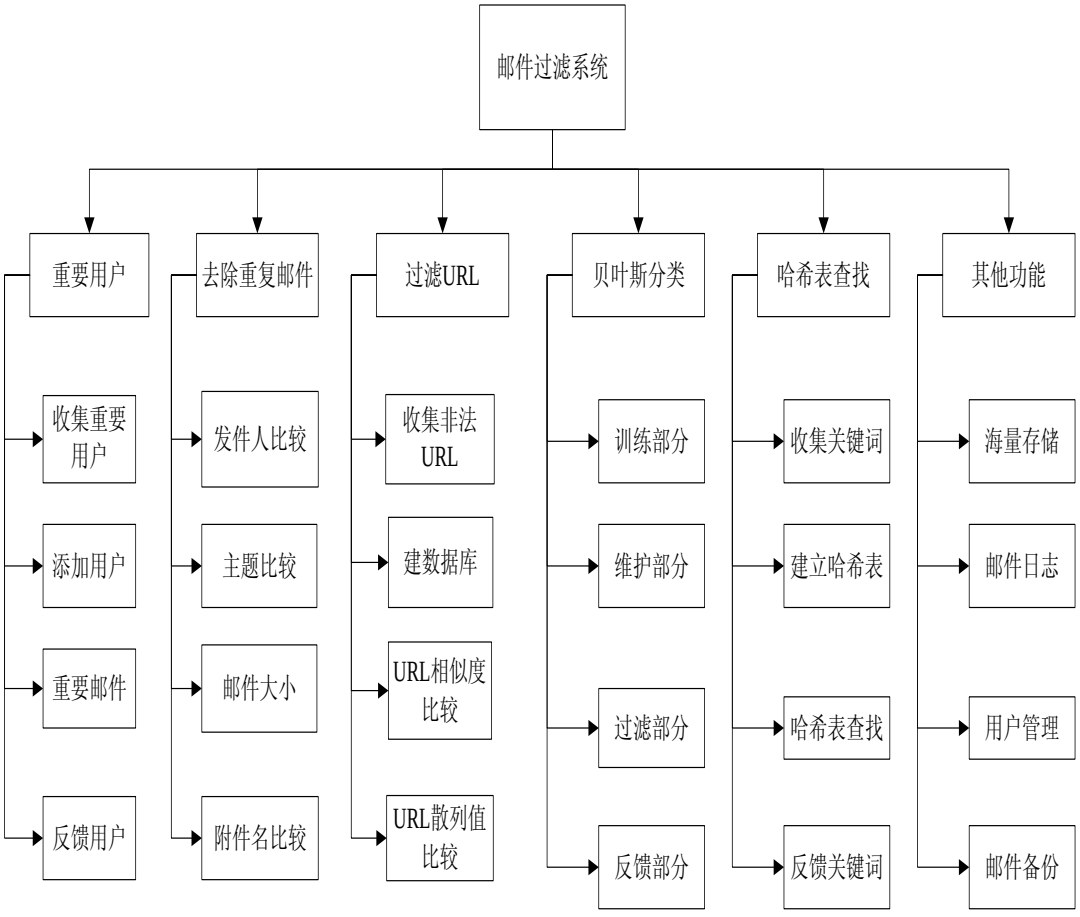


图 4.1 邮件过滤系统功能结构图

4.1.2 开发平台简介

该邮件过滤系统，是在 VC6.0 环境下开发的。VC6.0 是微软推出了一个编译器，该编译器是 C++语言集成开发环境。VC 是一个面向对象的可视化开发程序编译器，利用计算机图形学和方法，将抽象的数字、表格、逻辑结构用图形表示出来。用很直观图形表示原来抽象的编辑和操作，大大地简化了程序操作，提高了工作效率。

VC6.0 包含了工作区和客户区。具体编译界面如图 4.2 所示。

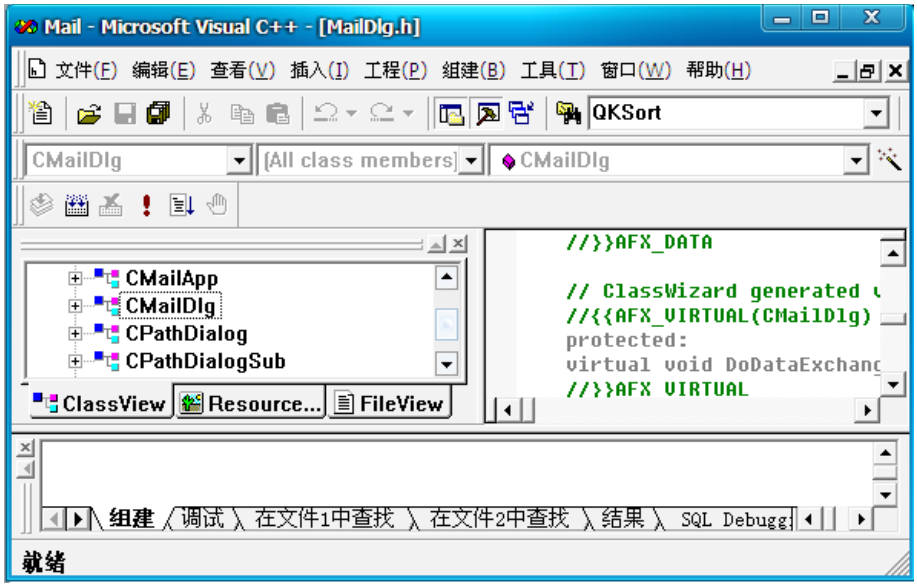


图 4.2 VC6.0 编译器主窗口

图 4.2 的左边是工作区包含 3 个视图。FileView(文件视图):显示所创建的工程,展开文件夹能看的工程包含的文件。ClassView(类视图):显示项目中定义的 C++类，展开文件夹显示工程中所定义的所有类，展开类可查看类的数据成员和成员函数以及全局变量、函数和类型定义。ResourceView(资源视图):显示项目中所包含的资源文件。图 4.2 的右边是客户区：是程序员代码编译，调试的地方。

程序中用到的数据库是 Access 数据库，是一个小型数据库。先打开 Access 应用程序，新建一个空的数据库。然后，在数据库里面，创建数据表。最后，设计表的里面的字段名称以及数据类型。最后，输入数据到表里。

本文通过 ADO 技术链接数据库,它是微软为数据链接开发的一个新的接口，能够面向 VC6.0 提供数据库快速链接。一个 ADO 模型包含 7 个对象和四个集合。7 个对象依次是：连接对象、命令对象、记录集对象、域对象、参数对象、属性对象、错误对象。四个集合分别是：域集合、参数集合、属性集合、错误集合。

4.2 邮件过滤系统整体流程的设计

按照处理邮件的先后顺序模块之间分别独立工作。由于处理的邮件数量比较大，专门需要一台电脑存储邮件。垃圾邮件处理后，再转发给用户。用杀毒软件，先对含有病毒的邮件，进行杀毒隔离。对处理垃圾邮件要备份，对一周前的垃圾邮件，要清理。由于使用几个方法处理垃圾邮件，本系统采用一个主函数调用，其他五个模块采用动态链接库实现。需要使用那一个功能的时候，就调用这个功能的动态链接库。为了及时过滤垃圾邮件，系统设定时间，每过 10 分钟，扫描一下服务器，看看有没有新的邮件。一旦发现新的邮件，就自动启用程序，进行邮件过滤。邮件过滤具体过程如下图 4.3 所示。

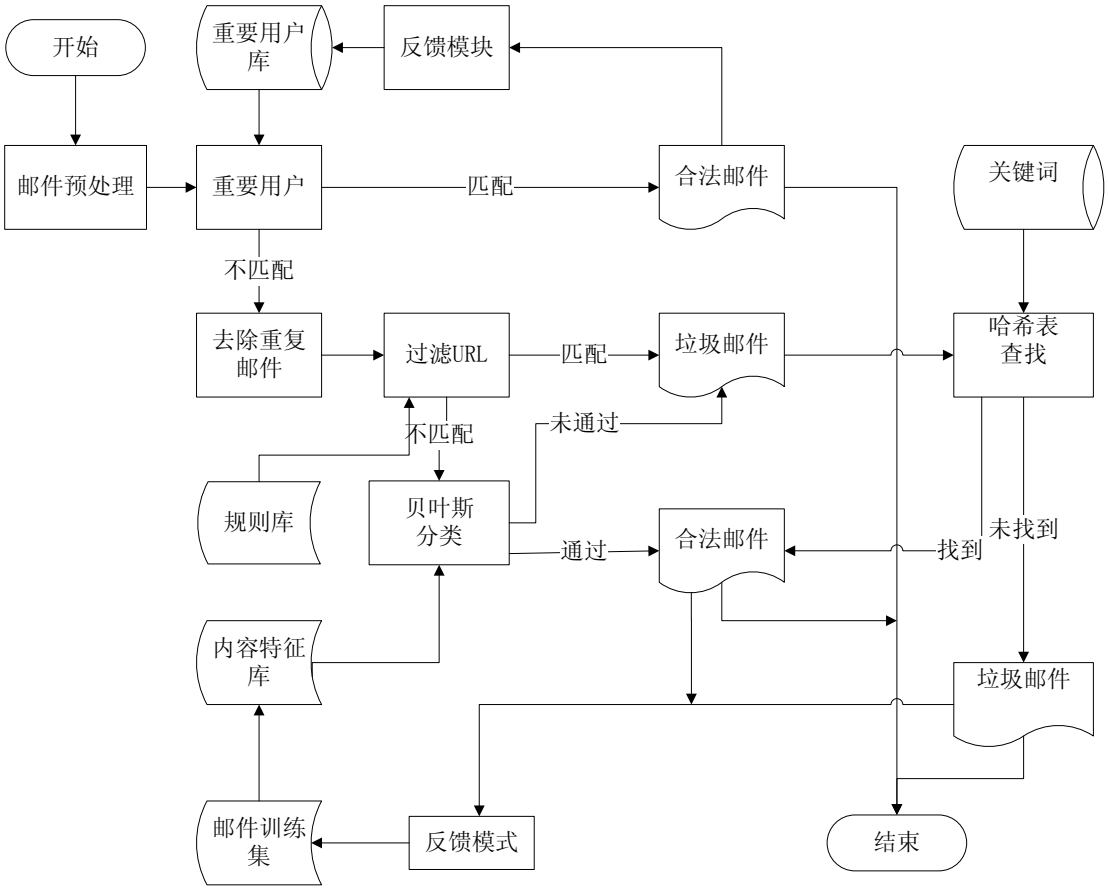


图 4.3 邮件过滤模型框图

邮件过滤基本步骤：

(1)先是收集重要的用户名，添加到重要用户数据库里面。同时在以后的收取邮件的过程中，可以添加和删除重要用户。重要用户的邮件，不需要处理，直接为正常邮件。

(2)收集一定量邮件垃圾样本，提取 URL 特征存放于规则库里面。

(3)选取一定数量邮件作为邮件训练样本，提取分类样本内容特征，存于内容特征库。训练邮件还可以自我学习，不断扩展。

(4)新邮件到达时先看看是不是重要用户的邮件，要是重要用户的邮件的话，直接作为正常邮件传递下去。假如不是重要邮件的用户，转到下一级别过滤第五步。

(5)先经过去除重复邮件，要是重复邮件的话就删除。如果不是重复邮件，经过基于 URL 垃圾邮件的规则过滤。符合垃圾邮件条件，作为垃圾邮件处理。当判断不出来时垃圾邮件，转到下一级别第六步过滤。

(6)按文本内容的贝叶斯分类进行过滤，基于贝叶斯分类和利用文本内容特征，来分类判别垃圾邮件和正常邮件。对判读为垃圾邮件，还需要查找一下哈希表的关键词，如果包含关键词，就把该份邮件当作正常邮件。

(7) 通过垃圾邮件和正常邮件的分类结果反馈学习更新特征库，从而更加优化过滤系统过滤能力。

系统主函数流程设计具体过程如图 4.4 所示。

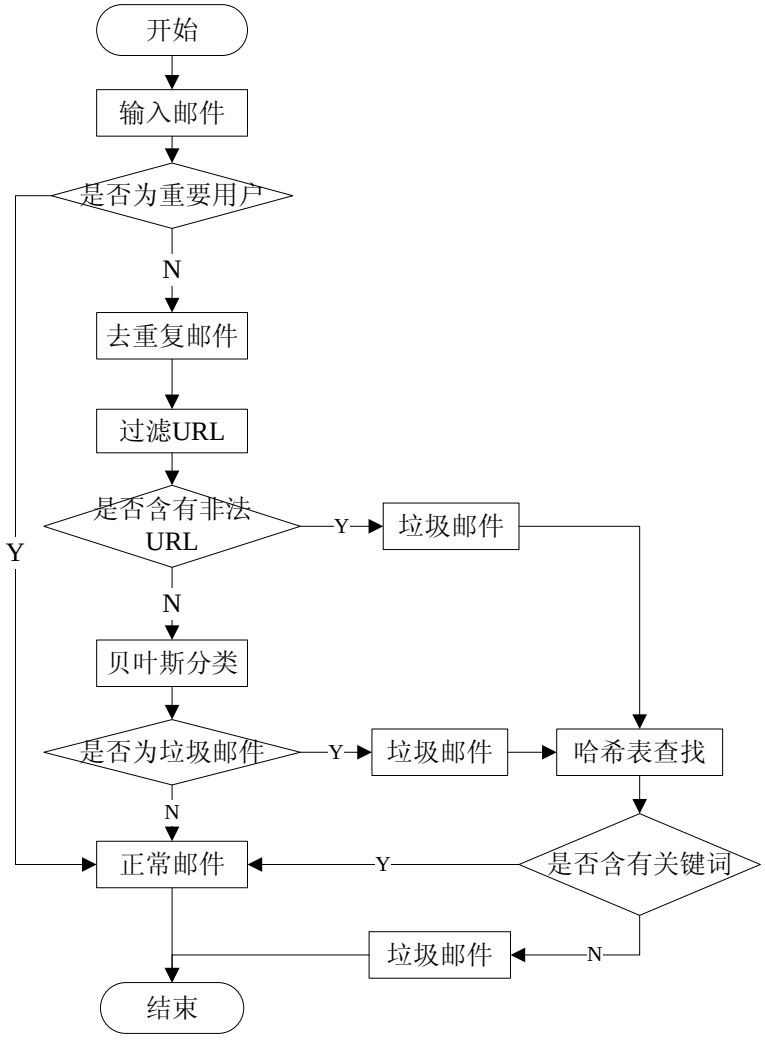


图 4.4 系统主函数流程图

4.3 邮件过滤系统的主要功能模块设计

4.3.1 邮件系统重要用户模块

基于重要用户的过滤的前提是收集大量重要用户。当用户数量比较少的时候，建立一个小文本，可以方便地操作，当用户比较多的时候，需要建立一个数据库访问。本章中基于重要用户的过滤是确保重要的发件人发送的邮件判为正常邮件。

一封邮件头包含：From: ?utf-8?B?SVQg5LiT5a62572R?=ews@ctocio.com.cn>，提取的 ews@ctocio.com.cn,作为新的邮件发件人的发送地址。然后提取新的发件人邮箱名与重要用户进行匹配，要是匹配成功，该邮件就当作正常邮件。要是匹配不成功，则进入下一轮，邮件过滤。

如果关键词重要用户比较多，可以将这些用户存放在数据库里面，通 ADO 操作 Access 数据库，来进行数据管理。

建立一个 Access 数据库，数据库的 mail.mdb,数据库里面的表名是 MailTable,表内字段名为 MailName(邮箱全名)。字段名：添加邮箱全名。如下图 4.3 所示。

程序连接数据库，然后读去表名 MailTable，再读取字段 MailName 下面的邮箱名。提取新的邮件的发件人。这样新的邮件里面的邮箱名，就和数据库的邮箱名比较。要是匹配的话，就是重要用户的邮箱。假如不匹配的话，邮件就进入下一轮的过滤。接着就是要去除重复邮件。

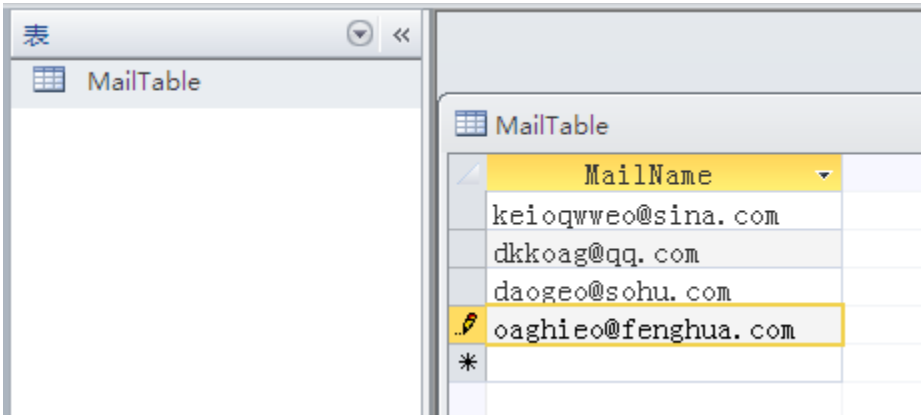


图 4.3 数据库存储

当用户少的时候，只需要用一个小的文本存取重要的邮件地址。这样运算速度更加快些。小文本名为 mainame.txt 存于 d 盘的跟目录下，文本里面存储邮件的格式，直接输入邮箱名全称，每一个邮箱名之间用回车换行分开。

提取新的邮件发件人邮箱地址，与文本中的每一个用户比较，要是文本了有这个用户的话，就把这份新的邮件当作重要邮件。

为了得到小文本里面的关键用户，先定义全局变量：CString key [2000];

从小文本文件，提取的重要邮箱名，算法如图 4.4 所示。

```
输入：一个文本文件 key.txt   /*存放重用用户名的文本*/
输出：CString key [2000];    /*提取关键用户名存入 key{2000}里面*/
1     int length=keyfile.GetLength();      // 取得文本文件长度
2     int i=0,p=0,j=0;
4     char   ucBufptr[100000];
5     memset(ucBufptr,0,length);          // 内存清零
6     keyfile.Read(ucBufptr,length);
7     keyfile.SeekToBegin();              //指针指向文件开头
8     for(i=0,p=0,j=0;p<length;)
9     {
10      j=0;
11      while((*(ucBufptr+p)!=';')&&(p<length))
12                              //循环没有找到分号(;)字符小于于文件长度
13     {
14       key[i].Insert(j,*(ucBufptr+p)); //提取重要用户邮箱,存于 key[i]
15       j++;
16       p++;
17    }
18       key[i].Insert(j,'\0');
19       j=0;
20       i++;
21       p=p+3;                                 //跳过(；回车换行)三个符号
22    }
```

图 4.4 提取关键用户名算法

4.3.2 邮件系统去重复邮件模块

该模块实现了对邮件去重复。将该邮件与前面所有邮件比较四个条件，即发件人，主题，大小差值在 3K 以内，四项都一致的话，判读该邮件是重复邮件了，否则将该邮件的四个条件加入比较队列中。依次比较四个条件，当一个条件不符合的时候，后面的条件就不用比较了，就判定该邮件是重复邮件。如果，该邮件不是重复邮件，该邮件的四个条件，添加到结构体里，作为后面邮件判断重复的条件。具体比较流程如图 4.5 所示。具体比较重复邮件算法如图 4.6 所示。

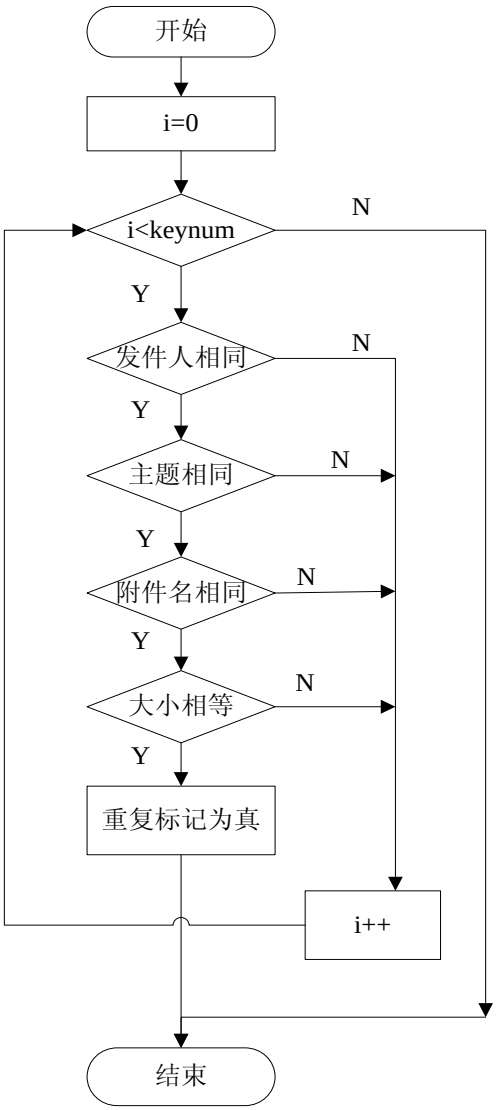


图 4.5 重复邮件的四个条件比较流程图

```
    输入：bool    biaoji              /* biaoji   为重复邮件的标记*/
    输出：bool biaoji
1   for(i=0;i<keynum;i++)
2   {if(strcmp(mail.From,key[i].From)==0)                    //发件人比较
3  {if(strcmp(mail.Subject,key[i].Subject)==0)               //邮件主题比较
4  {if(strcmp(mail.attachment,key[i].attachment)==0)        //邮件附件名比较
6                  {if(abs(mail.mailsize-key[i].mailsize)<=3)//邮件大小要在 3K 以内
7                   { biaoji=true;    break;}                //该邮件是重复邮件，标记为真
8                    }}}}}
9   if(biaoji==false)                                        //如果该邮件不是重复邮件
10    keynum++; i=keynum；
```

图 4.6 去重复邮件比较算法

4.3.3 邮件系统过滤 URL 模块

第三章已经介绍了垃圾邮件发送者大量使用 HTML 格式，据统计超过 80%的垃圾邮件都采用的 HTML 格式。通过对大量邮件的比对分析，会发现在垃圾邮件中存在超链接的比例，要比在合法邮件中存在超链接比例要高得多。因为大多数垃圾邮件包含超链接，并且这些超链接具有一定的规律，所以用超链接距离(或 URL 距离)来表示超链接的相似度。

目前有很多垃圾邮件的标题和主体内容都与合法邮件是一模一样的，不同的就是主体部分有个 URL 地址，这个 URL 地址链接的内容就是垃圾内容。URL 的类型很多，主要有表 4.1 所示 7 种类型：

表 4.1 URL 类型

URL 类型	示例
Web	http://***
Email	mailto:***
Related link	<link ***>
gopher	Gopher://***
Hyperlink	<a href ***>…
FTP	ftp://***
gopher	Gopher://***

注：表中的字符串*** 代表 URL 匹配的模式

第 3.2.1 节曾经提到有些 URL 频繁出现在垃圾邮件中，为了躲避软件过滤，垃圾发送者会在 URL 改变个别的字符，来逃避检查。对此，就提出了建立一种基于 URL 距离的聚类模型，来计算出该链接和黑名单中的出现率很高的链接之间的距离大小。如果当距离小于一个阈值时，就把该链接作为一个非法的链接来处理。如果距离大于阈值时，就认为该链接是非法链接。

当一份新的邮件过来的时候，先判读该邮件是否是 HTML 格式的邮件，然后再查找是否含有非法的 URL 链接。首先，判读该邮件是否是经常出现网页地址，该判读是过程是一个相似度比较，如果是就是垃圾邮件。如果不是经常出现网页地址，就先换算成散列值比较。因此，为了快速查找，要把大量其他的非法 URL 字符串转换为数字存放在黑名单。该散列函数[26]把长度不一样的 URL 地址散列成固定长度的数字，这样比较就快速多了。对每一份新邮件分析头部，先看看是否是 HTML 格式，在提取 URL 地址，然后把 URL 散列成固定的数字与黑名单中散列值来比较。具体程序流程如图 4.7 所示。

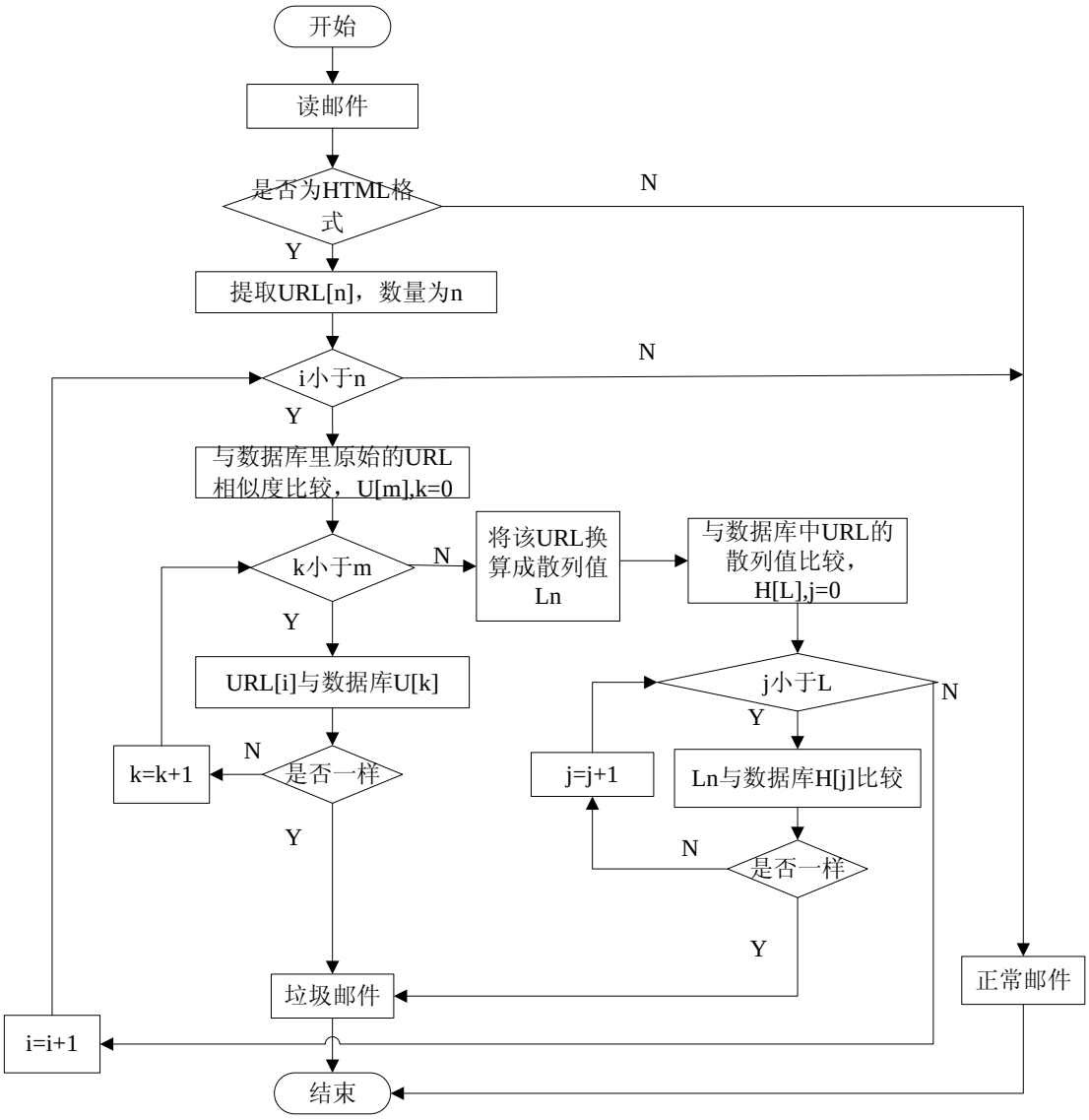


图 4.7 URL 查找程序流程图

第 3.2.3 节用一个数学模型计算出了相似度距离。定义两个 URL(x y)之间的距离，设 URL_x 的字符度为L_x ，URL_y 的字符长度为L_y ，则URL_x 和 URL_y 的 URL 距离为：

$$D(URL_x,URL_y)=scale\times\frac{(\sum_{i=0}^{L_x}URL_x[i]\neq URL_y[i])+|L_x-L_y|}{2\max(L_x,L_y)} \qquad 式(4\text{-}1)$$

URL[i]是 URL 的第 i 个字符，$URL_x[i] \neq URL_y[i]$是一个布尔值，不同的时候为真值(1)，相同的时候为假值(0)。公式(4-1)得到的距离值是一个从 0 到 scale 之间的值。距离值越小，表示两个 URL 之间的关系越密切。距离小于一个阈值，就认为他们是同一个链接。对于这些经常出现的非法链接，就先按照相似度比较，如果比较相似度达到要求，就是非法链接，具体实现的算法如图 4.8 所示。

```
输入：string a,string b //两个 URL 进行相似度比较
输出：int len
1 int find_max_string(string a,string b)  //两个字符串，取两者之间最长的字符串的长度
2  {
3      int x=a.length();
4      int y=b.length();
5      if(x>y)
6         return x;                          // x 长度大于 y 长度，就取的 x 长度
7      else
8         return y;                          //否则，x 小于 y 的长度，就取 y 长度
9  }
10 int find_minus_string(string a,string b) //求取字符串之间的绝对值长度
11 {
12     int x=a.length();
13     int y=b.length();
14     return abs(x-y);                    // 取两个字符串之差的绝对值
15 }
16 int add_string(string a,string b) //  依次比较字符串的字符的值，要是相同的话就加 1
17 {
18     int sum=0;
19     for(int i=0;i<a.length();i++)
20     {
21         if(a[i]==b[i])                  //比较两个字符串对应位置的字符
22           sum+=1; //如果两个字符串中，相应的两个字符是一样的话，为真加 1
23     }
24     return sum;                         //返回累计加的值
25 }
26 int length(string a,string b)     // 返回数学公式的值
27 {
28     int len=(find_minus_string(a,b)+add_string(a,b))/(2*find_max_string(a,b));
29     return len;                         // 返回计算值
30 }
```

图 4.8 URL 相似度比较算法

其他的非法链接换算成散列值比较，散列函数实现算法如图 4.9 所示。

```
输入：char *url ,int size                /*URL 链接和长度*/
输出：int n                              /*返回固定的长度数字*/
1 int Http(char *url ,int size)          // 散列函数，把不等长的 URL 散列成固定长度数字
2  {
3      unsigned int n=0;                          //初值 n 等于 0
4      char* a=(char* )&n;                        //将 n 首地址传给 a
5      for(int   i= 0;   i< strlen(url);   i+ +)  //取得了 url 的长度
6      a[i%4]^=url[i];              // i 模 4 后，然后再与 url 中的字符异或，结果再赋给 a
7      return n%size                              //返回 n 模 size 后的值
8  }
```

图 4.9 URL 散列值算法

4.3.4 邮件系统贝叶斯分类模块

1、训练部分实现

把分好的正常邮件和垃圾邮件，分别作为样本进行训练，找出垃圾邮件的特征概率，为新的邮件作为判读依据。

训练邮件需要用到的数据结构表如表 4.2、表 4.3、表 4.4 所示。

(1)文档单词集(wendangji)。该数据表格，用来存放，训练文档中的，特性单词，依据特征单子在文档中出现的次数。

表 4.2 文档单词集(wendangji)

序号	类型	名称	名称意义
1	intege	Mail_bh	邮件编号
2	string	Tezhen_name	特征单词
3	intege	Tezhen_num	特征单词在文本出现次数

(2)特征项单词表(tezhendanzibiao)。该表包含了特征向量中全部的单词。统计不同类别单词，以及属于该类别的特征单词出现的次数。

表 4.3 特征单词表集(tezhendanzibiao)

序号	类型	名称	名称意义
1	intege	Leibie_bh	类别名称
2	string	Tezhen_name	特征单词
3	intege	Tezhen_num	特征单词在类中的出现的次数

(3)训练文档表(xunlianwendang)。该表用来存储训练邮件的文档属性。内容包括文件名称、对应的文件类以及该文件所属的类别。

表 4.4 训练文档表(xunlianwendang)

序号	类型	名称	名称意义
1	string	Mail_name	文件名称
2	ntege	Mail_bh	文件编号
3	ntege	Mail_class	文件所属类别

模块分为四步实现：(1)读取邮件，采集邮件的过程。(2)邮件处理，先分割邮件，然后处理邮件头，处理邮件体。(3)提取关键词，计算概率。(4)更新数据库。具体实现流程图如图 4.10 所示。

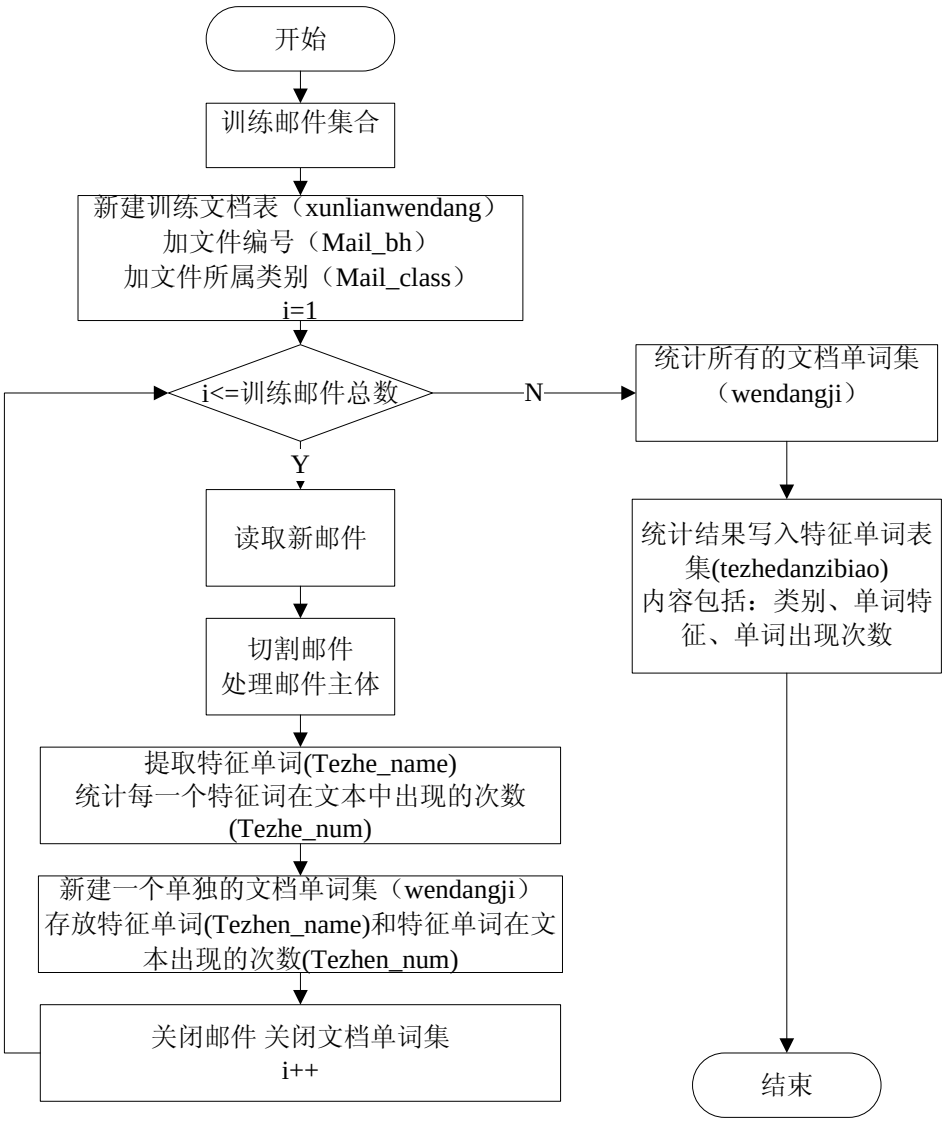


图 4.10 样本邮件的训练的流程图

2、分类部分实现

该模块是贝叶斯分类的重要部分，当一份新的邮件过来的时候，先处理邮件，找出邮件的特征单词，对照训练表的结果，再判别该封邮件的类别。

(1)待分类的单词表(daifenlei)。存储需要分类的邮件的特征单词出现的次数。

表4.5 待分类的单词表(daifenlei)

序号	类型	名称	名称意义
1	string	Tezhen_name	特征单词
2	intege	Tezhen_num	待分类的文档中出现的次数

分类模块是贝叶斯分类的核心部分，该部分的实现也是分为四个步骤：

(1)扫描新邮件。(2)提取新邮件的特征。(3)结合特征单词集计算邮件联合概率。根据给定的阈值判读邮件属于正常邮件还是属于垃圾邮件。(4)反馈学习。

贝叶斯分类模块的分类的具体实现的流程图如图 4.11 所示。

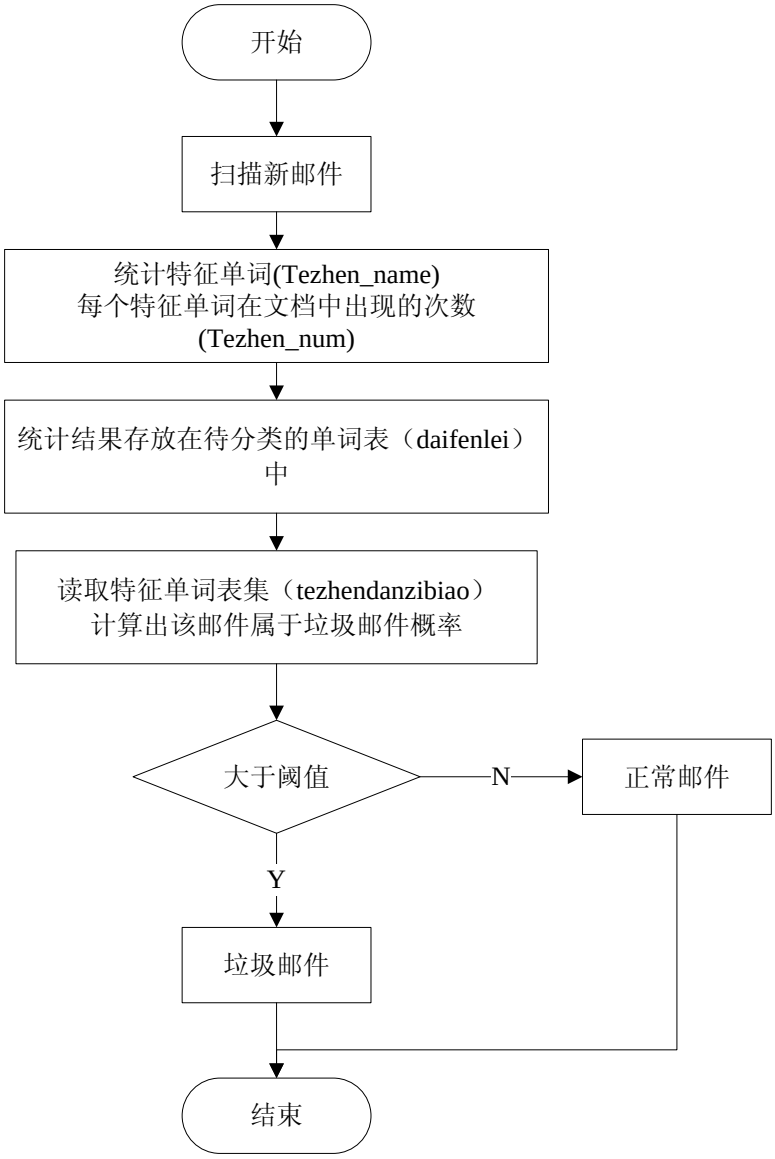


图 4.11 贝叶斯邮件分类流程图

4.3.5 邮件系统哈希表查找模块

针对部队单位的特征，就需要先建立一张特殊的关键词表格，在把关键词转

化成哈希值，存放在哈希表里。单位关键词表格如下所示：

表 4.6 关键词表

序号	关键词名称
1	政治部干事
2	司令部参谋
3	后勤部助理
4	军需部参谋
5	财务科助理
6	后勤部车队
7	干部科干事

把一封内容含有“后勤部助理”的邮件放入系统，通过哈希表查询该邮件是正常邮件。另外还可以继续在误判为垃圾邮件里面，找出新的邮件特征，进一步减低邮件被误判为垃圾邮件概率。哈希表查找流程如图 4.12 所示。

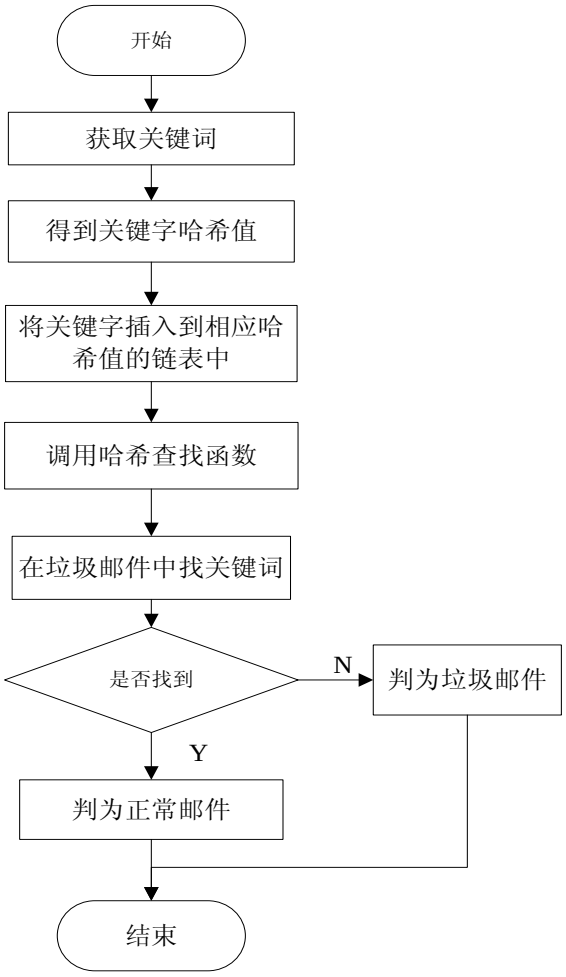


图 4.12 哈希查找的主流程图

在本文中需要把关键词的哈希值存放在哈希表里面，再通过哈希表查找。使用了散列技术后，查找关键字不需要比较，就可以获得记录的存储位置。所谓的散列技术，就是在记录的存储位置和关键字建立一个确定对应关系，每一个关键字对应一个存储位置。因此把这样一种对应的关系成为散列函数，也称为哈希函

数。采用散列技术将记录存储在连续的存储区，这样建立起来的表(也可以说是连续存储空间)，称为散列表或者哈希表(Hash table)。在第三章，采用连地址处理方法，插入关键词到哈希表里。具体的实现算法如图 4.13 所示。

```
输入：int m                    /*要插入的关键词*/
输出：struct keyNum*head       /*插入到链表 head */
1       p0=temp;               //要插入的节点（值为 m);
2     if(head==NULL)           //1,原来的链表为空，插入到 head 后
3      {
4           head=p0;
5           p0->next=NULL;
6      }
7      else//原来的链表不为空
8      {
9           while((p0->key>p1->key)&&(p1->next!=NULL))//移动到适当位置
10          {
11              p2=p1;
12              p1=p1->next;
13          }
14          if(p0->key<=p1->key)
15          {
16              if(head==p1)head=p0; //2，插入到第一个节点之前
17              else p2->next=p0;      //3,插入到 p2 指向的节点之后
18              p0->next=p1;
19          }
20          else                       //4,插入到结尾处
21          {
22              p1->next=p0;
23              p0->next=NULL;
24          }
25     }
```

图 4.13 关键词插入哈希表

查找的时候，通过相同的散列函数计算记录的散列地址，并按照散列地址访问该记录。当哈希表里面哈希值与邮件中的哈希值相符时，该邮件为正常邮件。如果在哈希表里面，没有找到邮件产生的哈希值，该邮件继续为垃圾邮件。哈希表查找算法[27]如图 4.13 所示。

```
输入：struct keyNum*head, int m    /*查找链表 head 中是否存在 m*/
输出：int k                          /* k 的值 0 或者 1*/
1       int k=0;
2       struct keyNum*p;
3       p=head;
4       if(head!=NULL)
5            do
6            {
7                 if(p->key==m)     //存在 m
8                 {
9                      k=1;
10                      break;       //跳出循环
11                 }
12                 p=p->next;
13           }while(p!=NULL);
14      return(k);                  //存在 m 值则返回 1，否则返回 0;
```

图 4.13 哈希表查找

4.4 邮件过滤系统的整体实现

邮件过滤系统实现过程就是依次调用每个过滤模块，依次分别处理邮件。为了，便于管理，只用一个主函数调用。每一个模块写成动态链接库的形式。重要用户模块叫 important.dll，去重复模块是 chongfu.dll，过滤 URL 模块是 feifaurl.dll，贝叶斯分类模块是 bayesfen.dll，哈希表查找模块是 haxibiao.dll。函数的处理过程，是按照处理的步骤进行，先是判读是否为重要用户，重要用户的邮件是正常邮件，剩余的邮件再去除重复邮件，依照四个条件判读条件快速比较。垃圾邮件判断，判读是否含有非法的链接，按照相似度比较是否含有经常出现的非法链接，其他的网页链接换算成散列值比较。然后，是贝叶斯模块比较，判读邮件属于垃圾邮件的概率。如果属于垃圾邮件概率是正常邮件概率的 8 倍，该邮件判为垃圾邮件。最后是对所有的垃圾邮件，进行哈希表关键词查找，查找是否含有单位特征的关键词，如果含有的话，该垃圾邮件重新判为正常邮件确保单位特征的邮件为正常邮件。因此，只要设计了一个主函数，每次获取动态链接库地址，给动态链接库传递一个文件路径。分别调用模块的动态链接库了。实现过程如图 4.15 所示。

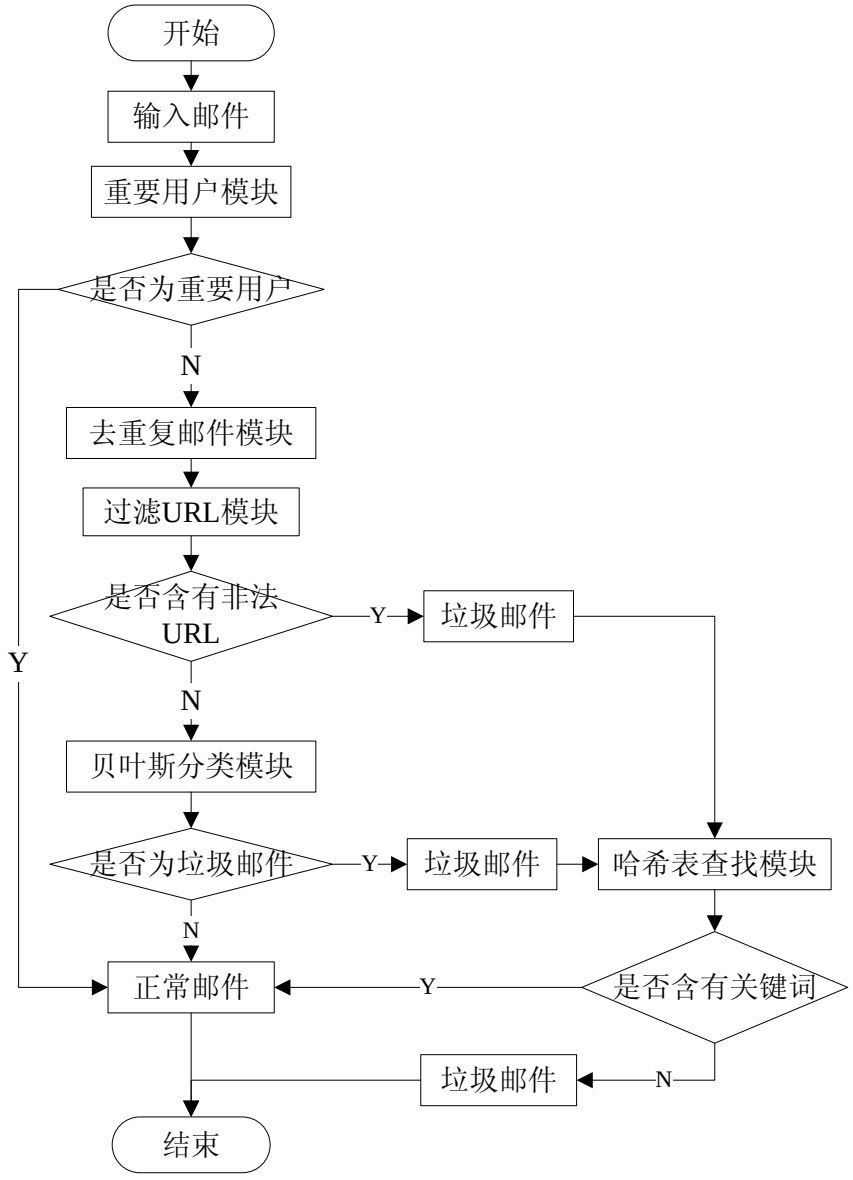


图 4.15 邮件过滤系统实现模块调用流程图

主函数通过调用每一动态链接库完成邮件的操作。主函数，先是获取邮件，然后调用重要用户模块，获取发件人邮件，判读是否为重要邮件，再调用去重复模块，去除重复邮件。再调用过滤非法链接模块，过滤网页邮件。剩余的邮件，再调用贝叶斯模块，进行邮件分类。最后，对所有的垃圾邮件，调用哈希查找模块确保有单位特征的邮件，重新判为正常邮件。通过，这样的五个模块的调用，既可以最大限度过滤垃圾邮件，又可以降低正常邮件被误判为垃圾邮件。主函数按照处理顺序调用动态链接库，具体的调用动态链接库的方法如图 4.16 所示。

```
输入：CString DirName     /*一个文件夹路径如：DirName="D:\\...\\tempmail"*/
输出：一个布尔值
1    /*调用函数的动态连接库开始*/
2    HINSTANCE hInst;                    //把 hInst 设置为动态库的句柄
3   hInst=LoadLibrary("important.dll");   //不成功，LoadLibrary 函数就返回 NULL
4    typedef BOOL(*PFNRECT)(CString c);
5    PFNRECT Repeat;                     //把 Repeat 设置为 PFNRECT 函数的地址
6    Repeat=(PFNRECT)GetProcAddress(hInst,"DisposeDirectory");//获取动态库地址
7    if(!Repeat)
8    {
9          MessageBox("获取函数动态连接库地址失败!");
10          return false;
11    }
12   Repeat(DirName);                    //传递路径 DirName
13   FreeLibrary(hInst);                 //释放动态链接
14   /*调用函数的动态连接库结束*/
15   return true;                        //返回 true
```

图 4.16 主函数动态数据库的调用算法

4.5 本章小结

本节主要是研究了邮件过滤系统的实现。首先简单介绍了系统的简单功能，然后是介绍了邮件过滤系统的几大模块。总的来说对于一封新的邮件，要经过三个步骤，第一步看看是否是重要用户发的邮件。如果不是重要用户发的邮件，就要去除重复邮件。第二步读取邮件的 URL，遍历 URL ,看看是否是垃圾邮件的 URL。第三步基于最小风险贝叶斯过滤系统过滤。为了进一步降低正常邮件被误判的风险，对于被系统判为垃圾的邮件。通过查找关键词的哈希值，进一步提高邮件识别率。最后介绍了系统整个模块的调用。

第五章 邮件过滤系统的测试

本章主要是对邮件过滤系统进行测试。为了得到详细的测试结果，首选对每个动态链接分别测试。然后调用所有的链接，对邮件过滤系统整体测试并分析了邮件系统过滤的优越性。

5.1 测试环境简介

本章对邮件过滤系统软件的基本功能进行测试。该软件包含：一个主的应用程序，还有 5 个动态链接，每一个动态链接实现一个功能。动态链接的具体情况是：zhongyaoyonghu.dll 重要用户过滤，chongfumail.dll 去除重复邮件，urlmai.dll 过滤 URL 功能，beiyesiclass.dll 贝叶斯分类，haxifind.dll 哈希查找。测试样本是：准备好的 100 份正常邮件和 100 份垃圾邮件。

垃圾邮件过滤系统的测试性能指标包括使用准确率、查全率和F_1测试值来衡量结果的好坏。

$$准确率(\text{precision}) = \frac{n_{S\to S}}{n_{S\to S}+n_{L\to S}} \quad 式(5\text{-}1)$$

$$查全率(\text{recall}) = \frac{n_{S\to S}}{n_{S\to S}+n_{S\to L}} \quad 式(5\text{-}2)$$

$$F_1测试值 = \frac{准确率\times查全率\times 2}{准确率+查全率} \quad 式(5\text{-}3)$$

本系统的测试环境表 5.1 所示。

表 5.1 试验平台环境说明

操作系统	Windows XP 是微软公司发布的一款视窗操作系统。
开发环境	Visual C++ 6.0，简称 VC 或者 VC6.0[28]是微软推出的一款编译器，将高级语言翻译成机器语言，使用的编程语言是 C++。
硬件环境	内存：1.87G CPU：2.93Hz 硬盘空间：800G

5.2 功能测试与分析

功能测试是按照垃圾邮件过滤系统的 ，来检查被测试的程序是否满足要求。分别对程序的某个模块和整个系统进行测试，检查过滤系统能否达到预期的效果。

本文采用黑盒测试[29]，就是把程序看出一个黑匣子，不考虑程序内部的情况，

通过输入和输出，对程序进行基本功能测试。首先，对单个的动态链接库分别进行测试，来检查系统的部分功能。最后，对整个过滤系统测试，检测整个过滤系统的效果。限本章对每一个功能模块分别进行功能测试。

5.2.1 重要用户模块功能测试

给定 200 封邮件，其中 2 封邮件，含有重要的用户发件人。系统识别出两份正常邮件，其余的 198 份邮件不作处理。对重要用户模块功能测试用例如表 5.2 所示。

表 5.2 过滤 URL 模块功能测试用例

功能模块名称	重要用户模块		
功能特性	调用对应的动态链接，并能在界面上显示结果		
测试目的	验证功能的正确性		
预置条件	准备好测试邮件，相应动态链接库，进入应用程序主界面		
用例编号	测试步骤	预期结果	测试结果
DL001	调用动态链接	程序会自动调用重要用户模块动态链接库；没有弹出一个对话框显示调用该动态链接失败	通过
DL002	等待结果	在输出的路径下会建立一个名为正常邮件的文件夹，存放识别出来的正常邮件。	通过

测试分析：这两份邮件的发件人邮件地址是被包含在重要用户名单中。其余的 198 份邮件，不作处理。

5.2.2 去重复邮件模块功能测试

给定 200 封邮件，其中 2 封邮件，是重复邮件。系统识别出两份重复邮件，其余的 198 份邮件不作处理。对去重复邮件模块测试用例如表 5.3 所示。

表 5.3 去重复邮件模块功能测试用例

功能模块名称	去重复邮件模块		
功能特性	调用对应的动态链接，并能在界面上显示结果		
测试目的	验证功能的正确性		
预置条件	准备好测试邮件，相应动态链接库，进入应用程序主界面		
用例编号	测试步骤	预期结果	测试结果
DL001	调用动态链接	程序会自动调用去重复邮件模块动态链接库；没有弹出一个对话框显示调用该动态链接失败	通过
DL002	等待结果	在输出的路径下会建立一个名为重复的文件夹，存放识别的重复邮件。	通过

测试分析：这两份邮件是重复邮件，其余的 198 份邮件，不作处理。

5.2.3 过滤 URL 模块功能测试

经过训练样本，收集了一些非法的 URL 动态链接，存入数据库里。用 100 份垃圾邮件和 100 份正常邮件做实验。系统识别出：50 封垃圾邮件，150 封正常邮件。对过滤 URL 模块动态链接库测试的功能用例如表 5.4 所示。运行结果如图 5.1 所示。

表 5.4 过滤 URL 模块功能测试用例

功能模块名称	过滤 URL 模块		
功能特性	调用对应的动态链接，并能在界面上显示结果		
测试目的	验证功能的正确性		
预置条件	准备好测试邮件，相应动态链接库，进入应用程序主界面		
用例编号	测试步骤	预期结果	测试结果
DL001	调用动态链接	程序会自动调用过滤 URL 模块动态链接库；没有弹出一个对话框显示调用该动态链接失败	通过
DL002	等待结果	在输出的路径下会建立两个文件夹，识别出来的正常邮件和垃圾邮件分别放入不同的文件中；主函数界面会显示，处理了邮件数、识别的垃圾邮件数、正常邮件数	通过

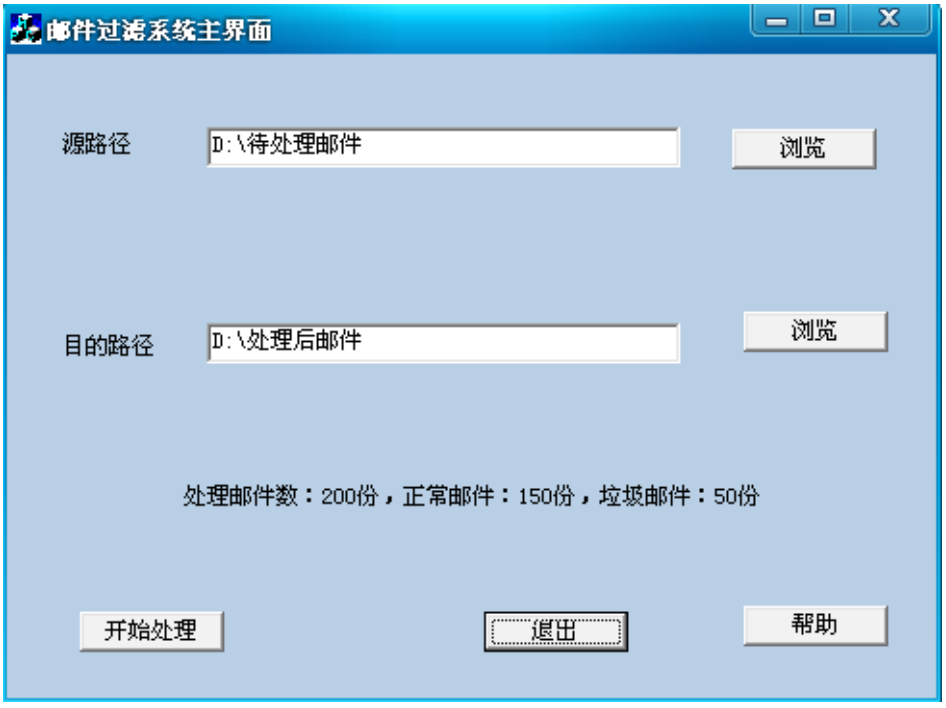


图 5.1 过滤 URL 模块测试结果显示效果图

测试结果分析：准确率是 100%，查全率是 50%，F_1 测试值是 66.67%,能够被过滤的垃圾的邮件，里面都包含了 URL 数据库的非法链接。对剩余的 50 份垃圾邮件分析，有 26 份没有超链接。另外的 24 份邮件没有包含 URL 数据库的非

法链接,功能测试达到预期效果。

5.2.4 贝叶斯分类模块功能测试

准备好 100 份垃圾邮件和 100 份正常邮件。调用所贝叶斯分类模块动态链接测试，显示该模块的功能。应用程序以及该动态链接库测试的功能用例如表 5.5 所示。运行结果如图 5.2 所示。

表 5.5 贝叶斯分类模块功能测试用例

功能模块名称	贝叶斯分类模块		
功能特性	调用贝叶斯分类模块动态链接，并能在界面上显示结果		
测试目的	验证功能的正确性		
预置条件	准备好测试邮件，所有的动态链接库，进入应用程序主界面		
用例编号	测试步骤	预期结果	测试结果
DL001	调用动态链接	程序会自动调用该模块动态链接库；没有弹出一个对话框显示调用该模块动态链接失败	通过
DL002	等待结果	在输出的路径下会建立两个文件夹，识别出来的正常邮件和垃圾邮件分别放入不同的文件中；主函数界面会显示，处理了邮件数、识别的垃圾邮件数、正常邮件数	通过

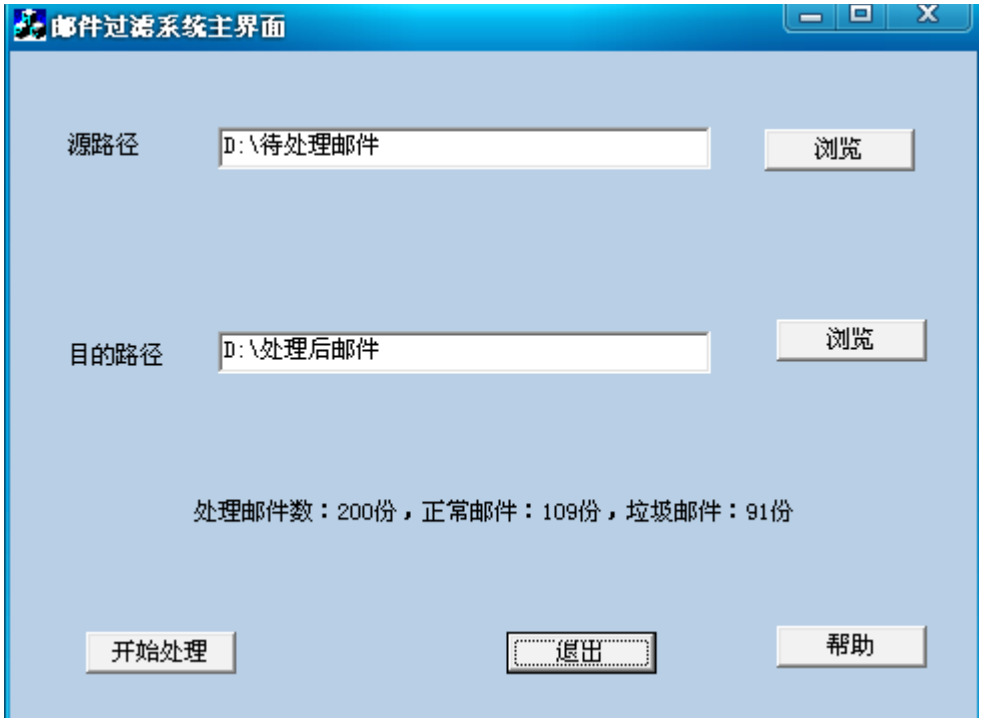


图 5.2 系统功能测试效果图

测试结果分析：用 100 份正常邮件和 100 份垃圾邮件测试。调用该动态链接库测试的结果为：能够识别全部的正常邮件：99 份，识别垃圾邮件：90 份，正常邮件判为垃圾邮件为：1 份，垃圾邮件判为正常邮件是：10 份。准确率是 98.9%,

全差率是 90%,F_1测试值是 94.24%。因此只要找到合适训练样本以及合理的阈值，就能比较好过滤邮件。

5.2.5 哈希表查找模块功能测试

将一份垃圾邮件的内容：添加关键词：“司令部参谋”，另外一份邮件的内容添加关键词“政治部干事”。重新放入邮件过滤程序，两份邮件都判为：正常邮件。对哈希表查找模块测试用例如表 5.6 所示。

表 5.6 去重复邮件模块功能测试用例

功能模块名称	哈希表查找模块		
功能特性	调用对应的动态链接，并能在界面上显示结果		
测试目的	验证功能的正确性		
预置条件	准备好测试邮件，相应动态链接库，进入应用程序主界面		
用例编号	测试步骤	预期结果	测试结果
DL001	调用动态链接	程序会自动调用哈希表查找模块动态链接库；没有弹出一个对话框显示调用该动态链接失败	通过
DL002	等待结果	在输出的路径下，将两份邮件从垃圾邮件文件夹移到正常邮件文件	通过

测试分析：含有关键词邮件可以重新判为正常邮件，测试通过。

5.3 性能测试

在第四章贝叶斯算法设计的基础，先对传统的朴素贝叶斯算法进行训练集和训练特征选择，分别得到朴素贝叶斯算法相应的准确率、查全率和F_1综合测试值。接着对改进后基于最小风险的贝叶斯算法进行阈值选择，分别得到最小风险贝叶斯相应的准确率、查全率和F_1综合测试值。最后，对改进后算法比较，得出最小风险贝叶斯算法能够提高准确率，降低正常邮件被误判为垃圾邮件的风险。

5.3.1 朴素贝叶斯性能分析

过滤器的精确度和训练集的大小有直接的关系，为了得到一个合适的训练集，本文采用不用数量的训练集，通过训练来构造最优的过滤器。运用朴素贝叶斯算法在 VC6.0 软件开发环境下进行测试。

由于特征项数量的多少直接影响到学习算法的性能。特征项数量少的时候，分类的精确度会随着特征数量的增加，越来越高。然而特征项数量达到一定的数量，分类精度反而会随着特征数量增加下降。同时伴随特征数量的增加，学习的复杂度和时间也增加了。

在构建过滤器的时候，要考虑到训练集的数量也要考虑到特征项数量，当特征向量数量过多时，要想办法减少特征向量数量。

本实验表明维数不超过 3 的特征项，会对实验数据影响很大。因此先不考虑训练样本集中出现次数少于给定频度 4 的特征项。

定义: (频度) 在文档 d 中,N-gram 信息 t 项的频度[30]用它在文本 d 中出现的次数用 fq 表示。

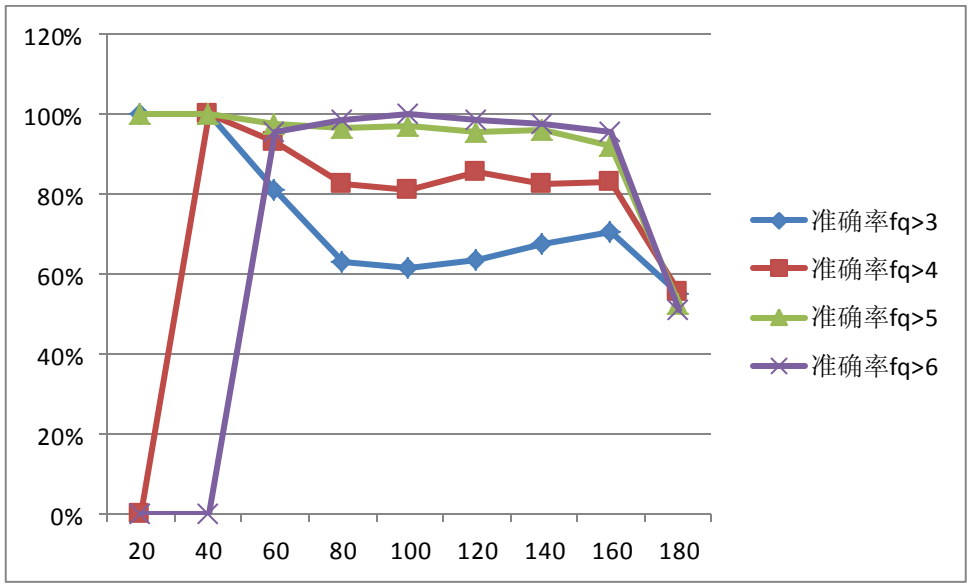
实验具体步骤如下：

(1)通过多种渠道收集了 380 篇邮件，其中包括了 190 篇垃圾邮件和 190 篇合法邮件。在文本实验中又把邮件划分为正常邮件(C_1)与垃圾邮件(C_2)两类;

(2)把邮件训练集又分为样本集和测试集，其中训练样本集 180 篇邮件，测试集 200 篇邮件;

(3)训练样本集从邮件数 20 篇开始，依次取出 20,40,60,80,100,120,140,160,180 篇邮件作为训练样本，对测试集的邮件进行分类与过滤。

根据实验结果计算朴素贝叶斯的每个准确率如图 5.3 所示。

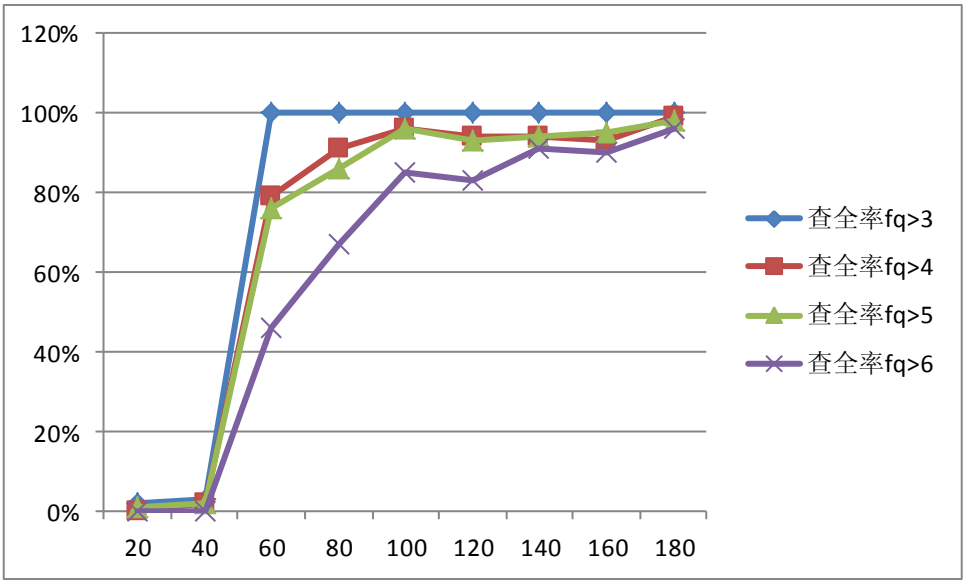


X 表示训练集数量大小，Y 轴表示准确率大小

图 5.3 训练集数量大小和准确率大小关系图

从图 5.3 可知，当邮件训练数量少的时候，邮件的过滤的准确性不稳定，当邮件数量达到 80 的时候，各种频度的下邮件的稳定性呈现出一定的稳定性和规律，准确率呈现上升趋势，然后当训练邮件的数量达到了 160 篇后准确率开始下降，尤其是训练的邮件数量快达到 180 篇准确率下降幅度较大。

根据实验结果计算朴素贝叶斯的每个查全率如图 5.4 所示。

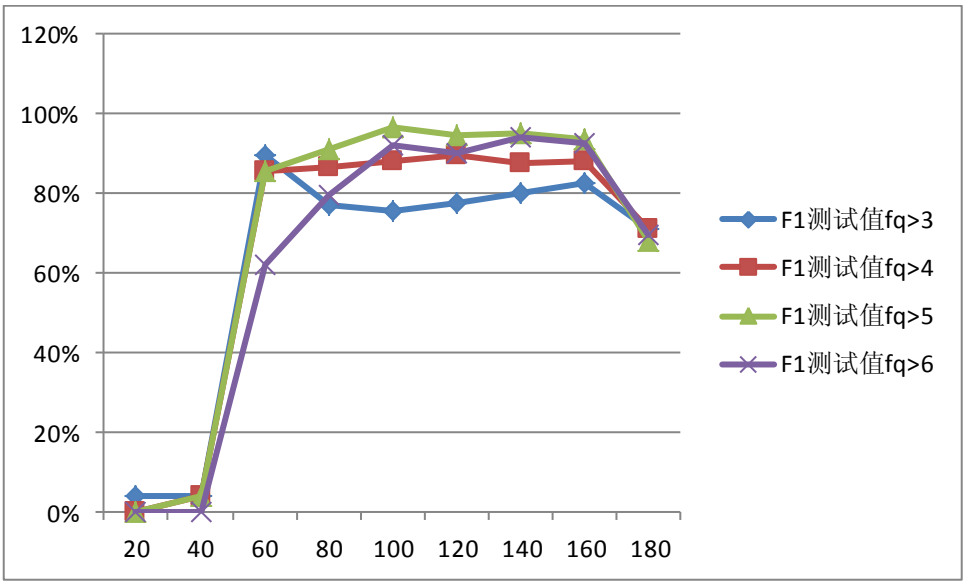


X 表示训练集数量大小，Y 轴表示查全率大小

图 5.4 训练集数量大小与查全率大小关系图

从图 5.4 可知，随着训练样本集的增大，邮件的查全率也增加。当样本。在样本数量是 20 和 40 之间，查全率很低，没有超过 2.00%。当训练的样本数量在 40 与 100 之间的时，查全率上升的很快。当训练样本数在 100 至 160 之间，查全率曲线变化处于基本稳定状态。当训练样本超过 160 篇后，查全率随着样本集的增大而增大。有一个例外的是频度 fq>3 变化的更加早些。因此查全率是随着样本的增加不断增加。

为了选择合适的训练样本集，需要计算F_1综合测试值。将图 5.3 和图 5.4 相应的数据分别计算F_1综合测试值，并把得到的结果列作图 5.5 所示。



X 表示训练集数量大小，Y 轴表示F_1测试值大小

图 5.5 训练集数量大小与F_1测试值大小关系图

由图 5.5 可知，当样本数小于 40 时，F_1 测试值几乎为零。当样本大于 40 后，F_1 测试值增加很快。当数量到达 60 后，fq>3 的测试值反而下降，其他三个的测试值增加变慢。当样本数量超过 80 后， 所有的F_1 测试值都增加，并且保持增加到样本数量是 160 的。不过，样本数量过了 160 后，所有的F_1 测试值开始下降了，其中邮件特征频度 fq>5 时F_1 测试值最大。因此，样本数量取 160，特征项频度 fq>5 来构造过滤器的训练集最好。

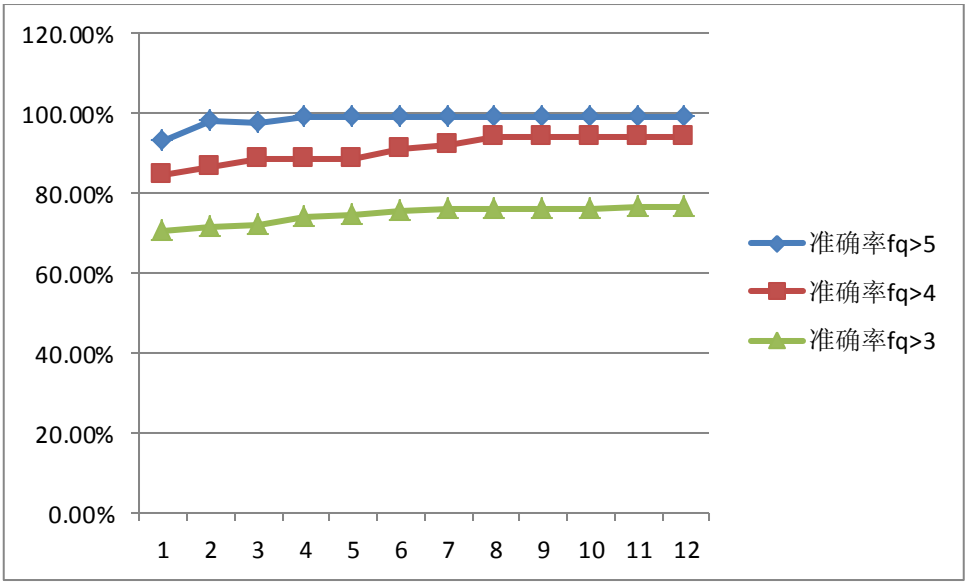
5.3.2 改进后的最小风险贝叶斯

朴素贝叶斯是基于最小错误的抉择，正常邮件别误判为垃圾的邮件的可能性比较大。当正常邮件被误判为垃圾邮件的时候常常会给用户带来很大的损失，所以考虑用最小风险贝叶斯算法对朴素贝叶斯算法，希望让用户损失最小。一个邮件属于合法邮件和垃圾邮件的概率分别是P($\frac{C_1}{d}$)和P($\frac{C_2}{d}$)。

$$\frac{p(\frac{C_2}{d})}{p(\frac{C_1}{d})} \geq \lambda \qquad \text{式(5-4)}$$

λ的值称为阈值。先用 160 篇做成训练样本，测的下面的数据。

根据实验结果计算最小风险贝叶斯的每个准确率如图 5.6 所示。



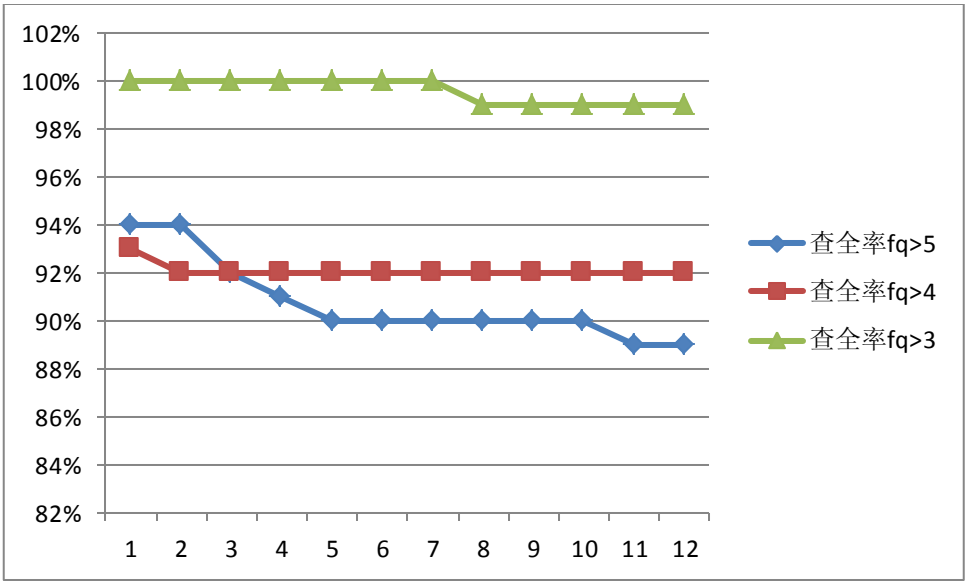
X 表示阈值(λ)大小 ， Y 表示准确率的大小

图 5.6 基于最小风险主动贝叶斯邮件过滤算法阈值大小与准确率的关系

随着阈值λ增加，三个频度准确度也都增加，当阈值λ增加到 8 时，准确度增加到最大基本上趋于稳定。再到阈值λ增加到 10，准确度又有些增大。从图 5.6 上，可以看出，准确度的情况是：

准确率(fq>5)>准确率(fq>4)>准确率(fq>3).增加阈值后，正常邮件被误判垃圾邮件的概率降低了，邮件的准确度就提高了。

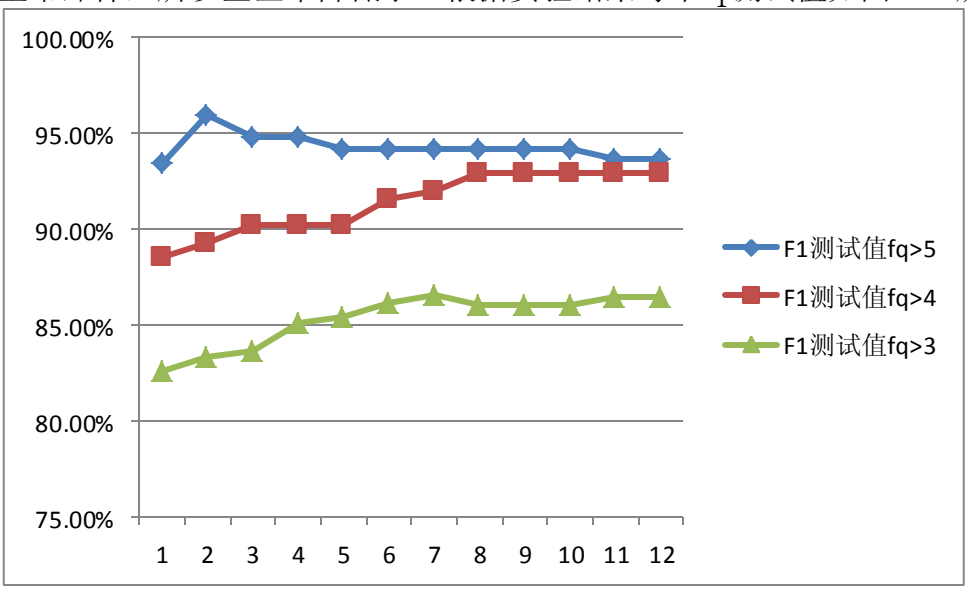
根据实验结果计算最小风险贝叶斯的每个查全率如图 5.7 所示。



X 表示阈值(λ)大小 ，Y 表示查全率的大小

图 5.7 基丁最小风险主动贝叶斯邮件过滤算法阈值大小与查全率的关系

随着阈值λ 增加，三个频度查全率也都减少。当阈值λ增加到 8 的时候，查全率稳定下来了。因为设置阈值λ，正常邮件被误判为垃圾邮件的概率减少了，也就是提高了判断垃圾邮件的门槛。从另外一方面来看，是更多的垃圾邮件被误判为正常邮件，所以查全率降低了。根据实验结果每个F_1测试值如图 5.8 所示。



X 轴表示阈值(λ)大小 ，Y 轴表示F_1测试值的大小

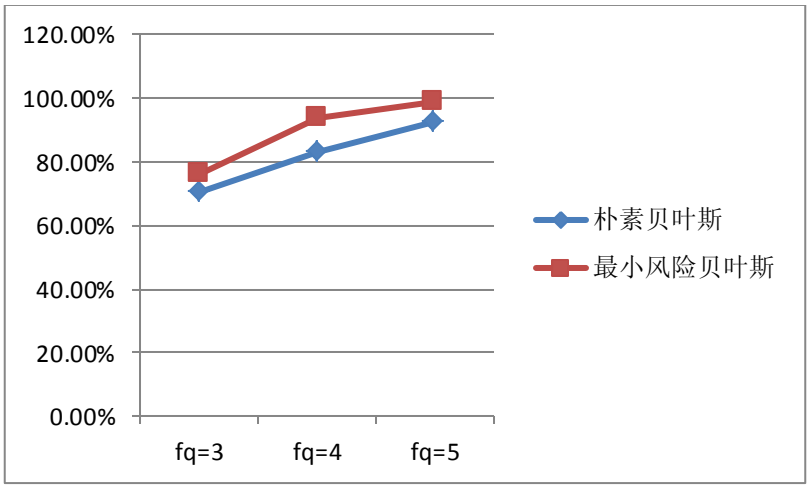
图 5.8 基于最小风险贝叶斯邮件过滤算法阈值大小与F_1测试值的关系

随着阈值λ 增加，邮件F_1测试值逐渐增加，F_1测试值(fq>5)> F_1测试值(fq>4)> F_1测试值(fq>3)。阈值λ到达 4 时候，F_1测试值增加缓慢。当阈值λ到达 8 时候，F_1测

试值基本上增加到了最大值，趋于平稳了，F_1测试值(fq>5)也达到了最大值。综合以上所叙述，阈值取值λ = 8,可以得到较高的测试值。

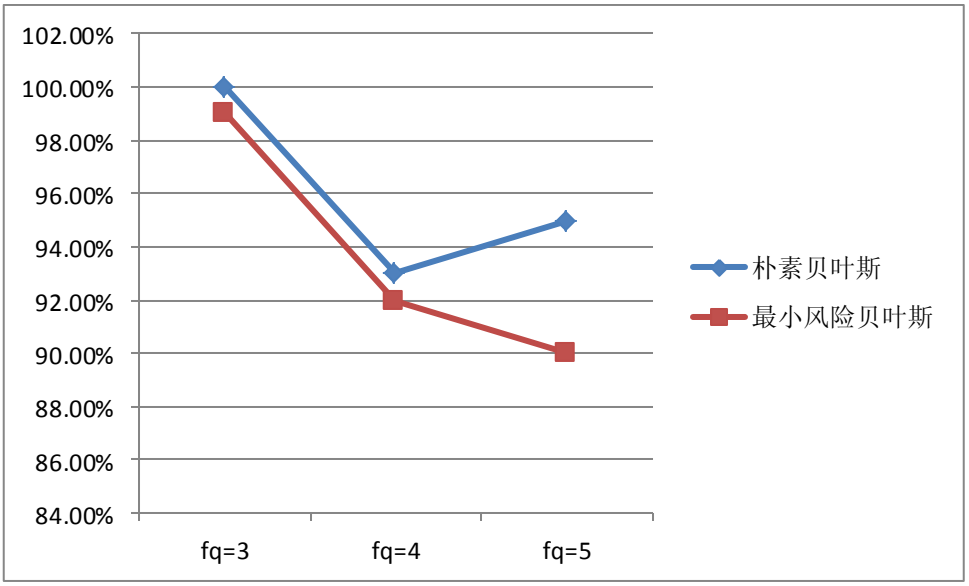
5.3.3 两种贝叶斯算法性能比较

为了比较朴素贝叶斯过滤和改进后最小风险贝叶斯过滤的情况，把两组数据整理如下图 5.9、图 5.10、图 5.11，这样就能够更好评价基于最小风险贝叶斯优越性。



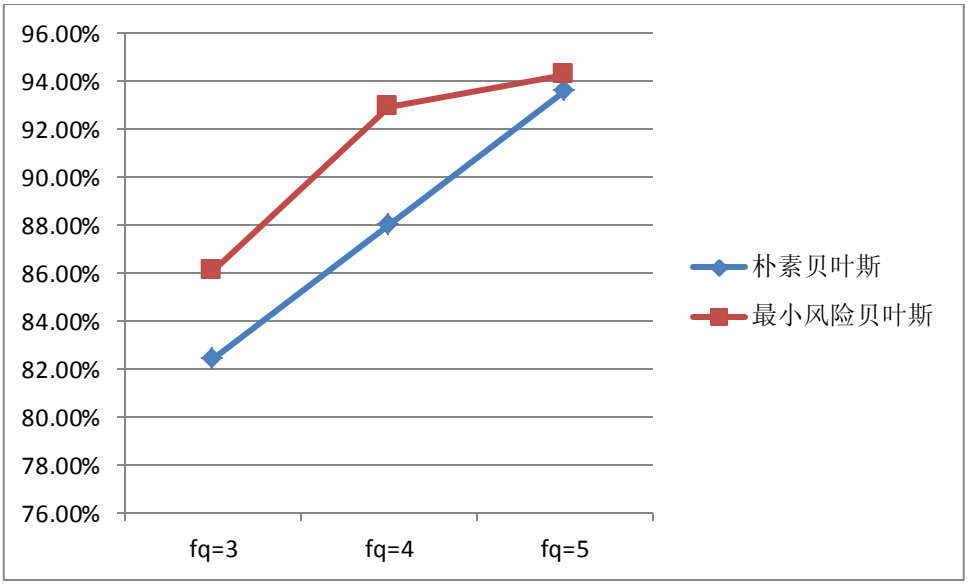
X 轴表示频度大小，Y 轴表示准确率大小

图 5.9 朴素贝叶斯和最小风险贝叶斯准确率比较



X 轴表示频度大小，Y 轴表示查全率大小

图 5.10 朴素贝叶斯和最小风险贝叶斯查全率比较



X 轴表示频度大小，Y 轴表示F_1测试值大小

图 5.11 朴素贝叶斯和最小风险贝叶斯F_1测试值比较

从图 5.9、图 5.10、图 5.11 分析可知，基于最小风险的贝叶斯算法准确率和F_1测试值是要比朴素贝叶斯算法高一些。因为判为垃圾邮件的门槛提高了，基于最小风险的贝叶斯算法比朴素贝叶斯算法的查全率是要低一些。综上所述，当阈值$\lambda = 8$和 fq>5 时，基于最小风险的贝叶斯就能减低正常邮件误判为垃圾邮件。

5.4 本章总结

本章对过滤系统的每一个模块进行了单独功能测试。测试样本是选好的邮件。单独测试的动态链接，还是达到了预期的效果。又对其中一个改进算法模块进行了性能测，结果表明，改进后的算法优于前面的算法，邮件过滤的效果更好。因此该邮件过滤系统，既可以过滤垃圾邮件，又可以降低邮件风险。

第六章 结束语

6.1 工作总结

日益泛滥的电子邮件给大家带来很大的麻烦，因此过滤邮件是一个很有意义的事情。本来在分析和研究传统邮件过滤方法的基础上，改进了两个算法，同时将哈希表查找应用于邮件关键词的查找中，提高工作效率。本文，还搭建了试验平台，对过滤系统进行验证分析。

本文主要研究工作和贡献：

(1)建立好了关键用户表，确保重要发件人的发送的邮件当作正常邮件。分析重复邮件特征，自定义去除重复邮件的四个标准，发件人、邮件主题、邮件附件名、以及邮件大小。四个标准可以准确，快速地去除重复邮件。

(2)改进了过滤非法 URL 的算法。先是分析邮件非法 URL 特征，对经常出现的非法 URL，进行近似度比较，减少了 URL 数据库的维护工作量。其他的 URL 先换算成固定长度散列值进行比较，大大提高比较速度。

(3)改进了贝叶斯算法。在朴素贝叶斯分类的基础上，运用 EM 算法。同时训练了邮件样本，通过实验设置阈值大小，减少了邮件误判垃圾邮件的风险。改进后的贝叶斯分类算法，在邮件过滤判读指标：准确率，查全率、F_1测试值，都优于改进前的算法

(4)使用哈希查找在邮件中查找关键词。对过滤后的垃圾邮件，查找关键词。通过哈希表查找，确保有职业特征的邮件重新判为正常邮件，提高了效率。

(5)对整个邮件过滤系统进行了设计和测试，结果表明该邮件过滤系统既能有效过滤垃圾邮件，又最小风险地降低正常邮件误判为垃圾邮件。

6.2 工作展望

邮件过滤技术是一个日新月异的快速发展技术。因为垃圾邮件发送者，也总是在不断地提高了垃圾制造的水平，所有邮件过滤技术也要跟着进步。新问题和新技术将不断出现。本文虽然改进了两个算法，应用了 5 个过滤技术，但是还有一些不足。主要有以下几个方面。

(1)数据库资料的维护和配置还要完善，要求更加方便和实用。

(2)垃圾邮件是日新月异不断变化的，今后还学习新的过滤方法，提高效率。

(3)需要建立一个用户反馈机制，让用户反馈垃圾邮件特性。

(4)垃圾邮件数量巨大，还需要在海量存储和数据挖掘方面，下工夫，应用数据挖掘技术，找到邮件的有用信息。

(5)除了普通的垃圾邮件外，病毒邮件，也是日益泛滥，在后面的工作，要考虑把病毒邮件，也纳入垃圾邮件范畴。

(6)邮件过滤算法，还可以从其他方面进行改进，这也是下一步的工作重点。

总之，垃圾邮件过滤系统目前还有许多不足，下一步应当对这些工作，进行研究，争取建立更好的过滤系统。

致谢

在整个论文的最后，我要对在我的成长道路上所有给予我帮助和鼓励的人表示由衷的感谢。

首先，我衷心感谢我的导师郑有才老师，郑老师严谨的治学态度、娴熟而深厚的专业知识和对学生无私的关心和照顾，深深影响了我。在研究生学习期间，老师在专业知识方面的传授和治学修身的教导让我终身受益。在论文的选题、写作和修改，老师都倾注了大量的心血，给我了详细和耐心的指导，提出了许多宝贵的意见，使我能够顺利完成论文。因此，我要再次向辛勤教导我的导师表示深深的感谢和敬意！

其次感谢软件研究所刘伟老师和王献青老师，在研究生学习期间，刘老师和王老师给予我很多帮助，他们传授给我很多有用的专业知识，他们身上的很多优秀品质值得我们学习。

当然，还要特别感谢同学蔡清茂，还要感谢 1307 试验室的谭继辉、王栋、李素超、方卫华、吴荣芳、张爱莲、李贺,感谢他们在生活中对我的帮助,科研实践中的合作，论文写作中给予我许多建设性意见。

此外，还要感谢我的父母、妻子和两个姐姐。读研期间，母亲中风卧病生活不能自理，一直是父亲和两个姐姐照顾。妻子也是一个人远在海南上班。没有家人的默默支持，就没有我的今天。再次向我的家人表示最由衷的谢意。我还要感谢所有关心和爱护我的朋友们。

感谢培育了我的西安电子科技大学计算机学院，感谢和我朝夕相处的同学们，我将永远珍惜我们共同营造的那份轻松与欢乐，衷心祝你们事业有成!

最后，衷心感谢在百忙之中评阅论文和参加答辩的各位专家、教授！

参考文献

[1]冯英健.Email 营销[M].北京：机械工业出版社，2003 年.

[2]陈胜权,任平,陈杰.UTM(统一威胁管理)技术概论[M].北京:电子工业出版社.2009 年.

[3]李玮,赵燕平.基于社会网络分析的 E-mail 内容动态监测模型[J].北京理工大学学报.2006,26(z1):79-83.

[4]周永,陈永章,张伟文.一种复合的双引擎智能垃圾过滤方法[J].计算机应用研究.2008,25(1):268-270.

[5]李石君,李洲,余军,张科.基于 URL 过滤与内容过滤的网络净化模型[J].计算机技术与发展.2006,16(1):5-7.

[6]阈值[OL].http://baike.baidu.com//view/409216.htm.

[7]杜慧军,杨宁.基于路由表哈希匹配算法的压缩策略[J].系统工程与电子技术.2007,29(11):945-948.

[8]孙杰. 网管员世界[J]. 北京:电子工业出版社.2011.13: 70-71.

[9] [美]Andrew S.Tanenbaum 著.计算机网络[M]. 第四版. 潘爱民译. 北京: 清华大学出版社 .2004 年

[10]中国反垃圾邮件联盟[OL].http://www.anti-spam.org.cn.

[11]刘洋，杜孝平，罗平. 垃圾邮件的智能分析过滤及 Rough 集讨论[J].第十二届中国计算机学会网络与数据通信学术会议.2002, 12:541-546.

[12]边肇祺 ,张学工. 模式识别 [M] . 北京：清华大学出版社 .2000.

[13] W.Cohen. Learning rules that classify email. 1996 in Proceedings of the AAAI spring,symposium of Machine Learning in Information Access，Palo Alto，California：1 8-25.

[14]邻近算法[OL]. http://baike.baidu.com/view/1485833.htm.

[15]吴立德.大规模中文文本处理[M]. 上海:复旦大学出版社,1997.

[16]孙振辉.贝叶斯理论在反垃圾邮件中的应用研究[J]. 科技广场. 2009, 3:73-75.

[17]骆丽娟.贝叶斯方法在垃圾邮件过滤中的应用[J].科技信息. 2007, (28):324-328.

[18]汤伟,程家兴,纪霞. 统计学理论在邮件分类中的应用研究[J]. 计算机技术与发展.2008,18(12):231-234.

[19]Bing Liu 著.Web 数据挖掘[M]. 俞勇,薛贵荣译.北京:清华大学出版社.2009.

[20]王美珍.垃圾邮件行为模式识别与过滤方法研究[D].武汉: 华中科技大学.2009.

[21]郑炜,沈文,张英鹏. 基于改进朴素贝叶斯算法的垃圾邮件过滤器的研究.西北工业大学学报. 2010, 28(4):622-627.

[22]姚伽华. 基于最小风险的主动贝叶斯邮件过滤算法研究[D].广州:华南理工大学.2004.

[23]欧红星.电子邮件安全过滤与检查技术研究[D].长沙: 中南大学.2008.

[24]肖雯.基于贝叶斯分类的邮件过滤方法及模型研[J]. 南京师范大学学报.2006,6(2):86-89.

[25]严蔚敏,吴伟民.数据结构[M].北京:清华大学出版社.1997.

[26]詹川.反垃圾邮件技术的研究[D].成都:电子科技大学.2005.

[27]哈希表[OL]. http://www.cnblogs.com/lpshou/archive/2012/2490191.html.

[28]黄维通.Visual C++面向对象与可视化程序设计[M]。第二版.北京:清华大学出版社,2003.

[29]齐志昌,谭齐平,宁洪.软件工程[M].第二版.北京:高等教育出版社,2004.

[30]周水庚，关估红，俞红奇，胡运发.基于 N-Gram 信息的中文文档分类研究[J].中文信息学报.2001(1):34-39.

邮件过滤系统的研究与实现

作者：袁晓容
学位授予单位：西安电子科技大学