一個適性化的垃圾郵件過濾系統

何篤生* 張阜民** 高勝助*
*中興大學資訊科學系
**朝陽科技大學財務金融系
*{w9156007,sjkao}@cs.nchu.edu.tw
**fmchang@mail.cyut.edu.tw

摘要

在本文中，我們設計與實作一個以貝氏機率方法為基礎，並同時結合伺服器端與使用者端過濾功能的適性化垃圾郵件過濾系統。在本系統，藉由黑、白名單機制以及權重觀念的應用，讓使用者端可依個人需求，自訂黑、白名單及權重值，以建立整合伺服器端與客戶端需求之個人化垃圾郵件過濾規則。對於一般比較不熟悉電腦系統的使用者，則可以完全透過系統的處理，不需使用者自訂條件，也能逐步得到較佳過濾垃圾郵件的效果。另外，垃圾郵件過濾規則可在系統自動的訓練下，不斷持續地更新，讓系統不會因為垃圾郵件的日新月異，而降低垃圾郵件判別率。為了驗證系統的各項功能，我們在 Linux 的環境下實作一雛型系統，並從分類資料庫及權重參數調整對郵件判別率之影響、及與自由軟體之垃圾郵件過濾器，Spamassassin 與 POPFile，的比較等三個方面來做相關的分析報告，實驗的結果也說明了本系統的可行性。

關鍵詞：垃圾郵件、過濾器、貝氏機率、適性化

1 簡介

電子郵件的方便性及低成本，使其成為網際網路上普遍的一種服務。但隨著垃圾郵件的過度氾濫，造成網路頻寬被大量佔用，網路服務提供者的郵件主機內硬碟空間被垃圾郵件充斥，收件者也必須花費許多額外的時間去檢視及刪除垃圾郵件，降低了工作的生產力及造成時間上的浪費。因此，一個正確與有效的垃圾郵件過濾系統是有必要的。

一個正確與有效的垃圾郵件過濾系統，通常需結合好幾項過濾技術，才可以讓過濾效果提升[2]。目前來說已經有許多的垃圾郵件過濾器技術被提出，如基本結構文字過濾器(Basic Structured Text Filters)[2][3]、白名單加上確認驗證過濾器(Whitelists/Verification Filters)[2][3]、分散型適應式黑名單(Distributed Adaptive Blacklists)[2]、規則分數計算法(Rule-Based Rankings)[2]、貝氏文字式垃圾郵件過濾器(Bayesian Word Distribution Filters)[3]、及貝氏字詞式垃圾郵件過濾器(Bayesian trigram filters)[3][10]。在這些技術中，以貝氏機率方式為基礎所設計的垃圾郵件的過濾系統，不論在正確判斷垃圾郵件及正確判別出一般郵件的效果都遠高於其它種的技術。這些技術完整的比較分析，可以在 David Mertz 的文章中找到[2]。

目前大部份垃圾郵件過濾軟體的設計方式，可分為伺服器端處理及使用者端處理的方式。伺服器端處理的方式最大的優點是方便管理，一次可以對所有使用者來處理。缺點是缺乏了個別性，因為個別使用者對於垃圾郵件的定義可能是不同的，因此正確率較差。而使用者端的處理方式則彈性較大，正確率較高，但一般在使用初期效果通常不佳。在本文中，我們提出一個以貝氏機率方式為基礎的適性化垃圾郵件過濾系統。此系統在設計上同時結合伺服器端與使用者端的過濾功能，來增加系統的可適性。使用者端可以利用自動或手動調整權重值的方式，方便使用者端可以依個人需求，自訂符合自己的黑白名單及權重值，從而建立起個人郵件資料庫後與系統資料庫做整合。對於一般比較不熟悉電腦系統的使用者，則可以完全透過系統的處理，不需使用者自訂條件，也能得到較佳過濾垃圾郵件的效果。

2 系統架構

2.1 系統說明及架構

本系統是以貝氏機率為基礎所發展出來，使用者可透過 WEB 介面，自行調整權重值。整個系統是由三部份來組成：執行子系統，訓練子系統，與 WEB 介面子系統。子系統中有些模組在功能上是相互重疊的，因此這些相同功能的模組可以共用。整個系統架構如圖 1 所示。

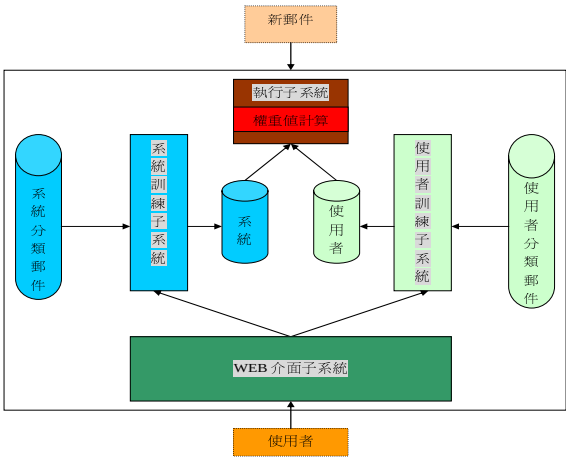

圖 1 系統架構圖

2.2 執行子系統

執行子系統的主要功能是判別新進郵件是否為垃圾郵件。共包含七大模組：新進郵件接收模組、郵件解析模組、標記處理模組、資料庫模組、權重計算模組、貝氏機率模組及郵件遞送模組。新進郵件接收模組接收新進郵件後，傳送給郵件解析模組，郵件解析模組解析後郵件後，會對照黑白名單。若是郵件地址已列在黑白名單中，直接將郵件送往郵件遞送模組處理；如不在黑白名單中，則表示需要經過判別，因此將解析的結果送往標記處理模組。標記處理模組取出郵件中的標記後，再將標記送往權重處理模組處理，權重處理模組分別到系統資料庫及使用者資料庫取出由訓練子系統得到的標記機率表，依照權重值大小，得到一個使用者專屬的標記機率表。由此標記機率表，可計算出機率值最高及最低的 15 個標記機率值後，送到貝氏處理模組。貝氏處理模組就以貝氏機率公式計算出這封新進郵件的總機率值，這總機率值代表著新郵件為垃圾郵件的機率，當這個總機率值超過某一個門檻值(由使用者自訂，一般為 0.9)[3]時，即判定為垃圾郵件，再將新進郵件送往郵件遞送模組，將新進郵件依照判定結果，將新郵件做遞送的處理。整個執行子系統處理流程與架構如圖 2 所示。

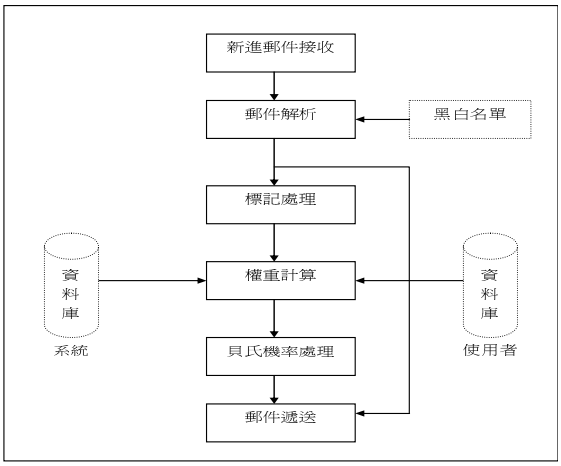

圖 2 執行子系統流程與架構圖

2.3 訓練子系統

訓練子系統的主要功能是對伺服器端及使用者端已分類好的郵件(一般郵件及垃圾郵件)做訓練的動作。共可分為三大模組：郵件解析模組，標記處理模組，及資料庫模組。先利用郵件解析模組將郵件做解析的處理，再利用標記處理模組，取出郵件中的標記，統計算出標記出現在一般郵件集合和垃圾郵件集合中的次數，再依公式計算得出每一個標記為垃圾郵件中的機率值，並且將這個機率值組成的機率資料表分別依系統端及使用者端儲存在資料庫內。整個訓練子系統處理流程與架構如圖 3 所示。

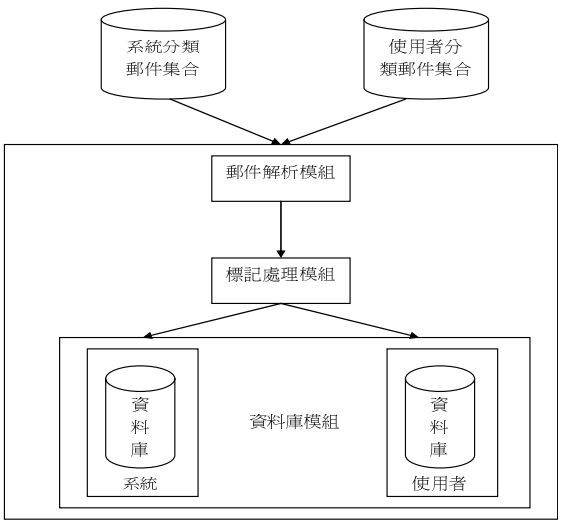

圖 3 訓練子系統處理流程與架構圖

2.4 WEB 介面子系統

WEB 介面子系統的主要功能是登入介面模組的提供，讓使用者可以透過瀏覽器 WEB 介面方式進入系統內。共分為四大模組：登入介面模組、操作介面模組、黑白名單模組、權重值模組。

2.5 各模組功能說明

郵件解析模組

負責讀取各種格式的郵件檔案，如 7 位元、8 位元、Uuencode 及多用途網際信件延伸(MIME)[11]格式，然後依照郵件內容，提供給其它模組使用。解析出的表頭(header)及內容(body)必須分別取出做適當處理。

標記處理模組

負責接收郵件解析模組的資料，將這些資料做適當的處理及分割，擷取出我們需要的標記部份，如字(word)或片語(phase)。再配合已分類郵件中一般郵件和垃圾郵件的總數量，以便和標記出現的總次數利用公式做計算，得到所有標記機率值的機率表後，再送往資料庫模組中儲存機率表資料。

資料庫模組

負責儲存標記出現次數的資料表和所有標記的機率表。標記出現次數資料表是用來計算每一個標記在垃圾郵件中的機率值；標記機率表是用來做貝氏統計計算的基本資料。

權重計算模組

負責接收標記處理模組的資料，依照標記處理模組處理出來的標記資料，再到系統資料庫及使用者資料庫中取出相關標記，再依照權重值大小，計算得到一個新郵件所有標記機率表，再將此標記機率表做排序處理，將排序後最大及最小的 15 個機率值傳送至貝氏處理模組中做處理。

貝氏處理模組

負責接收權重計算處理模組送過來的 15 個標記機率值。利用這 15 個機率值，經貝氏機率公式計算後，得到郵件的總機率值，此總機率值可用來判定郵件是否為垃圾郵件，並將判定的結果，傳送到郵件遞送模組。

郵件遞送模組

負責接收貝氏處理模組的判定資料，依判定的結果，對於郵件做遞送的處理。

登入介面模組

以 Linux 系統為基礎,以瀏覽器 WEB 介面方式,做帳號密碼的稽核，通過認證，即可進入本系統中做系統或個人過濾權重值啟動、設定….等相關功能。

操作介面模組

提供瀏覽器操作環境，系統管理者及個人使用者可方便做黑白名單及權重值模組的相關資料設定。

黑白名單模組

提供系統端及使用者端設定黑白名單的功能。黑白名單提供執行子系統中的郵件解析模組比對資料，只要新郵件相關標記符合黑白名單中的資料，郵件解析模組即可將新郵件直接送往郵件遞送模組做郵件的遞送或相關的處理。

權重值模組

提供權重值啟用、關閉及設定的功能。並提供資料給執行子系統中權重處理模組做郵件標記機率值的功能。

2.6 權重設計說明

本系統是利用二個資料庫再加上權重值來產生新的判別用機率表資料庫，以判斷新郵件是否為垃圾郵件。資料庫資料產生的過程中，會使用到的參數有四個：判別用資料庫標記機率(N_prob)、系統端資料庫標記機率(S_prob)、使用者端資料庫標記機率：(U_prob)、及權重值(weight)。新判別用機率表資料庫方法的產生可分成三種狀況：系統端及使用者端資料庫皆有標記機率資料、系統端資料庫有標記機率資料，而使用者端資料庫沒有標記機率資料、和系統端資料庫沒有標記機率資料，而使用者端資料庫有標記機率資料。其計算公式分別為

N_prob=S_prob*(1-weight) + U_prob*weight 、

N_prob= S_prob*(1-weight) + 0.4*weight

N_prob=0.4*(1-weight) + U_prob*weight

當用者端或系統端任一方沒有標記機率時，若直接採用公式一做計算，將使得判別用資料庫標記機率值嚴重傾向於一般郵件標記機率。為解決這種不合理的計算方法，其機率值設定為 0.4[3]。

3 系統實作與實驗結果

3.1 系統實作

本系統是在 Linux 系統環境下實作，使用者以 Linux 系統中原有已設定好的帳號及密碼登入到本系統內，再依個人的需求，設定系統端決定，或使用者自訂權重值。使用者除了權重值設定之外，並可以訂立白名單及黑名單，加速垃圾郵件過濾器的處理速度。在系統端的部份，我們在/etc/procmailrc 中設定將所有郵件轉向到本系統的垃圾郵件過濾器程式。在使用者端的部份，個人可透過 Web 介面，建立個人家目錄下的 .procmailrc 檔案，來做為垃圾郵件過濾個人郵件及執行相關動作。系統程式碼採用 Perl 語言，主要的原因是因 Perl 對於文字處理能力非常好，加上 Perl 程式可以跨平台，並且 Perl 在 Unix-like 系統下發展已很久，穩定性非常地好。

3.2 實驗結果

為了驗證本系統的各項功能，我們從分類資料庫大小對郵件判別之影響、權重參數調整對郵件判別之影響、及與自由軟體垃圾郵件過濾器之比較三個方面來做相關的實驗與比較。

3.2.1 分類資料庫大小對郵件判別之影響

由於本系統是以貝氏機率為基礎來做垃圾郵件的判別。因此在實際運作前，必須先分類出一般郵件及垃圾郵件二大部份，而一般郵件及垃圾郵件數量的多寡會對判別時產生極大的影響。本系統中所採用訓練及判別的郵件是由網路上極為著名的自由軟體垃圾郵件過濾器 spamassassin[18]及 SpamArchie[12]的網站上所公開的網路資料庫所取得，資料庫內含有多國語系，並且已分類好一般郵件及垃圾郵件。我們擷取出實驗用訓練郵件資料庫，含有一般郵件的數目 5214 封，垃圾郵件的數目 8537 封及實驗用測試郵件資料庫一般郵件 500 封、垃圾郵件 1000 封。並由實驗用訓練郵件資料庫，分別隨機取出 50 封、100 封、200 封、300 封、500 封、800 封、1000 封、2000 封、3000 封、5000 封做訓練的動作，在經過訓練的動作後，測試由實驗用測試郵件資料庫中，隨機挑選出的測試資料庫，含有一般郵件 200 封及測試用垃圾郵件 200 封，分別以本系統進行判別工作。

表 1 不同訓練郵件數下對垃圾郵件的判別率

一般郵件數 訓練用	垃圾郵件數 訓練用	垃圾郵件數 測試用	垃圾郵件數 判別	判別率
50	50	200	87	43.5%
100	100	200	120	60.0%
200	200	200	129	64.5%
300	300	200	140	70.0%
500	500	200	154	77.0%
800	800	200	171	85.5%
1000	1000	200	170	85.0%
2000	2000	200	171	85.5%
3000	3000	200	179	89.5%
5000	5000	200	179	89.5%

從表 1 中，可以明顯看出，當使用的訓練用資料庫愈大時，明顯可以看出判別率有著明顯的提升，尤其在訓練用資料庫達到 800 封之後，垃圾郵件正確的判別率已達到八成五左右。因此，以這種演算法來做垃圾郵件的判別，一定要有足夠量的訓練用資料庫量，才能夠使判別率有著明顯提升。由於一般郵件被判別錯誤情況重要性遠大於垃圾郵件的誤判，所以再次測試本系統對一般郵件是否可以正確判別出，實驗結果如下：

表 2 不同訓練郵件數下對一般郵件的誤判率

一般郵件數 訓練用	垃圾郵件數 訓練用	一般郵件數 測試用	垃圾郵件數 判別出	誤判率
50	50	200	2	1.0%
100	100	200	1	0.5%
200	200	200	0	0.0%
500	500	200	0	0.0%
1000	1000	200	0	0.0%
2000	2000	200	0	0.0%
3000	3000	200	0	0.0%
5000	5000	200	0	0.0%

由表 2 的結果看出，本系統對於一般郵件的誤判率極低，表示郵件不會被誤判為垃圾郵件的情況幾近是沒有，即使是在訓練的郵件數極少的情況下，表

現依然極為優異。

3.2.2 權重參數調整對郵件判別之影響

為驗證適性化的效果，本實驗設計為系統端訓練用使用一般郵件及垃圾郵件各 1000 封所產生的訓練用郵件資料庫做為系統端訓練用郵件資料庫。這裡採用 1000 封的原因，是由上述的實驗中得出，在垃圾郵件的判別上，800 封之後，整體的判別率已到達一個極限，為整體系統的負荷及速度上考量，這裡我們選擇取用 1000 封來當系統端訓練的資料庫。在使用者端訓練用郵件方面，則分別以 50 封、100 封、200 封、300 封、500 封、800 封及 1000 封的資料量，在不同權重狀況下來對系統做判別工作，權重值變化由 0~1 之間變化，每次增加 0.1。0 表示完全由系統訓練出來資料庫做判別，1 代表完全由使用者訓練出來資料庫做判別，數值愈大，表示以使用者端訓練的資料庫為主。針對七個不同資料量實驗出的比較表，將判別比率最高的權重值整理如表 3 所示。

表 3 不同訓練郵件數下，判別率最高時的權重值

一般/垃圾(郵件) 使用者訓練用	垃圾郵件數 判別	判別率	判別率 最高的 權重值
50/50	164	82.0%	0
100/100	157	78.5%	0.1
200/200	163	81.5%	0.1
300/300	170	85.0%	0.3
500/500	189	94.5%	0.7
800/800	193	96.5%	0.9
1000/1000	197	98.5%	1

由表 3 的結果，顯示出在不同訓練郵件數，在判別率最高的權重值，會隨著使用者端訓練郵件數的增加，而數值會漸漸由 0 往 1 增加變化，依照表 3 的變化，本系統的權重分配原則如表 4 所示

表 4 系統權重分配表

訓練郵件數目	權重值
100 封以下	0
100～200	0.1
200～300	0.3
300～500	0.6
500～700	0.8
700～800	0.9
800～1000	1

以此方式來分配權重，在使用者端尚未有任何分配好的郵件當做訓練的資料，判別率至少約可維持八成以上，直到使用者端達到 1000 封之後，系統則建議可以完全由使用者訓練的資料來做垃圾郵件判別的資料庫，以這種結合伺服器端及客戶端的方式達到適性化的目的。

3.2.3 與自由軟體垃圾郵件過濾器之比較

我們將自由軟體中採用純伺服器端軟體 Spamassassin[18]及純使用者端軟體 POPFile[15]與本系統來做比較。為比較三個垃圾郵件過濾器的判別率效果，在本實驗中採用了前二節中的實驗用測試郵件資料庫，從中隨機挑選 200 封郵件，分別由三種垃圾郵件過濾器來做判別工作，因為這三種垃圾郵件過濾器都有使用貝氏機率公式來判別垃圾郵件，因為先分別訓練 1000 封後，再開始進行判別。判別結果如表 5 所示。

表 5 垃圾郵件過濾器過濾效果比較表

軟體名稱	性質	垃圾郵件數 測試用	垃圾郵件數 判別	判別率
Spamassassin	伺服器端	200	182	91%
POPFile	客戶端	200	160	80%
適性化自動垃圾郵件過濾器	伺服器及客戶端合併	200	172	86%

由表 5 可看出，判別率效果最佳的是 Spamassassin 垃圾郵件過濾器，分析的原因，最主要原因是因為 spamassassin 雖原本只採用規則分數計算法的方式來做判別，但是最近系統中已加入貝氏公式來增加判別率，除此之外，此系統還有加上許多種判別方法，例如 razor 及 pyzor 的分散式黑名單郵件過濾方式…. 等等。在垃圾郵件過濾器的建置及處理上，當系統採用愈多過濾方法來做合併的過濾判別時，判別效果將會愈佳。但相對地會使系統的負荷工作量加大，降低整個伺服器的效能。

4 結論

本系統中垃圾郵件的判別處理，主要是以貝氏機率公式為方式來產生郵件過濾的規則。但是貝氏機率公式為基礎的垃圾郵件過濾系統，在缺乏足夠垃圾郵件的訓練量時，有著垃圾郵件判別率低的缺點。在系統中引進權重的觀念後，並結合黑、白名單的機制，可以改善此一缺點。系統依照權重分配的方法，配合系統端預設的資料庫，來達到任何的時間，使用者都可以有不錯的垃圾郵件判別率，並且可以隨著垃圾郵件的成長，隨時調整權重的比率及個人垃圾郵件的定義，達到較佳個人化的垃圾郵件過濾器，並且有著極佳的垃圾郵件過濾效果。

在系統中要達到更有效的垃圾郵件過濾，使用者必須對個人垃圾郵件做分類處理。此外，本系統中尚未結合 Web 郵件系統的使用，若能及早開發出來，本系統將可更加實用。再者，目前國際間已普遍採用 UTF 編碼，因此未來本系統需要增加 UTF 編碼處理，才能更符合時代的需求。

參考文獻

[1] Official SPAM Home Page, http://www.spam.com

[2] David Mertz, ”Spam filtering techniques”, http://www-106.ibm.com/developerworks/linux/library/l-spamf.html, September 2002.

[3] Paul Graham,“A Plan For Spam”,http://www.paulgraham.com/spam.html, August 2002.

[4] Tagged Message Delivery Agent, http://tmda.net/

[5] Anon Distributed Checksum Clearinghouse, http://www.rhyolite.com/anti-spam/dcc/dcc-tree/dcc.html, April 2001.

[6] Vipul’s Razor, http://razor.sourceforge.net/

[7] Pyzor, http://pyzor.sourceforge.net/

[8]Wilson, Ralph F. “20 ways opt-in e-mailers can outsmart spam filters” , http://www.wilsonweb.com/wmt8/spamfilter_avoidance.htm,December 2002.

[9] The Apache SpamAssassin Project, http://spamassassin.apache.org/

[10] Graham, Paul,”Better Bayesian Filtering”, http://paulgraham.com/better.html January 2003.

[11] N. Freed, “Multipurpose Internet Mail Extensions Part One:Format of Internet Message Bodies”, RFC 2045,November 1996.

[12] John Graham-Cumming, The Spammers' Compendium, http://www.jgc.org/tsc/ , April , 2005.

[13] Bayes' theorem, http://www.absoluteastronomy.com/encyclopedia/B/Ba/Bayes_theorem.htm

[14] Introduction to Bayesian Filtering, http://www.process.com/precisemail/bayesian_filtering.htm

[15] POPFile, Automatic Email Classification, http://popfile.sourceforge.net/

[16] Welcome to procmail.org, http://www.procmail.org/

[17] Donate Your Spam to Science, http://www.spamarchive.org/

[18] Welcome to Open WebMail Project http://openwebmail.org/

[19]Spamssassin Public Corpus, http://spamassassin.apache.org/publiccorpus/