香川大学経済論叢
第81巻 第2号 2008年9月　1－17

実用的なベイジアンスパムフィルターシステム

中村邦彦

Ⅰ はじめに

筆者は先に一般利用者向けのスパムメール対策として，Thunderbirdに組み込まれた迷惑メールフィルターの特性を分析し，その機能を効果的に利用するする方法を提案した[1]。ただし，Thunderbirdのベイジアンフィルターは日本語の処理に不十分な点があり，その本来の性能を十分に発揮できていない。そこで本稿では，日本語処理に優れたベイジアンフィルターを用いた迷惑メール対策を利用環境別にいくつか提案することを試みた。この場合，メーラーとは別にベイジアンフィルターを稼働させることになり，設定上でも，運用上でも少々手間が増えることは確かであるが，フィルターの性能がよいだけに，いったん設定し，安定運用に入ってしまえば運用の手間は減少して行くので，長期的には手間が減少するとも考えられる。使いやすく評価の高いものとして，ここではPOPfile[2]とbsfilter[3]を取り上げ，利用環境に合わせて次の4例を紹介する。

1．クライアント側で処理するPOPサーバー対応システム
2．クライアント側で処理するIMAPサーバー対応システム
3．サーバー側で処理する複数の利用者向けシステム
4．複数のメールアカウントを一括して処理する個人向けシステム

ここで，1と2ではPOPFileを，3と4ではbsfilterを用いた。利用環境は，

クライアントマシンがWindowsパソコン，サーバーはLinuxである。その他，メールサーバー（MTA）としてはPostfix，メールのローカル配送エージェントはprocmail，IMAPサーバーにはCourier-IMAPを使用している。

Ⅱ　スパムフィルター構築事例

2.1　クライアント側で処理するPOPサーバー対応システム

これはメールサーバーがPOPに対応し，クライアントもPOPに対応しているケースである。フィルターのPOPFileはPOPプロキシーとしてクライアントパソコン上で動作させる。POPというプロトコルの性質上，主としてメールの読み書きに使用するパソコンが1台の場合に向いている。POPFileに対応したメーラーとしてPOPFileのサイトでは，pochy，EdMax，電信八号，EdMax等を紹介しているが，ここではAL-Mail[4]を使用する例を紹介する。AL-Mailはシェアウェアではあるが，学生，教育機関を対象にした送金免除制度があり，かつては香川大学でもよく使われていたからである。またAL-MailにはPOPFileのためのプラグインAL-POPFile[5]があるので，POPFileの使い勝手はよい。

・POPFileのインストール

インストールはセットアッププログラムに従って進めていけばよい。途中で次のような，日本語分かち書きプログラムの選択ダイアログが表示される。

> Kakasi：漢字→かな（ローマ字）変換プログラム（推奨）
> 分かち書きの精度はMeCabに比べると低い（ひらがなやカタカナで構成されている単語の情報を持っていない）ですが，MeCabに比べ辞書サイズが小さくてすみます（MeCab 40 MBに対して2 MB程度）。
> Kakasiと辞書ファイルはインストーラに内蔵されています。
>
> MeCab：Yet Another Part−of−Speech and Morphological Analyzer

Kakasi よりもより正確な分かち書きを行うことができますが，辞書サイズが大きくなります（40 MB 程度）。
インストーラには内蔵されていません。MeCab オプションを選択した場合，およそ 13 MB のファイルがインターネットよりダウンロードされ，インストールされます。

内蔵パーサ：文字種による分割
外部プログラムを使わずに，文字の種類（漢字，ひらがな，カタカナなど）だけをたよりに分かち書きを行います。
辞書を使用した分かち書きに比べ分かち書きの精度は落ちますが，辞書を必要とせず，高速に動作します。

また，そのダイアログに，次のようにも書いてある。

※分かち書きの精度と POPFile の分類精度の間には直接の因果関係はなく，どのプログラムを使用した場合でも分類精度にはほとんど違いがないという結果が出ています。

気になるところだが，それは後で検討することにして，ここは推奨の KAKASI[6] を選択しておくことにする。

・コンポーネントの選択
IMAP を使いたい場合は IMAP をチェックする。このケースでは必要がない。

・バケツの作成
POPFile は実はスパムメールを振り分けるだけのフィルターではない。メールを内容により自動的に振り分けるツールであり，振り分け先のバケツを多数持つことができる。デフォルトでは spam，personal，work，other の 4 つが作

成されるが，少なくとも２つは必要なので，ここではspamとworkを残すことにする。(1)

・AL-Mailのインストール

特に問題になるようなことはなく，AL-Mailをインストール後，AL-POPFileをAL-Mailのpluginフォルダーにコピーする。これを利用することにより，サブジェクトに余分な文字を挿入させることなく，POPFileの判定に基づいてメールをAL-Mailのフォルダーに自動的に振り分けることが可能になる。また，AL-MailからPOPFileに誤判定を修正し学習させることができる。POPFileはworkとspamのバケツにそれぞれ少なくとも１つ以上学習したメールがないと，すべてUnclassifiedに分類するので，まずは１通ずつ学習させることを勧める。

筆者はしばらくこのシステムを試用していたが，POPFileの判定はかなりよいと言える。ときどきPOPFileが正しくスパムと判定しているにもかかわらずAL-Mailがスパムとして認識しない場合があった。調べてみると，その種のメールはサブジェクトの中にEOFコードが埋め込まれていた。AL-POPFileはEOFを検出した時点で処理を中止したために振り分け処理が行われていなかったのである。その問題は現在のAL-POPFileでは修正されている。

2.2　クライアント側で処理するIMAPサーバー対応システム

メールサーバーがIMAPに対応している場合は，POPFileのIMAP機能を利用することにより，サーバーにあるメールボックス内でメールの振り分けを行うことができる。メールを複数の異なるパソコンで読み書きする場合に適している。IMAPを利用する場合は，POPFileはIMAPプロキシーとして働くのではない。POPFileはメールクライアントとは独立に，定期的に処理をしている。

（１）　バケツの数をある程度多くする方がよいとする説[9]があるが，ここでは単純に２つだけにした。

IMAP 方式のメリット

・振り分けを学習させるには，メールクライアント側でメールを移動させる。POPFile の UI（ユーザーインターフェイス）を使う必要がない。

・振り分けは POPFile が定期的に行うので，ダウンロード時に待たされない。

・複数の環境からアクセスする場合でも，POPFIle は一ヵ所に設置するだけでよい。

IMAP 方式のデメリット

・定期的な振り分けの前にアクセスすると，その時点では振り分けられていない（更新間隔の初期値は 20 秒になっている）。

・POPFile 起動中はひとつの IMAP コネクションが開いたままになる。

POPFile のインストールは，コンポーネントの選択画面で IMAP をチェックすることを除いて POP サーバーの場合と同じである。

POPFile インストール後の設定は UI から次のように行う。

- 詳細設定の　imap_enabled を 1 に，pop 3__enabled を 0 にして POPFile を再起動する。
- 設定ページの IMAP で次の設定をする。
  - ・IMAP サーバーのホスト名
  - ・IMAP サーバーのポート番号
  - ・SSL が利用できる場合は SSL を使用する
  - ・IMAP アカウントのユーザー名
  - ・IMAP アカウントのパスワード
- フォルダーリストの更新ボタンを押す
- 監視フォルダー 1 を設定。通常は INBOX を指定して適用ボタンを押す
- 各バケツに分類されたメールの移動先を設定
- 適用ボタンを押す
- フォルダーリストの更新

メールクライアントにはThunderbirdを使ったが，IMAP対応のメーラーであれば利用可能である。IMAPサーバーにはCourier-IMAPを使った。

スパムフォルダーはThunderbirdではINBOX.Junkになる。スパムでないものをINBOXに残すことにしてPOPFileのUnclassifiedを未分類というフォルダーに置くことにする。Thunderbirdで「未分類」フォルダーを作ると，POPFile内では文字化けが発生して「INBOX.&ZypSBphe-」のように表示されたが，そのままでうまく動作した。監視フォルダー，バケツと移動先フォルダーは次のように設定した。

監視フォルダー１　INBOX
'spam'　バケツに分類されたメールの移動先　→　INBOX.Junk
'unclassified'　バケツに分類されたメールの移動先
→　INBOX.&ZypSBphe-（未分類）
'work'　バケツに分類されたメールの移動先　→　INBOX

この状態でPOPFileを起動しておく。振り分けの学習はThunderbirdから，未分類になったもの，判定が間違っているものを正しいフォルダーに移動することにより行える。したがって，通常の利用ではPOPFileのUIを開く必要がなく，使い勝手はよい。なお，メールをPOPFileのUIで再分類しても，実際のメールは移動しない。

2.3　サーバー側で処理する複数の利用者向けシステム

ここでは多数の利用者がそれぞれ固有のベイジアンフィルターを利用できる環境を作成する。ベイジアンフィルターとしてはbsfilterを使う。メールのローカル配送にprocmailを使い，メールが到着した時にbsfilterにかけてスパムか否かを判定し，結果をヘッダーに追加する。procmailはその結果により，スパムであればスパムフォルダーへ振り分ける。

bsfilterのインストールは，Debian etchではパッケージが用意されているの

でそれを利用した。bsfilter は ruby で書かれているので，ruby も必要である。

bsfilter は日本語の分かち書きの方法として bigram, block, MeCab[7], ChaSen, KAKASI をサポートしている。ここでは KAKASI を利用した。ruby から KAKASI を呼び出すインターフェイス libkakasi-ruby 1.8 も Debian のパッケージに含まれている。

bsfilter は標準では個人のホームディレクトリー内に .bsfilter というサブディレクトリーを作成し，そこにデータベースと設定ファイルを置く。

設定ファイル bsfilter.conf の内容

```
jtokenizer kakasi
insert-flag
insert-probability
spam-cutoff 0.7
```

insert-flag によりメールヘッダーに “X-Spam-Flag : ” が追加される。値は Yes または No である。Insert-probability によりメールヘッダーに，“X-Spam-Probability : 0.5” のようにスパム確率が追加される。spam-cutoff によりスパムの判定基準を設定する。ここでは 0.7 にしたが，これは最初に簡単なテストを行って決めた。今のところこれでうまく行っているように見える。

個人の .procmailrc には次の記述を追加する。

```
:0 fw
| /usr/bin/bsfilter --pipe

:0
* ^X-Spam-Flag:Yes
```

. Junk/

メールの読み書きはIMAP対応のメーラーを使えば問題ないが，メール振り分けの学習をどうするかが問題である。いくつか検討したがフリーのウェブメールツールのSquirrelmail[10]にSpam-Buttons[11]というプラグインがあり，これを使うことにした。これを組み込むとSquirrelmailの操作画面に「スパム」と「スパムでない」という2つのボタンが追加され（図1），判定が間違っている場合には学習させることができる。学習は判定が間違っている場合にのみ行わせた。

図1　Spam_Button プラグインにより SquirrelMail に追加されたボタン

ウェブサーバーから個人のデータを更新することになるので，sudoを使ってbsfilterを利用者の権限で実行できるようにする。設定はSpam_Buttonのconfig.phpに記述する。

スパムとして学習させるコマンド（実際には1行に記述する）

```
sudo -u ###USERNAME### /usr/bin/bsfilter
--homedir=/home/###USERNAME###/.bsfilter -u -s
```

非スパムとして学習させるコマンド（実際には1行に記述する）

```
sudo -u ###USERNAME### /usr/bin/bsfilter
--homedir=/home/###USERNAME###/.bsfilter -u -c
```

###USERNAME###は，実行時に実際のユーザー名に置き換えられる。

/etc/sudoers への追加

```
www-data    ALL=(ALL) NOPASSWD: /usr/bin/bsfilter
```

Spam_Buttons から学習させると次のような奇妙な日本語が表示されて微笑ましい。

「順調にスパムじゃないのように届け出ました」

このシステムでは，通常は Thunderbird を使うことを想定している。メールは Thunderbird を起動したときにはすでに分類されているが，その分類が間違っているときはウェブブラウザーで Squirrelmail を呼び出して修正しなければならない。その点は使い勝手が悪いが，操作は該当するメールをチェックしてボタンをクリックするだけなので操作性は悪くない。学習が進めば誤判定は少なくなるので実用上は許される範囲である。もっとも，メーラーは Squirrelmail だけで十分という場合は Thunderbird を使う必要はないし，学習が進むまでは Squirrelmail をメインに使うという対応も考えられる。

多数の利用者に利用させるシステムでは，スパムフィルターを使わない利用者への配慮が必要になるかもしれない。その場合は初期設定でスパムボタンを表示しないようにしておき，希望者だけが表示できるようにすればよい。その修正の例を示す。

・Squirrelmail の defaut.pref に次の行を追加する。

```
sb_enable_or_disable=0
```

・オプション設定のページに Spam Button を使うか使わないかのオプションを追加

function spam_buttons_display_options_do()　の中

```
$sb_enable_or_disable = getPref($data_dir, $username,
                'sb_enable_or_disable', $sb_enable_or_disable);

$my_optpage_values[] = array(
   'name'          => 'sb_enable_or_disable',
   'caption'       => _("Spam Button Enable"),
   'type'          => SMOPT_TYPE_BOOLEAN,
   'initial_value' => $sb_enable_or_disable,
   'refresh'       => SMOPT_REFRESH_NONE,
);
```

・config.php の $show_spam_buttons_on_message_list =1 ; となっているところを修正

```
$show_spam_buttons_on_message_list = getPref($data_dir, $username,
                    'sb_enable_or_disable', $sb_enable_or_disable);
```

2.4　複数のメールアカウントを一括して処理する個人向けシステム

通常なら2.3のようなシステムでよいのだが，筆者の場合は複数のメールアドレスがあり，それぞれにベイジアンフィルターを用意して学習させるのは手間である。そこですべてのメールを専用のアカウントに集めてそこに判定用の共通のデータベースを作成し，個々のアカウントのメールの判定はそのデータベースを利用して行うようにした。

設定は個々に処理する場合とほぼ同じであるので省略する。

実際にすべてのメールをひとつのアカウントに集めなくても，判定用のデータベースだけを共通にすることも考えられるが，今回は一ヵ所に集めて学習させた。このようにすればすべてのメールについての学習を一ヵ所で行うことができてよいのだが，異なるアカウントにコピーして配送するだけでもヘッダー

表1 5ヵ月間のスパム確率の頻度

スパム確率	非スパム	スパム
0.0～	1,332	0
0.1～	158	10
0.2～	147	0
0.3～	175	0
0.4～	912	1
0.5～	1,728	52
0.6～	52	17
0.7～	1	66
0.8～	1	126
0.9～	2	13,989
1.0	0	96,460
計	4,508	110,721

部分が修正されてしまう。そのため，学習用のアカウントでは正しく判定されていても，個々のアカウントでは誤判定される場合がまれに発生する。

このシステムは2007年11月末から使い始めた。12月はトレーニング期間として除外し，2008年1月から5月までの5ヵ月間のスパム判定率を求めたところ99.9%であった。スパムでないものでスパムと判定されたものは，サブジェクトなし，本文なしなどの学生からの無作法なメールの他，C言語のソースプログラムを含むもの，本文にCSV形式のデータを含むものなどがあった。プログラムやデータは少量でも添付ファイルにする方がよさそうである。表1にスパム，非スパムのスパム確率の頻度を示す。網掛けは誤判定にあたる。ただし，これはスパム確率0.7を閾値にして学習させた結果であることに注意してほしい。

なお，このシステムでは，これまで学習させたメールはスパムが181通，非スパムが65通，データベース内の登録単語数は23,343，データベースのサイズは約2.3 MBになっている。

Ⅲ 分かち書きツールによる違い

分かち書きツールによる違いを検討してみた。

3.1 処理速度

MeCabのウェブサイトには「平均的にChaSen, Juman, KAKASIより高速に動作します」と書いてある。KAKASIより速いのであれば使ってみたいと思って速度を比較してみた。メールは日本語と英語を合わせて2,000通を用意して，bsfilterで処理した結果を示す。

表2 bsfilterにおける分かち書きツールの比較

比較項目	bigram	KAKASI	MeCab
単語学習(秒)	30.9	36.5	58.6
DB更新	8.1	6.9	6.8
学習時間	39.0	43.4	65.4
英語単語数	17,611	17,611	17,611
日本語単語数	38,412	30,971	29,182
DBサイズ(KB)	5,244	4,984	4,944

これで見る限り，筆者の環境ではMeCabはKAKASIより遅い。後に見るように分かち書きの精度が高いので余分な単語が少なくなり，データベースのサイズは小さくなるが，単語学習には約1.6倍の時間がかかっている。

3.2 分かち書きの精度

POPFileにおいて，分かち書きツールの違いはフィルターの精度に出ないという報告がある[15]。これは意外な感じがする。そこでその分かち書きツールを比較してみた。元にしたのは次のような簡単なメールである。

Subject：通常のメール
香川一郎様

いつもお世話になっています。
この度はスパムメールの実験に協力いただきありがとうございます。
このメールは普通のメールのつもりです。
よろしくお願いいたします。

−−

香川大学経済学部

中村邦彦

これをPOPFileの内部パーサーとKAKASIにかけたものを比べたのが表3である。

表3 POPFileの内部パーサーとKAKASIによる分かち書き単語

POPFile Internal	KAKASI
。	。
いいたします	いたします
いただきありがとうございます	いただきありがとうございます
いつもお	いつもお
この	この
	に
になっています	になっています
の	の
のつもりです	のつもりです
は	は
よろしくお	よろしくお
スパムメール	スパムメール
メール	メール
一郎	一郎
学部	
願	願い
協力	協力
経済	経済学部
香川	香川
実験	実験
世話	世話
大学	大学
中村	中村
通常	通常
度	度
普通	普通
邦彦	邦彦
様	様

後で示すように，さすがにMeCabは少し違うが，POPfileの内部パーサーとKAKASIの分かち書きの結果はほとんど同じである。

次に，bsfileとPOPFileの学習単語を比較してみた。KAKASIの結果を処理したものを表4に，MeCabの結果を処理したものを表5に示す。表の○印は学習されたことを表す。「願い」はbsfilterでは「願」として学習された。

表4　bsfilterとPOPFileによる学習単語の比較1

KAKASI	bsfilter	POPFile
。		
いたします		○
いただきありがとうございます		○
いつもお		○
この		○
に		
になっています		○
の		
のつもりです		○
は		
よろしくお		○
スパムメール	○	○
メール	○	○
一郎	○	○
願い	願	○
協力	○	○
経済学部	○	○
香川	○	○
実験	○	○
世話	○	○
大学	○	○
中村	○	○
通常	○	○
度	○	○
普通	○	○
邦彦	○	○
様	○	

表5 bsfilterとPOPFileによる学習単語の比較2

MeCab	bsfilter	POPFile
。		
ありがとう		○
い		
いたし		○
いただき		○
いつも		○
お願い	○	○
お世話	○	○
この		○
ござい		○
つもり	○	○
て		
です		○
なっ		○
に		
の		
は		
ます		○
よろしく		○
スパムメール	○	○
メール	○	○
一郎	○	○
協力	○	○
経済学部	○	○
香川	○	○
香川大学	○	○
実験	○	○
中村	○	○
通常	○	○
度	○	
普通	○	○
邦彦	○	○
様	○	

意外なのは，bsfilterはひらがなの単語を学習しない傾向が強いことである。この例では学習したひらがな語は「つもり」だけである。実際，筆者の使用しているbsfilterの学習済み単語の中にはひらがなを含むものは皆無であった。それに対してPOPFileはひらがな語をよく学習しているが，1字の単語は漢字であろうとひらがなであろうと捨てているように見える。

以上から，POPFileにおいて，内部パーサーとKAKASIの間で差が出ない理由は明らかである。一方，MeCabとKAKASIとで異なるのはひらがなを含む語だけと言ってよい。両者で差が出ないというのなら，ひらがな語はスパム判定への寄与が小さいと言えるのではないか。ひらがな語を全く学習していないbsfilterが好成績をあげていることからもその可能性は高いと言えよう。

Ⅳ むすび

ベイジアンフィルターを利用したスパムメール振り分けシステムをいくつか紹介した。フィルターは日本語環境に十分対応しているものを使えば，かなりの好成績でスパムを排除できることが確認できた。ベイジアンフィルターによるスパムの振り分け[12][13][14]は，理論的にはすっきりしているが，いざ実装しようとするとさまざまな工夫が必要になる。本稿で取り上げたPOPFileとbsfilterについてだけでも，確率の計算方法，日本語メールそのものの取り扱い，日本語の分かち書き，学習する単語の選択等において，それぞれ独自の工夫が施されている。また，それを実際に使用する場合はユーザーインターフェイスを工夫する必要もある。

今回調べてみて意外だったのは，筆者の使用しているbsfilter + KAKASIによるフィルタリングシステムは，好成績をあげているにもかかわらず，ひらがな語をまったく学習していなかったことである。一方で，ひらがな語を多く学習しているPOPFile + MeCabのシステムはKAKASIを使った場合と差がないと言う。ひらがな語の扱い方を検討してみる余地がありそうである。

参 考 文 献

[1] 中村邦彦,「Thunderbird の迷惑メールフィルター」, 香川大学総合情報センター年報, 第5号, 2008年2月

[2] John Graham-Cumming, "POPFile" : http://getpopfile.org/

[3] nabeken, "bsfilter / bayesian spam filter / ベイジアン スパム フィルタ" : http://bsfilter.org/

[4] AL-Mail : http://www.almail.com/

[5] AL-POPFile （AL-Mail 32 用プラグイン） : http://www.crosstech.co.jp/al-popfile/

[6] KAKASI 漢字→かな（ローマ字）変換プログラム : http://kakasi.namazu.org/

[7] MeCab : Yet Another Part-of-Speech and Morphological Analyzer : http://mecab.sourceforge.net/

[8] ChaSen 形態素解析器 : http://chasen-legacy.sourceforge.jp/

[9] popfile の設定方法 : http://www.tatsuya.dnsalias.com/cgi-bin/soho/index.php?popfile

[10] SquirrelMail : http://www.squirrelmail.org/

[11] Spam_Buttons SquirrelMail Plugin : http://www.squirrelmail.org/plugin_view.php?id＝242

[12] Paul Graham, "A Plan For Spam" : http://www.paulgraham.com/spam.html

[13] Gary Robinson, "Spam Detection" : http://radio.weblogs.com/0101454/stories/2002/09/16/spamDetection.html

[14] Gary Robinson, "A Statistical Approach to the Spam Problem" : http://www.linuxjournal.com/article.php?sid＝6467

[15] POPFile/Accuracy : http://amatubu.skr.jp/?POPFile/Accuracy