迷惑メール配送停止システムの一般公開を目指した開発と運用に関する研究

Researches on unveiling spam mail protection system

システム工学部：○和田俊和，松浦広明，斎藤彰一，加藤丈和

T. WADA, H.MATSUURA, S.SAITO, T.KATO

Faculty of Systems Engineering

○印研究代表者連絡先：twada@sys.wakayama-u.ac.jp，電話073-457-8079

本研究に関連するホームページＵＲＬ：http://vrl.sys.wakayama-u.ac.jp/~twada/NNIPF.html

要約：本研究は，我々が開発してきた迷惑メール配送停止システムを学内外に公開するためにインフラの整備と運用技術を開発することを目的としている．我々の開発したシステムは当初の目的どおり完成し，学内外で稼動している．しかし，情報学センターによる迷惑メール配送遮断措置が当プロジェクトの進捗に大きな打撃を与え，当初の方針通り研究と普及を進めることが困難になったため，よりポータブルで精度の高いソフトウエアNNIPFを開発し，対外的に公開・普及した．現在までのダウンロード数は数千回を超え，社会的に大きなインパクトを与えた．

１．はじめに

電子メールは，日常業務や日常生活に欠かせないツールとして生活に定着している．便利なツールとして使われる一方で，マスメール(SPAM，迷惑メール，ジャンクメール)が大きな社会問題となっている．マスメールとは，不特定多数の受信者に送られた大量のメールの中で，受信者が意図せずに受け取るメールのことであり，その内容には主に，

・商品やサービスの宣伝を目的としたもの

・詐欺を目的としたもの

・メールアドレスそのものの収集を目的としたもの

などが含まれている．これらは，公序良俗に反するサイトへの誘導，受信者に対する心理的・経済的な不利益の誘発，メールトラフィックの増大，システム運用管理への弊害，など様々な問題を引き起こしている．このように，マスメールはインターネット上に留まらない大きな社会問題となっており，マスメールを完全に検出する技術の発展が望まれている

本研究では，マスメールの検出と除去をメール配信システム(Mail Transfer Agent:MTA)の側で行なうシステムを開発し，これを学内及び学外に普及することを目的としていた．このシステムは，当初の目的どおり完成し，現在も和歌山大学システム工学部および，4次元データ株式会社内で稼動している．しかし，情報学センターにTransWare社の迷惑メール除去システムActiveHunterが導入されたため，我々のシステムの必要性が理解されずセンターへの普及が行なえなくなった．また，同センターで，大学の対外接続ルータ側でメールの通信を一旦トラップして，通信パターンから迷惑メールと思われる通信を遮断する措置が講じられた．この結果，大学に流入する迷惑メールの数が減少し，判定の元となるデータベースに蓄積される迷惑メールの収集量が減少したため，我々のシステムの判定精度が低下してしまった．このように，情報学センターの活動が当プロジェクトの進捗に大きな打撃を与え，当初の方針通り研究と普及を進めることが困難になった．

そこで，当プロジェクトでは，対外的な普及・発表活動を一時停止し，よりポータブルで精度の高いソフトウエアの開発を行うことに注力し，この結果できあがった成果をフリーソフトウエアとして対外的に公

開・普及することを新たな目標として再設定した.

ActiveHunterのようにメールのボディ部に基づいて識別する方式では正常なメールが迷惑メールとして誤検出されることが避けられないため，新方式のソフトウエアNNIPFではメールの先頭部分にあるヘッダーと呼ばれる部分を高精度に解析し，迷惑メール送信者のIPアドレスを正確に割り出す方式を採用した．さらに学習が瞬時に終了し，識別も高速な最近傍識別器でこのIPアドレスを識別する方式を開発した．これらに基づくソフトウエア（NNIPF）を2007年3月末に完成させ，4月６日からホームページにおいて一般公開し，現在，九州大学，京都大学，4次元データ(株)，Palmo Linuxなど様々な組織でテストあるいは運用が行なわれている．

このNNIPFはWEB上で大きく取り上げられ，一時期はGoogleによる検索だけでも，8万件以上のWRBページにヒットする状況になった．公開後6日の時点でWEBサーバのログを調べたところ，学外195箇所からソフトウエアのダウンロードが行なわれており，現在までのダウンロード回数は数千回を超えている．

以下，既存のマスメールフィルタ，我々が開発し，普及を進めようとしてきた事例ベース迷惑メールフィルタ，新たに開発を行なったシステムのそれぞれについて述べる．

２．既存のマスメール検出方法

マスメールの検出方法として，様々なものが研究・運用されてきた．そのいくつかを以下に列挙する．

2.1 MTA における配送遮断や抑止

これまでマスメールの配送は，不正利用の対策の施されていない MTA を踏み台として行われる例が多かった.このような MTA の IP アドレスをブラックリストとして収集しておき，自分の管理する MTA にてリスト中の MTA からの配送を拒否するという手法がある[1][2][3].

しかし，この手法は正常な設定の MTA が誤判定によりブラックリストに掲載されてしまう欠点があることや，最近のマスメールは踏み台を用いずに送られてくることが多いことから，この手法の有効性は低くなっている．

2.2 プロトコルレベルでの配送遮断

多くの MTA においては，相手からの応答が返ってこないときは数秒のあいだ応答を待つように設計されている．また，メールの送受信ができなかったときは時間をあけて再送信するように実装されている[4]．しかし，マスメール送信業者が使う MTA(送信ツール)では,送信先から応答が返ってこなかったり,送信先サーバがメールを受け付けなかったりした場合，その送信先を無視するようになっていることが多い[5][6].そこで,自分の管理する MTA の設定を，応答を遅らせるようにしたり，一旦受信を拒否したりすることで，マスメール送信業者の MTA に無視されるようにすることができる．

この手法は，送信ツールで送られてくるマスメールには強力であるが，そうでないマスメールには全くの無力であるという欠点がある．

2.3 ルールベース判定法

マスメールに現れる特徴(ルール)を抽出し，スコアリングを行ってマスメールを検出する手法がある．この手法に基づくツールの代表的なものが SpamAssassin[7]である．この手法は，マスメールの内容による判定ができるため，前述の手法より多くのマスメールを検出することができる.また,個々のユーザによらない一般的判定のみを行うため，ユーザごとの設定が不要であり設定と管理が容易である．その反面，メールの特徴がマスメールに似ていると誤判定(False Positive)されやすい．

2.4 識別型判定法

各ユーザに通常のメールとマスメールを集積したコーパスを用意させ，それぞれから特徴を抽出してマスメール判定の指標を自動的に作成(学習)し，その後届いたメールについてはその指標に基づいて自動的にマスメールを検出するシステムが提案されている.この手法の代表的なものが P.Graham によって提案された，Bayesian Filtering である[8]．この手法はルールベースによる手法よりも高い精度でマスメールを判定できる．しかし，学習次第で検出精度が向上する反面，初期の学習過程においてユーザにメールのコーパスを用意させるという負担を強いる欠点もある．また，検出精度を保つため，使用中も再学習し続ける必要もある．学習過程や判定における処理量も他の手法に比べると大きい．この手法でも False Positive が生じ得るということである．

2.5 事例ベース判定法

既知のマスメールを大量に収集してデータベース化し，新たなメールが届くたびにこのメールをデータベースと比較することによってマスメールを検出

する手法が考えられる．この手法に基づくシステムには，Razor[9]や Pyzor[10]などがある．マスメールのデータベースを全ユーザで共有することで，各ユーザに学習や分類の負担を減らすことができ，また，実際のマスメールとの比較により判定を行うため False Positive が生じる可能性が極めて小さい．ただし，マスメールにランダム要素が入っている場合，検出率の低下は免れないため，いかにうまくランダム要素を取り除くかが重要となってくる．

この手法は，マスメールのデータベースを共有することで，広範囲で同性能のマスメールの検出システムを運用することができるという利点や，演算負荷が小さいためにサーバサイドでの運用に適しているという利点がある．

３．我々が開発してきたマスメールフィルタ

この章では，我々がこれまで開発を進めてきたマスメールフィルタについて述べる．

3.1 事例ベース型マスメール駆除システム

ここで述べるのは，メールコンテンツを事例と見なしてマスメールをフィルタリングする方法である．しかし，我々は，個々のメールをコンテンツによって識別する問題そのものは主題として取り扱わない．むしろ，これらの方法，あるいは人手によって発見されたマスメールコンテンツのチェックサムを計算して公開し，これを参照することによって，マスメールを排除する方式について検討を行う．この方式は，

１） マスメールは基本的に同じ内容のメールが大量に送受信されるため，同種メールの識別を何度も繰り返す必要はない．
２） 同じデータベースを参照している集団には，登録されたメールが届かないようにすることができる．
３） クライアントソフトウエアは MTA や POP・IMAP サーバだけでなく，POP・IMAP クライアントやメール整理ツールまで，幅広く対応でき，包括的なマスメールフィルタリングが実現できる．
４） 一度データベースに登録されたものが抹消されることは間違いを除いてありえない．そのため，ローカルキャッシュを参照することができ，データベースの負荷を必要以上に上げないアプリケーションの構築が行える．

等の利点がある．特に，１） ２）は集団的なマスメール駆除を行うことのメリットであり，３）もまた個別の識別システムでは実現することが困難な機能である．また，このデータベースの作成過程でマスメールコンテンツ自体を収集しているため，識別型フィルタの学習データも副産物として得られる．

BT の顧客に対する SPAM フィルタリングや，AOL の新クライアントに Anti-SPAM 機能が盛り込まれるなど同種のマスメールフィルタリング方式はすでに一部実用化されているが，日本語テキストへの対応や，特定組織内での利用など制限が多いためまだ十分には広まっていない．

このようなデータベースを収集するには，マスメールを効率よく集めてこなければならないがその方法は必ずしも明確ではないという問題がある．

また，個々のマスメールのボディ部分にも送信相手に固有の文字列や無意味な文字列が埋め込まれているケースが多く，単純なチェックサム計算では同種のマスメールに対して異なるチェックサムしか得られず問題となる．

さらに，チェックサムによってマスメールと判断された場合に，いきなりメールを削除もしくは受け取らない，などの二者択一的アクションを行わず，誤ってマスメールと判定される正常なメールがあり得る事を考慮したユーザインタフェースを提供する必要がある．

以下，これらの問題に対して検討を行った結果について述べ,続いてシステムの概要について述べる．

3.1.1 マスメールデータベース構築

マスメールデータベース構築に当たって，必要になることは，以下のように整理できる．

- できる限り多種のマスメールコンテンツを，流通しはじめてからすぐに収集し，データベースに反映させること．
- 収集したメールの内容のうち，送り先に固有の情報や解析を避けるために埋め込まれたランダムな文字列を回避して，マスメール毎に固有のチェックサムを求める方法の開発．
- マスメールデータベースを参照してメールをブロックするクライアントの開発

以下，上記の内容について説明する．

マスメールの収集

マスメールの収集は，現在以下のようにして行っ

ている.

- 現在マスメールを受信しており，実質的な使用者が存在しないメールアドレスから転送を行う.
- 囮メールアドレスを WEB ページのコメントもしくは,実質的にクリックできない mailto アンカーに埋め込む.
- 囮メールアドレスに関して，マスメール送信業者にメール送信を行わない旨のメールを送る.
- 新規ドメインを取得し，webmaster@domain, info@domain などそのドメイン内の全てのメールアドレス宛のメールを収集する.
- 当該プロジェクト協力者宛のメールを分析対象とする.

上記の方法で，メール（マスメール候補）を収集した後，以下のようにしてメールコンテンツの分析および登録を行う.

- コンテンツチェックサムデータベースにすでに登録されているか否かをチェックする.
- 登録されていれば，すでに登録されている内容へのメールとしてデータベースに格納する.
- 登録されていなければ，メールコンテンツの識別を行い，マスメールと判定された場合には新規マスメールコンテンツとしてデータベースに登録を行う.
- マスメールと判定されなかった場合は，メールスプールに書き込む.

という順序で登録処理を行う．メールコンテンツの識別方法については，ヘッダ部分に着目して分析を行う方法と，メールボディ部分を解析して識別する方法の２つが考えられるが，現時点では，メールボディの識別は開発中であり，ヘッダ情報（中継 MTA の IP アドレス情報）に基づいて，識別を行う方法を実装している.

チェックサムデータベースの構築

マスメールのボディ部分は，下記の理由により，同じコンテンツであっても異なる文字列となっていることが一般的である.

1. base64, quoted-printable などのエンコーディングがかかっているケースがある.
2. ユーザ毎に割り当てられた特殊な ID 文字列や，ユーザのメールアドレス，あるいは HTML のコメントタグなどが挿入されているケースがある.

このうち，1. に関しては，デコードをして対処する.

2. に関しては，正規文法として受理すべき文字列を決め，下記のヒューリスティクスを使い，メールコンテンツから不要な文字列を削除している[1].

- 英数字の混じった文字列
- コントロールコード
- アンカーURL 以外の HTML タグ
- メールヘッダの Delivered-To フィールドに現れる文字列

上記の正規化を行った後に MD5 によってチェックサムの計算を行っている.

MD5 は異なるデータに対して同じチェックサムが得られる可能性が極めて低く，実質上確率０と見なしてよいこと，および，元の正規化された文字列が復元できない，という条件が満たされているため，当システムのチェックサム計算法としては最も妥当なものであると考える.

このようにして得られたマスメールのチェックサムデータは DNS に自動的に登録され，公開されている.

3.1.2 マスメールの取り扱い

上記データベースを参照して動作するクライアントプログラムとしては,多数のものが考えられるが,現在は単純なメールフィルタ，POP サーバ，POP クライアント，などを開発している.

メールフィルタ： メールチェックサムを計算して DNS に対する問い合わせを行う．その結果に従い，メールの通過とブロックを行う．この実装の欠点は，ブロックしたメールに正常なメールが含まれていた場合に対処することができないということである.

POP サーバ： POP サーバでチェックサムを計算し，DNS を参照してマスメールと判定された場合は，マスメールである旨を表現するメールヘッダを挿入する．メールリーダの側でこのメールヘッダに基づいて自動分類するようにしておけば，仮に誤って正常なメールがマスメールと判定されてしまっても，ユーザはそのメールを失うことはない.

POP クライアント： 上記 POP サーバで行うのと同様の内容を POP クライアント側で行う.

[1] チェックサム計算の詳細アルゴリズムは，これを回避する措置が取られないよう，詳細については公開できない.

これらいずれの場合も,不正中継 MTA の IP アドレスが Received From 行に現れた場合にマスメールとして検出するという機能も追加できる．また，後者のチェックによってマスメールが検出された場合，DNS への自動登録を行うということも実現できる．このように，クライアントプログラムにデータベースへの自動登録機能を組み込み，それが安全に運用できれば，結果的にチェックサムの参照によるマスメール検出の確率が増加すると考えられる．

3.1.3 運用実績

上述の事例ベースフィルタは,「1)マスメール収集サーバ」と「2)メールサーバ上で動作する検出システム」のふたつからなるシステムである．1)で収集したマスメールから URL などの特徴を抽出してデータベース化しておき，2)において通過するメールに対して同様の特徴を抽出してその存在をデータベースに問い合わせることによって，そのメールがマスメールかどうかを判別する．このシステムは，和歌山大学システム工学部において，2003 年の 12 月より運用を行っている．

図 1 は,2004 年～2006 年にかけてのマスメール検出結果である．2004 年の間は，一日に平均 500 件程度の検出であったが，2005 年に入り急激にマスメールが増加し,検出数が 3000 件を超える日も少なくなかった．2006 年の 4 月末に和歌山大学のゲートウェイにおいて，プロトコルによる配送遮断を行うことで,マスメールの数は再び 2004 年の水準まで減少した．しかし,夏以降ふたたび増加の傾向が見られる．

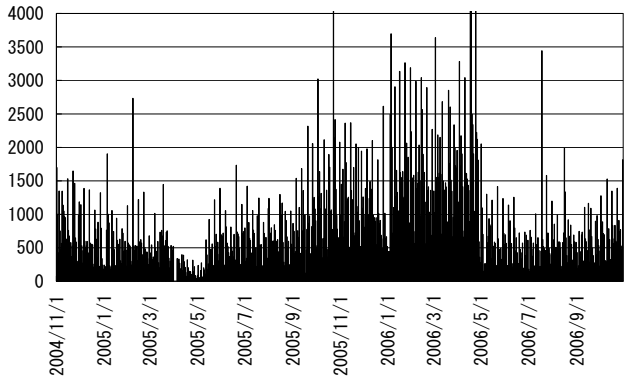

図 1．過去 2 年間のマスメール検出状況

3.2. 識別型フィルタの開発[14]

本プロジェクトでは，上記マスメールフィルタリングシステムを運用しつつ，以下に述べる新たな高精度マスメールフィルタリング手法についても開発してきた．

内容のよく似たマスメール同士を比較すると，それぞれのマスメールが同一のテンプレートを元にして作成されていることが分かる．すなわち,「文面や文構造のほとんどの部分が共通しており，一部にランダムな文字列や故意のスペリングミスが加えられている」といった特徴が見られるのである[15]．このような,類似したマスメールを検出するためには，文書を分類する技術や，類似文書を検索する技術が応用できると考えられる．

3.2.1 圧縮率に着目した類似文書分類手法

類似文書を分類する方法として，Benedetto らは，二つの文書を結合した文書の圧縮率から文書間の関係・類似度を測定する手法を提案した[16]．この手法は，「２つの文書が似ているのであれば，Lempel-Ziv の手法(LZ77 法)[17]により抽出される共起文字列も必然的に似ており，２つの文書を結合したファイルの圧縮率は高くなる」という原理を利用している．このような圧縮率に基づく文書分類手法については，国内でも様々な研究がなされている[18][19][20]．

この手法がマスメールの分類に用いることができるかどうかを確かめるために，次のような実験を行った．

まず，500 通のマスメール(メール 001～メール 500)を用意し，その中からメール a とメール b の任意の 2 通をとりだした．メール a とメール b の 2 つのメールを連結させてファイル Fab を作成し，これを gzip で圧縮し，圧縮率 Zab を求めた．

圧縮率 Zab がそのまま類似度を意味しているのではない．なぜなら，圧縮率 Zab にはメール b の圧縮率が反映され，類似度による影響よりも大きく寄与しているからである．そこで，類似度を求める指標を導入する必要がある（同様の指摘および類似度指標の導入は，他の研究でもなされている[18]）．本研究では以下のようにして類似度指標 Rab を計算した．なお，この指標は類似度を大まかに把握するためのものであり，数学的に厳密に正しいわけではない．

(1) Zab が最大値 Zmax をとる場合を考える．これ

は，メールａとメールｂが同一であるときである．圧縮率 Zmax は，Zaa に等しい．

(2) Zab が最小値 Zmin をとる場合を考える．これは，メールａとメールｂが全く一致しないときである．このときの圧縮率は，メールａとメールｂを個別に圧縮したときの圧縮率に近似できる．圧縮率 Zmin はつぎのようになる．

Zmin
= 1-(メールａ圧縮後サイズ+メールｂ圧縮後サイズ)
/(メールa圧縮前サイズ+メールb圧縮前サイズ)

(3) 実際の圧縮率 Zab が，Zmin と Zmax の間のどのあたりにあるか，比 Rab を求める．

Rab = (Zab-Zmin) / (Zmax-Zmin)

メールａをメール 001 に固定，メールｂをメール 001～メール 500 まで変化させ，横軸にメールｂ，縦軸に指標 Rab をプロットしたものが図 2 である．

その結果，0.8 以上の集団と 0.35 以下の集団に分かれた．この 0.8 以上の集団について，結合前のメールふたつを見比べてみると，互いに類似していることが確認できた．表 1 は，Rab=0.82 を示したメール 001（表 1 上）とメール 004（表 1 下）の文面である．太字であらわした部分が類似箇所であり，両者の類似度が高いことが分かる．このように，圧縮率を用いてマスメールが分類できることがわかった．

3.2.2 実運用に向けた改善手法の提案

圧縮率を用いる文書分類法は，言語に依存しない非常に優れた方法である．しかし，実際にマスメールの分類に利用しようとする場合は，次のような問題が存在する．

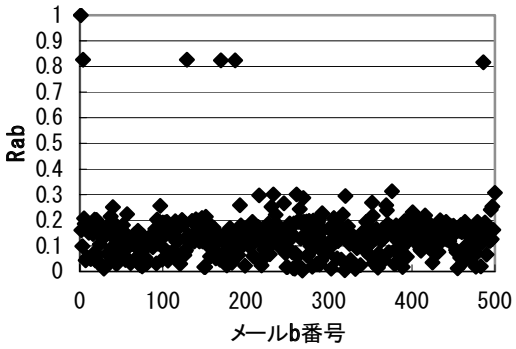

図 2：類似度指標 Rab を用いたマスメール分類

表 1：メール 001（上）とメール 004（下）

vicodin, **phentermine, viagra, soma, ambien, floricet, imitrex, paxil, prozac, zoloft . . . and many many more prescription drugs! no doctor's consultation fees! (saves you up to $100) order today by 2pm est, have it at your door tomorrow! want your prescription drugs as soon as tomorrow? then click here! <a title=http://**buttock**@www.pharmacyzone.biz/a.html href="">no thanks, please take me off your list** brmifca kusweidsvieg wld ukrx wx u a
phentermine, viagra, soma, ambien, floricet, imitrex, paxil, prozac, zoloft . . . and many many more prescription drugs! no doctor' s consultation fees! (saves you up to $100) order today by 2pm est, have it at your door tomorrow! want your prescription drugs as soon as tomorrow? then click here! <a title=http://hokan@www.pharmacyzone. biz/a.html href="">no thanks, please take me off your list w zyhrugjt u wmbrra fwzmc ljl kp mhi f dmxlzmelcj

(1) 新規のメールを分類するときには，既知のマスメール全てと圧縮率を計算しなければならない．これには膨大な計算時間を必要とする．

(2) (1)の困難を避けるために，既知のマスメールをクラスタリングしておいて圧縮率の計算量を少なくする方法が考えられる．しかし，既知のマスメールをクラスタリングできるのであれば，最初からそのクラスタリングアルゴリズムを用いて新規メールをも分類すれば良い．

このように,文書分類に圧縮率を利用する手法は,そのままでは実際の運用に用いることができず，何らかの工夫が必要である.

我々の研究グループは,「結合文書の圧縮率が高いとは元の文書の間に共起文字列が多いということである」という考えに基づき，LZ77 法によって抽出した共起文字列を用いて任意の文書を分類する手法を提案する．具体的には次のような手法である.

(1) あらかじめカテゴリ分けされた文書群（例えば「マスメール」「非マスメール」）から，LZ77 法を用いて共起文字列をとりだす．これを「**共起文字列辞書**」と呼ぶ．共起文字列辞書は分類するカテゴリの数だけつくられる.

(2) 判別したい新規メールの文章に対して，それぞれの辞書中の共起文字列がどの程度マッチするかを調べる．これは，Aho-Corasick 法などのマッチング手法を用いることで，高速に処理できる.

(3) マッチした場合，その共起文字列辞書のカテゴリに対するスコアあるいは確率を計算する.

(4) 最もスコアあるいは確率の高いカテゴリが，判別したい新規メールの所属するカテゴリである.

このアイデアが有効であるかどうかを確かめるための検証実験を行った．以下，共起文字列辞書の作成，性能評価の結果，他の方法との比較について述べる.

3.2.3 実験

実験に用いたメールは，英語・日本語の二種類，マスメール・非マスメールの二種類，計四種類のメールをそれぞれ 1000 通用意した．そのうちの 900 通を共起文字列辞書の作成に用いた（学習メール）．残りの 100 通を性能評価に用いた（テストメール）．以下では，英語については「**英**」，日本語については「**日**」という文字を辞書名につけて区別する.

3.2.4 LZ77 法による共起文字列辞書の作成

共起文字列辞書を作成する方法として，次の(a)(b)のふたつの方法が考えられる.

(a) 全てのメールを結合させたのちに共起文字列辞書を作成する．より多くの共起文字列を抽出できるために性能が良くなるというメリットがあるが，一方で，共起文字列辞書を作り直したり学習データを追加したりするときには，膨大な時間を要する.

(b) 個々のメールから共起文字列を抽出したのちに，それをまとめて共起文字列辞書とする．この方法では，１通単位でしか共起文字列を抽出できないために性能が(a)ほど良くならないが，一方で，辞書を作り直したり学習データを追加したりするときには 1 通分の処理時間で済む.

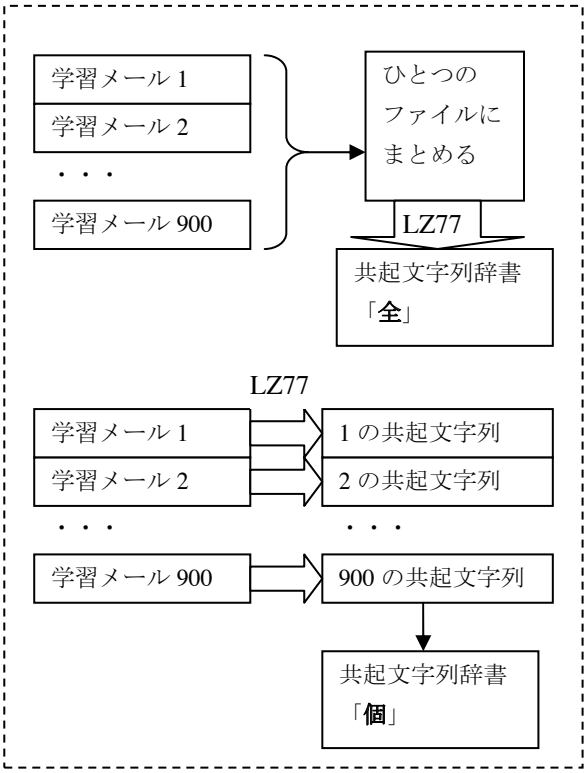

図 2.「**全**」と「**個**」の辞書作成手順の模式図

本研究では，両方の方法で共起文字列辞書を作成し，性能を評価した．以下では，(a)の方法では全てのメールを合わせて共起文字列を取り出すことから「**全**」，(b)の方法では個々のメールから共起文字列を取り出すことから「**個**」を，辞書名につけて区別する.ここまでの辞書作成の手順を模式的に表すと，図 2 のようになる.

3.2.5 有効文字列ごとの共起文字列辞書

共起文字列辞書には，断片的な文字列が多く含まれる.短い文字列はノイズとして働く可能性が高く，また辞書を不必要に大きくして処理速度を低下させる．共起文字列として何文字以上のものを残すのが

妥当であるのかを検討するため，共起文字列辞書から，2，4，6，8，10，12，14，16，18，20 文字以上の文字列のみを残した共起文字列辞書をそれぞれ作成した．以下では，n 文字以上の文字列を残したかの「n」を辞書名につけて区別する．

3.2.6 確率辞書の作成

以上の共起文字列辞書を用いて性能評価実験を行っても良いが，その場合はマスメール用共起文字列辞書と非マスメール共起文字列辞書とで２回のマッチングを行うことになる．今回の実験では，あらかじめ両者の共起文字列辞書を融合させ，各共起文字列ごとに「マスメールに用いられる確率」を計算することで，共起文字列辞書をひとつにした．共起文字列のマスメール確率を計算した統一辞書を「**確率辞書**」と名づける．

確率辞書の作成は，Bayesian-Filter が学習過程においてトークンごとにマスメールに現れる確率を求める計算をしていることに相当する．この実験では，確率辞書の作成に，Bayesian Filter に用いられている Gary-Robinson の式[21]を用いた．

このようにして，「英語」「日本語」の２種類の言語に対して「**全**」「**個**」の２パターンがあり，さらにそれぞれに有効文字列が2～20 文字以上の10パターンある，あわせて 40 個の確率辞書を作成した．

3.2.7 性能評価

このように作成した 40 個の確率辞書を用いて，学習に用いなかった新規のメールに対する検出実験を行った．

新規メールの文章に対して，確率辞書中の共起文字列がマッチングした場合，その共起文字列のマスメール確率を「マスメールスコア」に足していった．そのスコアを正規化し，0.5 以上であればマスメール，0.5 未満であれば非マスメールとみなした．

英語のマスメールと英語の非マスメールに対して「**英**[**全**|**個**]**n**」の確率辞書を用いて検出実験を行った結果を表 3 に，日本語のマスメールと日本語の非マスメールに対して「**日**[**全**|**個**]**n**」の確率辞書を用いて検出実験を行った結果を表 4 に示す．

考察：「英」と「日」の比較

英語でも日本語でもほとんどの辞書が，90%以上の割合でマスメールを正しく検出し，95%以上の割合で非マスメールを正しく検出した．本研究の手法は，英語でも日本語でも，実用的な性能を示しているといえる．

考察：「全」と「個」の比較

英語でも日本語でも，処理速度を犠牲にして性能を重視した「**全**」は，共起文字列を多く抽出しているために項目数が多く，その分検出性能が良い．一方で処理速度を重視して性能を犠牲にした「**個**」は共起文字列の抽出が比較的少なくなるために項目数が少なく，その分検出性能がおちる．ただし，「**個**」の性能が悪いといっても，確率辞書によっては 90%以上の検出率を示している．

考察：有効文字列の比較

有効文字列としては，辞書の項目数と性能の兼ね合いから，6 文字から 10 文字程度が適していると思われる．

表 3. 提案手法の検出結果（英語）

辞書名	項目数	英 SPAM を SPAM と検出	英非 SPAM を非 SPAM と検出
英全 2	327178	**100 %**	1 %
英全 4	309849	**99 %**	**100 %**
英全 6	240154	**100 %**	**100 %**
英全 8	168700	**99 %**	**100 %**
英全 10	113817	**97 %**	**100 %**
英全 12	75208	**97 %**	**100 %**
英全 14	50386	**96 %**	**100 %**
英全 16	**34932**	92 %	**100 %**
英全 18	**25269**	90 %	**100 %**
英全 20	**19037**	88 %	**100 %**
英個 2	100902	75 %	**95 %**
英個 4	91451	77 %	**100 %**
英個 6	67148	94 %	**100 %**
英個 8	**48212**	94 %	**100 %**
英個 10	**34146**	**96 %**	**100 %**
英個 12	**24289**	93 %	**100 %**
英個 14	**17671**	88 %	**100 %**
英個 16	**13212**	85 %	**100 %**
英個 18	**9988**	75 %	**100 %**
英個 20	**7718**	71 %	**100 %**

3.2.8 他の方法との比較

n-gram との比較

提案手法のメリットのひとつは，言語に依存しない同一のアルゴリズムで検出を行うことができることである．ここでは，同じく言語に依存しない*n*-gram での検出実験を行った．

3.3 節と同じ学習データを用いて 2-gram,4-gram, 6-gram, 8-gram の辞書をつくり，3.3 節と同様に確率辞書を作成した．3.3 節と同じテストデータを用いて，3.3 節と同じスコアリング方式で性能評価を行った．

英語に関する結果を表 5 に，日本語に関する結果を表 6 に示す．検出率では，英語では 6-gram，日本語では 4-gram が良い結果を示している．しかし，いずれも辞書の項目数が多いという欠点がある．

Bayesian Filter との比較

マスメールフィルタとして良く用いられている Bayesian Filter との比較を行った．Bayesian Filter は文書をトークン分割する必要があるため，日本語のような分かち書きされていない文書に対しては形態素解析を行う必要がある．この実験では，形態素解析ソフトとして kakasi[22]を用いた．

表 4. 提案手法の検出結果（日本語）

辞書名	項目数	日 SPAM を SPAM と検出	日非 SPAM を非 SPAM と検出
日全 2	179697	**97 %**	**95 %**
日全 4	148561	**97 %**	**100 %**
日全 6	97797	**98 %**	**100 %**
日全 8	62280	**100 %**	**100 %**
日全 10	**40450**	**100 %**	**100 %**
日全 12	**27573**	**100 %**	**100 %**
日全 14	**20294**	**100 %**	**100 %**
日全 16	**16207**	**100 %**	**100 %**
日全 18	**13703**	**96 %**	**100 %**
日全 20	**12011**	92 %	**100 %**
日個 2	57909	**96 %**	**97 %**
日個 4	**43715**	**96 %**	**100 %**
日個 6	**28209**	**99 %**	**100 %**
日個 8	**18876**	**100 %**	**100 %**
日個 10	**12889**	**99 %**	**99 %**
日個 12	**9187**	**98 %**	**100 %**
日個 14	**6986**	**99 %**	**100 %**
日個 16	**5549**	90 %	**100 %**
日個 18	**4544**	82 %	**100 %**
日個 20	**3774**	72 %	**100 %**

3.3 節と同じ学習データを用いて有効文字列 2,4, 6, 8 文字以上の辞書をつくり，3.3 節と同様に確率辞書を作成した．3.3 節と同じテストデータを用いて，3.3 節と同じスコアリング方式で性能評価を行った．

英語に関する結果を表 7 に，日本語に関する結果を表 8 に示す.英語では有効文字列 4～8 文字以上の辞書での検出率が良い．日本語では，マスメール，非マスメールともに誤検出が大きい．Bayesian Filter は分かち書きされていない日本語のような言語に弱いといわれているが，そのことが表れた結果となった．

提案手法と *n*-gram と Bayesian Filter との比較

提案手法での結果と，n-gram の結果と，Bayesian Filter の結果とを比較する．比較しやすいようにそれぞれの手法での良い結果をピックアップし，英語について表 9 に，日本語について表 10 にまとめる．

英語では,Bayesian Filter が最も検出率がよい．

表 5. *n*-gram を用いた検出結果（英語）

n-gram	項目数	英 SPAM を SPAM と検出	英非 SPAM を非 SPAM と検出
英 2-g	**6056**	1 %	**100 %**
英 4-g	217582	81 %	**100 %**
英 6-g	1052966	93 %	**100 %**
英 8-g	1991884	87 %	**100 %**

表 6. n-gram を用いた検出結果（日本語）

n-gram	項目数	日 SPAM を	日非 SPAM を

		SPAM と検出	非SPAM と検出
日 2-g	**11721**	**96 %**	**100 %**
日 4-g	367737	**98 %**	**100 %**
日 6-g	868193	94 %	**100 %**
日 8-g	1217012	83 %	**100 %**

表 7. Bayesian Filter の検出結果（英語）

辞書	項目数	英 SPAM を SPAM と検出	英非 SPAM を 非 SPAM と検出
英 BS2	59658	90 %	**100 %**
英 BS4	55123	**98 %**	**190 %**
英 BS6	**42655**	**99 %**	**99 %**
英 BS8	**27618**	**98 %**	**99 %**

表 8. Bayesian Filter の検出結果（日本語）

辞書	項目数	日 SPAM を SPAM と検出	日非 SPAM を 非 SPAM と検出
日 BS2	62404	**97 %**	3 %
日 BS4	58375	**97 %**	92 %
日 BS6	**42602**	85 %	**96 %**
日 BS8	**31879**	80 %	**96 %**

表 9. 英語の検出結果の比較

辞書名	項目数	英 SPAM を SPAM と検出	英非 SPAM を 非 SPAM と検出
英全 8	168700	**99 %**	**100 %**
英全 10	113817	**97 %**	**100 %**
英個 8	**48212**	94 %	**100 %**
英個 10	**34146**	**96 %**	**100 %**
英 6-g	1052966	93 %	**100 %**
英 BS6	**42655**	**99 %**	**99 %**

表 10. 日本語の検出結果の比較

辞書名	項目数	日 SPAM を SPAM と検出	日非 SPAM を 非 SPAM と検出
日全 8	62280	**100 %**	**100 %**
日全 10	**40450**	**100 %**	**100 %**
日個 8	**18876**	**100 %**	**100 %**
日個 10	**12889**	**99 %**	**99 %**
日 4-g	367737	**98 %**	**100 %**
日 BS4	58375	**97 %**	92 %

提案手法の「**全**」辞書は Bayesian Filter と検出率では遜色ないが辞書の項目数が多い．提案手法の「**個**」辞書は，項目数は妥当であるが，Bayesian Filter に比べてわずかに性能が悪い．*n*-gram は性能が悪いうえに辞書の項目数が多い．まとめると，英語に関しては Bayesian Filter が優位であるが，提案手法が大きく劣っているというわけではない．

日本語では，提案手法は「**全**」「**個**」ともに検出率が良い．*n*-gram は日本語では性能が良いものの，英語と同様に辞書の項目数が多い．Bayesian Filter は日本語では性能が悪く，非マスメールの誤検出が大きい．まとめると，日本語に関しては，提案手法は検出率や辞書の項目数といった点で優れている．

3.2.9 識別型フィルタのまとめ

LZ77 法により抽出した共起文字列を用いる文書分類手法をマスメール検出などに応用し，既存の手法と遜色ない,場合によっては良い性能が得られた．

この手法は異なった言語に対しても同一のアルゴリズムで処理できる特徴がある．実際に，言語に依存することなく一定の検出性能を示した.この点は，日本語の処理が苦手な Bayesian Filter よりも有利な点である．

提案手法は，有用であることが示されたものの，マスメール検出システムとして運用するには，まだ次のよう
な課題を克服しなければならない．

・共起文字列辞書の短い文字列を一律に切り捨てているが，切り捨てて良い文字列とそうでない文字列を論理的に判別すべきである．

・マスメールのスコアを求める計算は，今は単純な可算方式である．検出性能を上げるためには，より適切な演算方法を探求する必要がある

・システムとして統合されていない．コマンドひとつで使えるようなツールとしなければならない．

4. NNIPF の開発

これまでに開発を行なってきたマスメールフィルタは，メールの本文を解析するタイプのものであった．これらのフィルタは，

・事例を大量に集める必要がある．

・識別型フィルタでは短いメールの誤識別が避けら

れない.
・トレーニングを怠ると，誤検出が増加する.
などの問題点がある.

典型的な例としては,「お姉ちゃんと一緒に帰る」という子供の送ったショートメールは殆どの識別型フィルタで誤識別を起こす. この理由は,「お姉」あるいは「お姉ちゃん」という文字列のマスメールキーワードとしてのスコアは高く，他の部分では特徴的な文字列が表れないためにマスメールスコアが高くなり，結果的にマスメールと誤識別されてしまうためである.

このような誤識別の例は，メール本文の解析に頼らない識別が必要であることを強く示唆しており，その考えに基づいて Nearest Neighbor IP address based mail Filter(NNIPF)の開発が行なわれた.

4.1 送信元 IP アドレスの割り出し

しかし，マスメールのヘッダー情報は通常，偽造されており，メールヘッダの解析を行なっても，真のメール送信者を割り出すことは容易ではない.

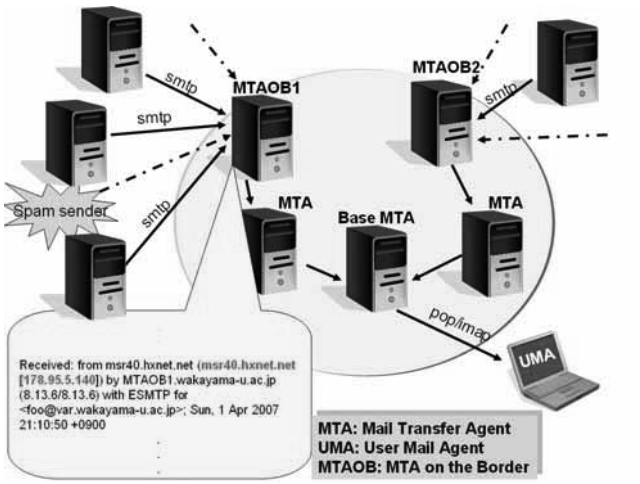

図 3. メールの配送と MTAOB

ところが，メールヘッダのうち Received 行と呼ばれる部分は，正常な MTA では偽造を行なわず機械的に生成されるため，これを利用すると真のメール送信者を割り出すことができる.

これは図 3 に示すように通常利用者が受けるメールを外部から受信している信頼できる MTA を知っていれば,その MTA が生成する Received 行には送信元の正確な情報が記録されているという事実を利用する方法である. このような MTA を MTAOB (MTA On Boundary) と呼ぶことにする. 例えば, 図 3 の場合, MTAOB の一つである MTAOB1 は以下のような Received 行をメールヘッダに追加する.

Received: from msr40.hxnet.net (**msr40.hxnet.net [178.95.5.140]**) by MTAOB1.wakayama-u.ac.jp (8.13.6/8.13.6) with ESMTP for <foo@var.wakayama-u.ac.jp>; Sun, 1 Apr 2007 21:10:50 +0900

この場合太字で書いた部分は, MTAOB1 自身が調べて得た情報であるので，信頼できる情報であると言える. したがって，このメールを MTAOB1 に送ってきた相手は 178.95.5.140 という IP アドレスを持っていることが保証される. このようにすれば，送信元の計算機の素性を調べることができる.

4.1.1 偽造に対する処置

但し, 上述の Received 行が偽造されたものである場合には，正しい IP アドレスは得られない. このような問題を回避するために，Received 行の by 以降に自分の登録した MTAOB が表れた場合，送信元の IP アドレスを調べ,得られた IP アドレスのマスメール送信者らしさを計算し，その最大値を用いてマスメールか否かを判定するという方式を採用する. こうすれば,マスメール送信者が Received 行を偽造することによりマスメールを正常なメールに見せかけることは行なえなくなる.

4.2 IP アドレスからのマスメールらしさの計算

上述のようにして,送信元 IP アドレスが抽出できたとき, どのようにすれば IP アドレスからマスメールらしさが計算できるだろうか？

これまでにも様々な識別器が提案されてきたが,
・識別能力が高いものは学習に時間がかかる
・教示を行なっても同じ誤りを起こす可能性がある
などの問題点がある.
このような問題点を持たない識別器の例として，最近傍識別器がある. これは, 例えばクラス Ω_1, Ω_2 に属するデータ集合 $I^1=\{x_1^1, x_2^1, \cdots, x_{n1}^1\}$ と $I^2=\{x_1^2, x_2^2, \cdots, x_{n2}^2\}$ が与えられ，各データ間の距離を $d(x_j^i, x_l^k)$ と表すとき，ある入力データ x とデ

ータ集合 I^i との距離を以下のように定めるものとする.

$$d(x, I^i) = \min_j (d(x, x_j^i))$$

このとき,

$$\begin{cases} d(x,I^1) > d(x,I^2) & \Rightarrow \quad x \in \Omega_1 \\ d(x,I^1) = d(x,I^2) & \Rightarrow \quad unknown \\ d(x,I^1) < d(x,I^2) & \Rightarrow \quad x \in \Omega_2 \end{cases}$$

という規準で入力データ x の識別を行うのが，最近傍識別である.

しかし，この規準だけではどの程度入力データが識別先クラスに近いのかが計算できない. このため，我々は次に示す弁別度と呼ばれる量を提案している.

$$P(\Omega_1 \mid x) = \frac{d(x,I^2)}{d(x,I^1) + d(x,I^2)}$$

これは，入力データ x が与えられたとき，それが Ω_1 に属する確率 $P(\Omega_1 \mid x)$ を $p(x \mid \Omega_i) = \dfrac{1}{d(x,I^i)}$ という確率密度関数を仮定して計算していることに等しい.

Ω_1 をマスメール，Ω_2 を非マスメールと対応付け，入力データ x を IP アドレスと見なして，上の弁別度を計算した場合，

・x がマスメール送信者の IP アドレスと一致した場合，$P(\Omega_1 \mid x)$=1.

・x が非マスメール送信者の IP アドレスと一致した場合，$P(\Omega_1 \mid x)$=0.

・$d(x,I^1) = d(x,I^2)$ の場合，$P(\Omega_1 \mid x)$=0.5

という性質を持つ．この性質から，

$$\begin{cases} P(\Omega_1 \mid x) > 0.5 & \Rightarrow \quad x \in \Omega_1 \\ P(\Omega_1 \mid x) = 0.5 & \Rightarrow \quad unknown \\ P(\Omega_1 \mid x) < 0.5 & \Rightarrow \quad x \in \Omega_2 \end{cases}$$

と判定しても，最近傍識別と同様の判定が行なえる. このような確率を計算することのメリットは，明確ではない場合の判定を留保して人間の判断に委ね，それ以外の明確な場合について，自動的に識別を行うことが実現できることである．すなわち，例えば

$$\begin{cases} P(\Omega_1 \mid x) > 0.65 & \Rightarrow \quad x \in \Omega_1 \\ P(\Omega_1 \mid x) < 0.35 & \Rightarrow \quad x \in \Omega_2 \\ otherwise & \Rightarrow \quad unknown \end{cases}$$

のように，明白ではないケースは unknown として人間の判断を仰ぐというやり方である．このような方式は NNIPF の判定でも採用している.

4.3 NNIPF の実装

これまで述べてきた IP アドレスの抽出法と，マスメールらしさの度合いを表す弁別度計算を組み合わせることで，マスメール検出器が容易に構築できる. このようにして出来上がるマスメールフィルタは，マスメールと正常なメールの各送信者の IP アドレスデータベースを用いて最近傍探索を行わなければならない．この最近傍探索を効率的に行うために，計算機のディレクトリの階層構造を用いた高速な探索法を新たに開発した．これらの機能は全てライブラリ関数として定義されており，このライブラリを用いて容易にアプリケーションプログラムが作成できるようになっている.

今回の公開のために作成したプログラムは，メールフィルタと，Web インタフェースである. 後者は，メールの内容の確認，MTAOB の登録，誤識別の修正，再学習，送信元アドレスの表示などの機能を持っており，利用者の利便性を向上させるためには必須のものである．図 4 に検出したメイルの一覧表示，図

図 4.　Web インタフェースの画面

図 5. NNIPF によるヘッダ部の解析結果（大きな文字で書かれている部分が送信者の IP アドレス.）

図 6. IP アドレスの解析結果（この例では，100%の弁別度が得られ，そのアドレスがメキシコのものであることを示している.）

5 に NNIPF が解析したメールヘッダ部の表示結果の例，図 6 に IP アドレスの解析結果の例を示す．これらの結果は全て Web インタフェースから表示できるようになっている．

ライブラリ，メールフィルタ，Web インタフェースは全て Perl で書かれており，sendmail，qmail，Postfix など良く用いられている殆どの MTA 上のフィルタとして動作することが確認されている．

4.4 評価

NNIPF を実際に運用した際の識別性能は，その利用者がどの程度トレーニングを行なうかに依存している．1 日に 100 通以上のマスメールを受信している複数の利用者が，毎日 1 回のチェックを行なった結果，98%～100%の識別性能が得られることが確認されている．また，一日のマスメール受信数が 10 通程度の利用者の場合，実運用において，ほぼ 100%の識別性能が得られている．

4.5 公開後の反応

NNIPF を公開した後に，公開ホームページには数万回のアクセスがあり，公開後 10 日程度で 1000 回を超えるダウンロードが行なわれた．この背景には，数多くの BLOG でこのプログラムの存在が話題になったことや，「窓の杜」と呼ばれる無料ソフトウエア

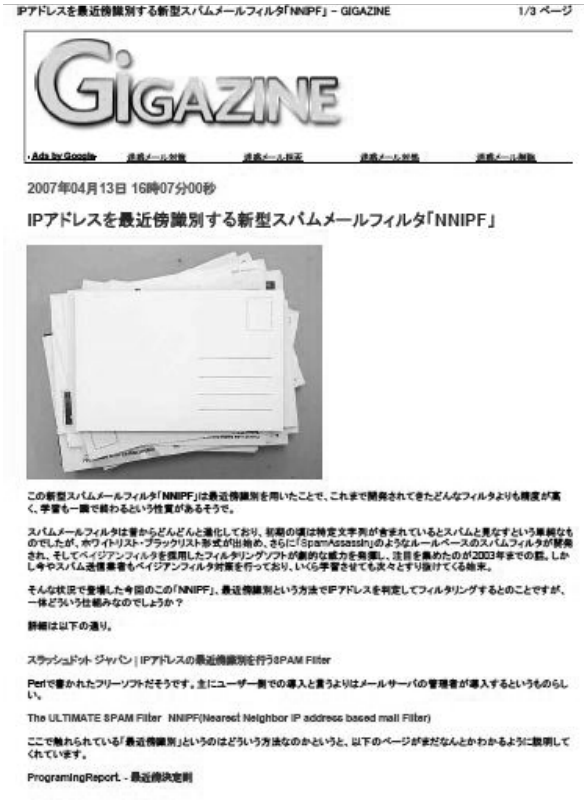

図 7. NNIPF が掲載された WEB ページの例

のホームページに掲載されたことなどが背景にある．事実，公開後約 10 日で Google による検索で，8 万件以上のホームページに掲載されたことが確認されている．

現在では約 700 程度のホームページに掲載されているに留まっているが，今後海外向けに公開を行なう準備を進めており，その際にはより多くのアクセスがあるものと期待されている．

利用者からは，「SpamAssassin[7]よりもかなり性能が高く驚いている」,「spam メールが全く届かなくなった」などの好意的な反応が多く寄せられた．その反面,「うちの組織では MTA 上で WEB サーバを動かすことが禁止されているので，利用できない.」「もう少し利用しやすくできないか」などの問題点の指摘や要望も少数ではあるが寄せられている．

５．まとめ

これまで，５年以上もマスメール検出法およびその実システムの開発をすすめてきたが，前述の通り当大学の情報学センターのマスメール対策措置の影響を受け，プロジェクトの遂行に大きな障害が発生した．しかし，情報学センターがこのような措置を講じることは極めて自然であり，むしろ我々がこういった反社会的行為に対する実効的措置を講じる方法を研究していることの方が不自然であったのかもしれない．それにもかかわらず，実験用ホストに対してだけマスメールのブロックを解除してもらえたことは，研究を継続する上で非常に大きな助けとなった．

また，これまでの事例ベース検出法を維持することができなくなり，NNIPF の開発を行なう契機になったのも，情報学センターのこういった活動の結果である．これにより，実運用において従来 60%～85%程度のマスメール検出率が，98%以上に一挙に跳ね上がり，日本国内での NNIPF の普及に弾みがついたことは，結果的にプロジェクトを成功に導くきっかけになったとも言えよう．

この場を借りて，情報学センターの皆様に感謝の意を表したい．

今後は，国内だけでなく海外への普及を進めるために，より洗練されたシステムにしていく予定である．特に

- MTAOB をメールの仕組みが良く分からない人に設定させることは困難であるため，受信したメールを元にして自動的に設定する機能を持たせること．
- マスメール送信元 IP アドレス，非マスメール送信元 IP アドレスのデータを複数の利用者が協調して収集，管理する仕組みの考案．
- Windows などのメールクライアントでも利用できるようにするため，pop3 proxy サーバへの組み込みを行なうこと．

などが重要な課題として残されている．

これらの技術的課題以外にも，NNIPF を国内外のサービスプロバイダなどに使用してもらうためのコンソーシアム作りも必要であり，対外的な広報活動も今後進めていく予定である．

研究を開始した 5 年前とは状況が異なり，現在のマスメールの大半（90%程度）はウイルスや Worm に感染した PC が不正に利用され,そこから送られてきている.このような不正利用された PC によって形成されるネットワークは BOTNET と呼ばれる．この BOTNET は数千台から数十万台の PC を含み，その種類も数百を超えると言われており,マスメール送信,DOS 攻撃などの反社会的行為に利用されている．BOTNET に組み込まれた PC ユーザの大半は自分の PC がこのような反社会的行為に利用されていることを知らずに居るが，これを放置しておけば自分がその加害者になり，処罰の対象になる．

NNIPF が収集している IP アドレスの大半はこの BOTNET に組み込まれた PC のアドレスであり，それを各ユーザに伝える手段があれば，BOTNET から PC を助け出すことにも応用できるはずである．実際，我々は国内のプロバイダのクライアントから送られてくるマスメールを分析し，プロバイダ経由で利用者に注意を喚起する活動も行なっている.この結果，NNIPF によって OCN 内の数十のクライアントを BOTNET から離脱させることに成功している．

これまではマスメールそのものをいかにして検出するかということを中心に研究を行ってきたが，今後は,マスメール送信の元である BOTNET に組み込まれた PC の発見についても研究を行っていきたい.幸いマスメール送信元の IP アドレスが特定できるという NNIPF の特性はこのような用途にも適しているため，比較的短期間でこのようなシステムは構築できるはずである．

参考文献

[1] ORDB.org: Open Relay Database, http://www.ordb.org/

[2] MAPS Inc.: Mail Abuse Prevention System, http://www.mail-abuse.com/

[3] SpamCop.net Inc.: SpamCop Black List, http://www.spamcop.net/bl.shtml

[4] RFC2821, SMTP: Simple Mail Transfer Protocol, http://rfc.net/rfc2821.html

[5] greylisting, http://greylisting.org/

[6] お馴染みさん方式, http://moin.qmail.jp/_a4_aa_c6_eb_c0_f7_a4_b5_a4_f3_ca_fd_bc_b0

[7] SpamAssassin development team: Spam Assassin, http://spamassassin.org/

[8] Graham P.: A Plan for Spam, http://www.paulgraham.com/spam.html, Hackers & Painters (2002)

[9] Vipul Ved Prakash, Vipul's Razor, http://razor.sourceforge.net/

[10] Frank J. Tobin, Pyzor, http://pyzor.sourceforge.net/

[11] 和田俊和, 斉藤彰一, 泉裕, 上原哲太郎, "コンテンツに基づくマスメールフィルタリング", 情報処理学会研究報告, 2003-QAI-8, pp.55 (2003).

[12] 松浦広明, 斉藤彰一, 上原哲太郎, 泉裕, 和田俊和, "マスメールデータベースとそれを用いたマスメール検出システム", 情報処理学会研究報告, 2004-DSM-32, pp.73 (2004).

[13] 松浦広明, 斉藤彰一, 上原哲太郎, 和田俊和, "データベースに基づくサーバサイド迷惑メール検出システム", 電子情報通信学会誌 B Vol.J88-B No.10 pp.1934-1943 (2005)

[14] 松浦広明, 和田俊和, "Lempel-Ziv 法によって得られた共起文字列を用いる文書分類：迷惑メールへの適用", NLP 若手の会第一回シンポジウム, 名古屋大学 (2006)

[15] J.Graham-Cumming, "The Spammer's Compendium," 2003 SPAM CONFERENCE, (2003).

[16] D.Benedetto, et. al.: "Language Trees and Zipping, " Phys. Rev. Lett., Vol.88, pp.1-4 (2002)

[17] A.Lempel and J.Ziv, IEEE Trans.Inf.Th., pp.337-343 (1977).

[18] 安形輝, "異なるデータに対応したデータ圧縮による類似データ同定", 情報処理学会研究報告, 2005-FI-79, pp.9-16 (2005)

[19] 松崎大輔ら, "圧縮性に注目した文書の関係分析手法", 情報処理学会研究報告, 2006-FI-84, pp.51-56 (2006)

[20] 木村洋章ら, " LZ78 の圧縮性を利用した文書検索手法の提案", 情報処理学会研究報告, 2006-FI-84, pp.65-70 (2006)

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

[22] H.Takahashi, "KAKASI- Kanji Kana Simple Inverter," http://kakasi.namazu.org/