社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2004－ＱＡＩ－11　（1）
2004／5／26

文章密度に基づくマスメールの高速検出手法と評価

藤川　裕充[†]　吉田　健一[††]　足立　史宜[†††]　鷲尾　隆[†††]　元田　浩[†††]
本間　輝彰[†]　中島　昭浩[†]　山崎　克之[††††]

† KDDI 株式会社
†† 筑波大学
††† 大阪大学 産業科学研究所
†††† 株式会社 KDDI 研究所

あらまし　昨今，インターネットの普及と共に増大する spam メールの問題は深刻化し，技術的な面のみならず，社会的にも問題となっている．本研究の手法では，spam メールを含むマスメールを，メールの内容を特徴ベクトル化し，その文章密度を利用することによって，高速に検出することが可能である．本研究の手法でマスメールの検出を行うためには多量のメールが必要となるが，教師なし学習の枠組の中で，Recall 100%，Precision 100%の精度でマスメールを検出することが可能である．また，Direct-mapped cache の機構を用いることにより，Pentium4 2GHz の Linux 上で 12,500 通/秒の処理速度でメールを処理することが可能である．
本論文では，約 5,000 万通の実メールトラフィックを本研究の手法で解析した結果も報告する．
キーワード　spam, マスメール, 高速, 検出

Density-based high speed mass-mail detection technique and its evaluation

Hiromitsu FUJIKAWA[†], Kenichi YOSHIDA[††], Fuminori ADACHI[†††], Takashi WASHIO[†††], Hiroshi MOTODA[†††], Teruaki HOMMA[†], Akihiro NAKASHIMA[†], and Katsuyuki YAMAZAKI[††††]

† KDDI CORPORATION
†† University of Tsukuba
††† I.S.I.R., Osaka University
†††† KDDI R&D Laboratories Inc.

Abstract　The volume of mass unsolicited electronic mail, often known as spam, has recently increased enormously and has become a serious threat to not only the Internet but also to society. This paper proposes a new spam detection method which uses document space density information. Although it requires extensive e-mail traffic to acquire the necessary information, an unsupervised learning engine with a short white list can achieve a 100% recall rate and 100% precision. A direct-mapped cache method contributes handling of over 12,500 e-mails per second. Experimental results, which were conducted using over 50 million actual e-mails, are also reported in this paper.
Key words　spam,mass-mail,detection

1. はじめに

昨今，インターネットの普及と共に増大するspamメールの問題は深刻化し，技術的な面のみならず，社会的にも問題となっている．また，メールを利用出来る携帯電話の普及に伴い，携帯電話に対するspamメールも問題となっている．携帯電話でのメールの受信は，PCでのメールの受信とは異なり，ユーザが端末側でspamメールの対策を行うことがほとんど出来ない．このため，携帯電話に届くspamメールの対策は，携帯電話宛のメールを受信するメールサーバで行う必要がある．携帯電話に特化せず，一般的に大規模なメールサーバでのspamメール対策用システムには，次の仕様が要求される．

- 高速な処理速度
  大規模なメールサーバでは1日あたり1億通以上のメールが扱われる．このメールをリアルタイムに処理するためには，1,000通/秒程度の処理速度が要求される．また，メールサーバ上に実装する場合，メール処理への影響を極力少なくする必要がある．

- 管理が容易である
  一般的なspamメールの対策システムとして，SpamAssasin [1]，bayesian filtering [6] を用いたbsfilter [2] などがあるが，これらは新しいタイプのspamメールが届く度にデータベースを更新する必要がある．この作業はメールの内容を読んでspamメールであるかどうかを判断する必要があり，多量のメールが届くメールサーバでの運用は極めて困難である．また，学習などを利用すると処理の負荷がかかることもあるため，学習を行わない，もしくは処理負荷の軽い学習を利用したアルゴリズムが必要である．

- 高精度
  spamメールの検出精度が高いことは必須であるが，これに加えて，通常のメールをspamメールであると判断する誤検出が無いことが要求される．

- プライバシーへの配慮
  spamメールであってもその内容を第三者が見ることは許されないため，メールサーバの運用者がメールの内容を直接見ずにspamメール対策を行える必要がある．

本研究では，spamメール対策の一環として，同一または類似した内容のメールが多量に届くマスメールの検出を行うシステムを作成し評価を行った．マスメールには多量に届くspamメールに加えて，メールマガジンなども含まれるがホワイトリストなどを用意することで対策が可能である．ISPなどのメールサーバでは，ユーザ宛のspamメールを多量に受信するため，マスメールの検出はspamメール対策としても有効である．

2. マスメール検出システム

2.1 マスメールの文章密度 [7]

spamメールの対策として [1], [2], [3], [4] など，bayesian filtering用いたものや，ルールベースのフィルタが存在する．これらのシステムでは，メールの内容を特徴ベクトル化したものを利用して学習を行っている．メールの内容を特徴ベクトル化し，その特徴ベクトルの分布をspamメールと通常のメールで比較すると，図1の例の様に，昨今の巧妙なspamメールは通常のメールと判別が難しい．また，携帯電話に届くspamメールは，携帯電話の液晶1画面分に収まるように作成される傾向があり，本文が極めて短いため，特徴ベクトル空間利用したspamメールの検出は極めて困難である．本研究の手法では図2に示す通り，文章密度を用いることにより，各メールの特徴ベクトル空間中での類似したメールの検出が可能となる．

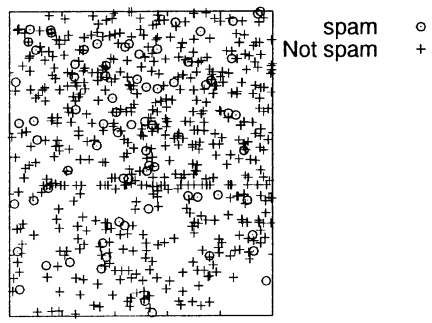

図1 メールの特徴ベクトル空間

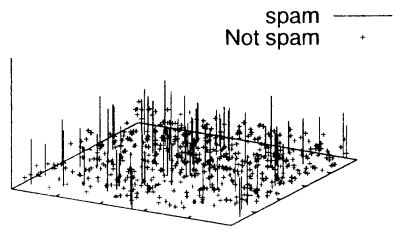

図2 類似したメールの分布

2.2 システム構成

本研究で作成したマスメール検出システムは図3の構成を持っている．システムは設置が容易に行えるように，メールサーバの前のスイッチングハブなどでパケットをミラーリングするポートを作成し，ミラーリングされたパケットを利用する構成を採っている．

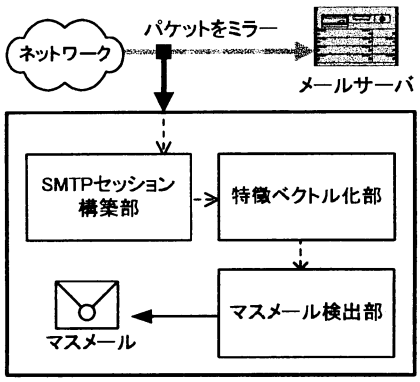

図3 マスメール検出システム構成

SMTPセッション構築部では，ミラーリングされたパケットを利用して，パケットの中からSMTPセッションを抜き出し，セッションの内容を解析してメールの内容を構築する．SMTPセッション構築部と，その他の部分は分離しており，メールサーバに直接組み込む場合や，ディスクなどに保存されたメールのデータを解析する場合は，SMTPセッション構築部を利用せず，直接メールのデータを特徴ベクトル化部に渡すことも可能である．

SMTPセッション構築部で構築されたメールの内容は，特徴ベクトル化部で，不可逆な特徴ベクトルの集合に変換される．作成された特徴ベクトル集合は，図5のDirect-Mapped CaheとHash Data Baseからなるマスメール検出部で処理される．前者は高速検索のためのデータ構造で，後者は特徴ベクトル集合すなわちメールの情報を格納する部分である．

2.3 キャッシュ構造

SMTPセッション構築部で作成されたメールの内容は，特徴ベクトル化部で特徴ベクトルの集合に変換される．メールの内容から特徴ベクトルへの変換には図4の通り，先頭からL文字を順次切り出しハッシュ値[9]を計算することで行う．メールの内容はそのまま，もしくは何らかの前処理を行ったあと，先頭から順番にL文字ずつ切り出しハッシュ値を計算する．ハッシュの計算は，切り出す先頭をM文字ずつずらして順次行い，メールの内容の終端もしくは一定量(N)の特徴ベクトルが作成されるまで行う．

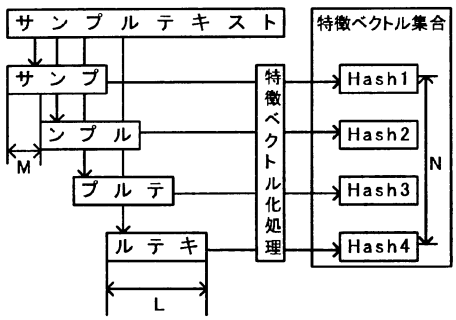

図4 特徴ベクトル化処理

特徴ベクトル化されたデータは，Hash Data Baseと呼ばれるキャッシュに格納される．Hash Data Baseの1つのエントリには，作成された特徴ベクトル集合と，後述のDirect-Mapped Cacheからの被参照数，および類似した回数を記録する類似カウンタが存在する．Direct-Mapped Cacheの1つのエントリには，各特徴ベクトルの値と，その特徴ベクトルが含まれるHash Data Baseのエントリへのポインタが含まれる．

特徴ベクトル集合をDirect-Mapped Cache，Hash Data Baseを利用してマスメールを検出するアルゴリズムは図6の通り．

新たに作成された特徴ベクトル集合は，まず各特徴ベクトルがDirect-Mapped Cacheに存在するか否かを順に確認する．Direct-Mapped Cacheに存在する場合は，そのエントリが指すHash Data Baseのエントリを参照し，参照先のエントリが持つ特徴ベクトル集合との類似度を判定する．類似度は，特徴ベクトル集合を構成する特徴ベクトルのうち同一のものの割合を利用する．類似度がしきい値以上の場合を類似しているとし，類似している場合は，該当のHash Data Baseの類似カウンタに1を加算し終了する．

すべての特徴ベクトルがDirect-Mapped Cacheに存在しないか，類似したHash Data Baseのエントリが存在しなかった場合は，新規にHash Data Baseにエントリを作成する．そして，特徴ベクトル集合の特徴ベクトルを指定した数だけDirect-Mapped Cacheへ登録する．既にDirect-Mapped Cacheに特徴ベクトルが存在している場合は，そのエントリが指すHash Data Baseのエントリへのポインタを新規に作成したHash Data Baseのエントリへのポインタに変更する．特徴ベクトルをDirect-Mapped Cacheに登録した際に，Hash Data Baseの被参照数を1加算する．また，古いHash Data Baseのエントリの Direct-Mapped Cacheからの被参照数を1減算する．Direct-Mapped Cacheからの被参照数が0になった場合は，そのHash Data Baseのエントリは不要

となるため，Hash Data Base からエントリを削除する．

Hash Data Base の類似カウンタがしきい値以上になった場合，その特徴ベクトル集合を持つメールはマスメールとして検出する．

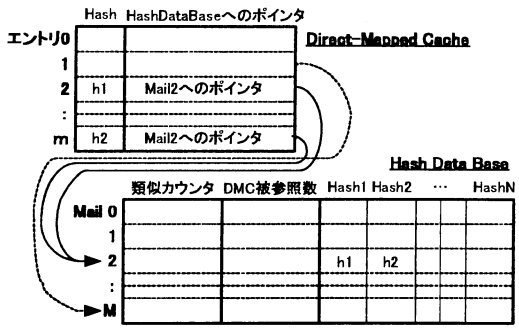

図 5　キャッシュ構造

3. 実メールトラフィックの解析

マスメール検出システムの性能と動作検証のため，ISP に流れるメールトラフィックを 1 週間収集し，そのデータを解析した．

3.1 システム構成

ISP のメールサーバ群が存在するセグメントのスイッチングハブで，メールサーバのつながっているポートの一部をミラーリングした．ミラーリングしたポートに，データ収集用の PC を接続し，SMTP セッション構築部のみを利用して，ミラーされたパケットからメールの内容を構築しディスクに保存した．接続構成図を図 7 に示す．(図では 1 台のメールサーバのミラーリングとなっているが，実際は複数台のメールサーバのトラフィックをミラーしている．)

データの収集は 1 週間連続で実施し，約 5,000 万通のメールが収集出来た．

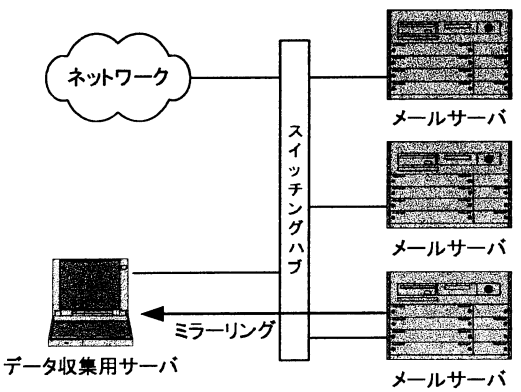

図 7　データ収集システム構成

```
Main Procedure Check-Mail
Input
    T: Text of Mail
Var h: Hash value
begin
    New-Hash-DB-Candidate
        ← Make N Hash values from T
    for h in New-Hash-DB-Candidate do
        if Similar(Mail in Hash-DB pointed by h,
                New-Hash-DB-Candidate)
        then Update-Similar-Mail(
                Mail in Hash-DB pointed by h)
            exit Check-Mail
    // If No Similar Entry exists in Hash DB
    Store-New-Mail(New-Hash-DB-Candidate)
end

Function Similar
Input
    H1: Hash-DB entry
    H2: New-Hash-DB-Candidate
begin
    if H1 and H2 share S same hash value
    then return Yes
    else return No
end

Procedure Update-Similar-Mail
Input
    H1: Hash-DB entry
Var h: Hash value
begin
    Increment "No.of Similar Mail" of H1
    for first n h in H1 do
        Set-DMC-Entry(current h, H1)
    if No.of Similar Mail > D
    then Mark H1 as "spam"
end

Procedure Store-New-Mail
Input
    H2: New-Hash-DB-Candidate
Var h: Hash value
begin
    Store H2 as New Hash DB Entry
    for first n h in H2 do
        Set-DMC-Entry(current h, H2)
    Set "No.of Similar Mails" as 1
    Set "No.of DMC Entry" as n
end

Procedure Set-DMC-Entry
Input
    h: Hash value
    H: Hash-DB Entry
begin
    Set DMC Entry for h so that it points H
    if Previous h already point Hash-DB Entry: e
        & e ≠ H
    then Decrement "No.of DMC Entry" of e
    if "No.of DMC Entry" of e = 0
    then Delete e from Hash DB
        & Clear DMC entry which point e
end
```

図 6　キャッシュアルゴリズム

3.2 結　　果

収集したデータの解析を，Pentium4 2.4GHz，メモリ2G バイトの機器で Linux を用い，表 1 のパラメータで解析を行った．

マスメールの定義として，特徴ベクトルの類似度が90%以上のメールが 100 通以上存在するものとした．また，システムに関するパラメータとして Direct-Mapped Cache のエントリ数を 200 万件，Hash Data Base のエントリ数を 100 万件，特徴ベクトルを作成する際の文字長を 9 文字，ずらす文字数を 3 文字，特徴ベクトルを作成する上限を 100 個，Direct-Mapped Cache エントリから Hash Data Base へ保存する特徴ベクトルの数を 10 個とした．

データの解析結果は表 2 の通りで，マスメールとして検出されたメールは約 12,000 万通，全体の約 22.8%であった．また，検出されたマスメールの種類は約 14,000 種類となった．

データ解析に要した時間は 4,340 秒 (約 72 分) で，処理速度は約 12,500 通/秒である．マスメール検出システムが使用したメモリは約 845M バイトである．これは，主に Direct-Mapped Cache のエントリ数と Hash Data Base のエントリ数で決定され，解析するデータ量での変動は無い．

表 1　データ解析パラメータ

Direct-Mapped Cache エントリ数	2,000,000
Hash Data Base エントリ数	1,000,000
マスメールのしきい値	100 通
L:文字長	9
M:移動数	3
N:特徴ベクトル化数	100
n:Hash Data Base への保存数	10
S:類似と判断する割合	90%

表 2　1 週間のメールトラフィックに関するデータ

メールの総数	約 53,985,000 通
マスメールの総数	約 12,325,000 通
マスメールの占める割合	22.8%
マスメールの種類	約 14,000 種類
処理時間	4340 秒
処理速度	約 12,500 通/秒

図 8 は，検出された 22.8%のマスメールの分布で，横軸が 1 種類あたりの通数，縦軸が種類数となっている．一般的にネットワークのデータが従う [5] zipf の法則にマスメールの分布も従っている．

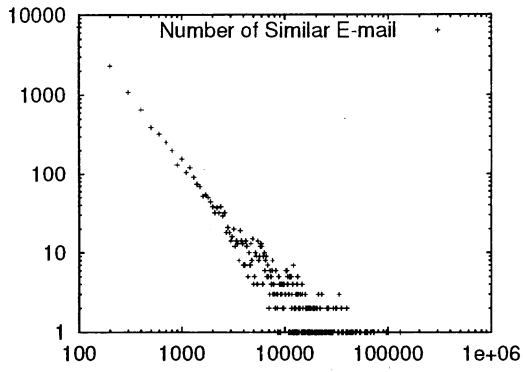

図 8　マスメールの分布

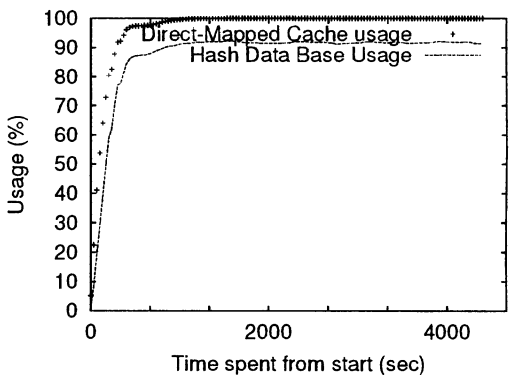

図 9　マスメールの分布

データ解析時の Direct-Mapped Cache と Hash Data Base の推移を図 9 に示す．点線が Hash Data Base の使用率の推移で，+が Direct-Mapped Cache の使用率である．

Direct-Mapped Cache の使用率が 100%となると Hash Data Base の消費が 90%程度の値で推移している．

3.3 性能評価試験

マスメール検出システムの検出性能の評価のため，実メールデータに疑似 spam メールを挿入して次の性能評価試験を行った．

3.3.1 評価試験用データ

評価試験用データとして，収集した実メールデータから先頭 1,000 万通を抜き出して元データとした．

疑似 spam メールは，いくつかの Web ページからランダムに抜き出した文章 (可変長) に，ランダムに収集した URL を挿入して作成している．

元データに，作成した疑似 spam メール 100 種類をそれぞれ 10 通ランダムに挿入したデータ，100 種類を 20 通ランダムに挿入したデータ，30〜100 通ランダムに挿

入したデータと，10種類のデータを作成した．作成したデータには，疑似spamメールに加えて，実メールデータに既に存在しているマスメールも含まれている．

3.3.2 結　　果

作成した10種類のデータを，それぞれマスメール検出システムを利用してマスメールの検出を行った．検出には実メールトラフィックを解析したPCを利用し，表1と同じパラメータを利用した．

図10は10種類のデータの解析結果で，各データのRecallと，検出出来なかった疑似spamメールの割合をグラフにしたものである．○でプロットされたものがRecall値，■でプロットされたものが検出出来なかった疑似spamメールの割合となっている．

結果，40通以上疑似spamメールを挿入したデータに関しては，すべての疑似spamメールが検出され，Recallは100%となっている．

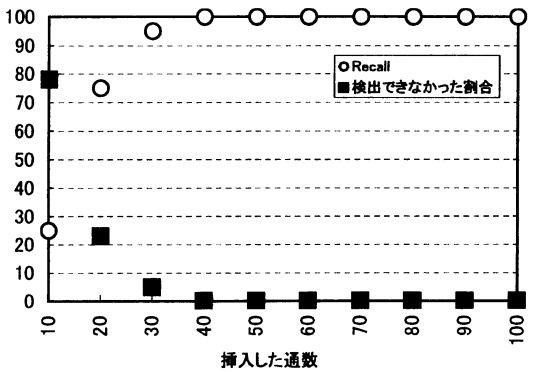

図10　検出性能

4. ま と め

本研究で作成したマスメール検出システムには次の特徴がある．

- 高速な処理速度
  Pentium4 2.4GHzのノートPCで，12,500通/秒(約12億通/日)の処理速度が出ることを確認した．これは，1日1億通程度のメールを扱う大規模なメールサーバでもメールトラフィックをリアルタイムで処理することが可能であると考えられる．

- データベースの管理をする必要がない
  新しい形式のspamメールが発生した際でも，本手法のDirect-Mapped CacheとHash Data Baseを用いたキャッシュ構造を利用していれば，データベースを人手で管理すること無く，新しい形式のspamメールの検出を行うことが可能である．

- 高い検出性能
  1,000万通の実メールデータに疑似spamメールをランダムに挿入して性能評価試験を行った結果，1,000万通に40通以上の割合で挿入された疑似spamメールはRecall 100%，Precision 100%という高い精度で検出することが出来た．

- プライバシーへの配慮
  メール本文を直接利用せず，特徴ベクトル化処理を行ったデータを用いてマスメールの検出を行うため，プライバシーの観点からも良好である．

5. おわりに

本システムは，大規模なメールサーバでマスメールを検出することを目的としている．そのため，本システムではマスメールの検出のために多量のメールが必要となる．

本研究では，SMTPセッション構築部を利用してミラーリングされたトラフィックからメールの内容を構築し解析を行ったが，実際にメールサーバへ導入する際には，特徴ベクトル化部，マスメール検出部のみを実装しマスメールを検出する構成を採ることが出来る．また，マスメールにはspamメールに加えて，メールマガジンも含まれている．そのため，実用化の際にはメールマガジンなど多量に届くがspamメールではないものをホワイトリストなどを用いて除外する必要があり，今後の検討課題である．

文　　献

[1] SpamAssasin *http://spamassassin.taint.org/*, 2004.

[2] bsfilter *http://bsfilter.org/*, 2004.

[3] Distributed checksum clearinghouse *http://www.rhyolite.com/anti-spam/dcc/*, 2004.

[4] 和田 俊和, 齋藤 彰一, 泉 裕, 上原 哲太郎,"コンテンツに基づくマスメイルフィルタリング", IPSJ SIG 研究報告,pp55-60, 2003.

[5] N. Nishikawa, T. Hosokawa, Y. Mori, K. Yoshida, and H. Tsuji, "Memory-based architecture for distributed www caching proxy," Proc. of World Wide Web Conference 98, pp.205-214, 1998.

[6] P. Graham. Better bayesian filtering. In *Proc. of the 2003 Spam Conference*, 2003.

[7] G. Salton and M. J. McGill. *Introduction to modern information retrieval.* McGraw Hill, 1983.

[8] A. S. Tanenbaum. *Structured Computer Organization (4th Edition).* Prentice-Hall, 1999.

[9] D. E. Knuth. *The Art of Computer Programming, Vol.3 – Sorting and Searching.* Addison-Wesley, 1973.