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

2006－DBS－139 (6)
2006－ＦＩ－ 83 (6)
2006／5／30

部分文字列の数え上げによるブログスパムの検出

成澤和志[†] 山田泰寛[††] 池田大輔[†††]

ブログの増加が著しい近年、ブログスパムが大きな問題であり、スパム検出の技術の発達が求められている。スパム検出に関する研究は内容解析やリンク解析によるものが多く、複雑な処理やアルゴリズムを使用する。我々はブログスパムの内容ではなく、コピーされ大量に生成される性質に着目した手法を提案する。テキストの部分文字列を数え上げた時、出現頻度と異なり数にはジップの法則が成り立つことを利用して、自然言語の知識を必要としない、高速なスパム検出の技術を得ることができる。また、我々は人工的なデータによる本手法の正当性を調べ、実際のブログデータから本手法によりブログスパムを検出することに成功した。

Blog Spams Detection by Counting Substring

KAZUYUKI NARISAWA [†] ,YASUHIRO YAMADA [††] and DAISUKE IKEDA[†††]

Blog spam detection is a key for the blog spam problem as the number of blog sites is extremely increasing. Existing methods for blog spam detection are based on contexts or link structures analysis, and does not work well completely. We suggest a method utilizing the fact that spams are massproduced at a low cost instead of their context. Our method does not need background knowledge of blog entries, such as natural languages, because of using Zipf's law for the frequency and the vocabulary size of substrings. We present the validity of our method by artificial data set, and succeed to detect blog spams from actual blog entries.

1. はじめに

ブログの増加は、作成の容易さ、コメントやトラックバック機能によるユーザ間の交流などによるものが大きい。しかし、そのインタラクティブなコメントやトラックバックなどの交流機能を利用したブログスパムが大きな問題となっている。

ブログスパムとは、一般にブログのコメントやトラックバックに書き込まれるブログの本文とは関係のない悪意的な文章である。ブログスパムは、広告やページランクの不正な操作、別サイトへのミスリードを目的とするための URL を含んでおり、同一内容を複数のブログに大量にコピーすることが多い。また、ブログスパムはその性質などにより、コメントスパム、リンクスパム、トラックバックスパム、スプログなどと呼ばれることもある[6),9)]。

スパムに関する研究☆は、(1) 制限法 (2) リンク解析 (3) 内容解析の 3 つに大きく分類することができる。

制限法は、スパムをあらかじめ規制する手法で、Google の提供するノーフォロータグ[5)] などがある。ブログ管理者はノーフォロータグを使うことにより、このタグの付いたコメントの URL によるページランクの操作を防ぐことができる。しかし、ブログスパム自体が消去できるものではなく、ブログスパムではないコメントによるページランクの上昇も不可能になってしまう。

リンク解析は、スパムに含まれているリンク情報を使ってスパムを検出する手法で、リンクファームの検出などがある[2),3),18)]。リンクファームはスパマー達によって作られる悪意的な相互リンクの集まりである。このような不自然な相互リンク集は、通常あまり存在しないためこれを検出することはスパム検出に効果的である。しかし、リンクによる解析であるため、内容的にそのリンクが不正であるかどうかの判断が困難である。

内容解析は、自然言語における単語や語句の出現や頻度に着目した手法で、ベイジアンフィルター[17)] などがある。これは内容的にスパムかスパムではないかという例を与えることで、学習に基づき今後の入力に対してスパムかどうかを判断するものである。また、テキスト中の単語の出現頻度の違いによる検出や、頻度がある閾値以上のものをスパムとする手法もある[4),7),15),19)]。しかし、このような学習型や単純な頻度に対しての手法は、学習していない単語や URL だけを含むスパムや、出現頻度

† 九州大学大学院システム情報科学府
Graduate School of Information Science and Electrical Engineering, Kyushu University
†† 九州大学ユーザーサイエンス機構
Kyushu University User Science Institute
††† 九州大学付属図書館
Kyushu University Library
☆ ブログスパム以外のスパムに関するものも含む。

が閾値より少しでも少ないスパムを検出できないなどあまり効果的な方法とは言えない。

我々はこのような手法とは異なり、ジップの法則[20),21)]を基にしたテキストの部分文字列の出現頻度と異なり数を用いた手法を提案する。ブログスパムは生成の際、広告やページランクの不正操作という目的を達成するために、同一内容のコピーを大量配布しなければならないという性質がある。この性質に着目することで、自然言語で書かれた内容に依存することなく、また、部分文字列の数え上げには接尾辞配列[8),12)]を使うことで高速にブログスパムを検出することができる。

現在、池田らによる部分文字列増幅法[10),11)]というテンプレート検出の手法が存在する。この手法はテキストの部分文字列の出現頻度と総出現数をグラフにプロットし、グラフ中に出現するピークを見つけることでテンプレートを検出する。テンプレートとブログスパムが、高頻度で十分な長さをもつという同様の性質を持つため、部分文字列増幅法をブログスパム検出に適応する。

部分文字列増幅法をブログスパム検出に適用させるにはいくつかの問題がある。テンプレートは入力サンプル中のほとんどに出現する文字列であるのに対して、ブログスパムは入力サンプル中☆にほとんど出現しない文字列であり、部分的には高頻度に見えるがサンプル全体に対しては非常に少ないため部分文字列増幅法では検出が難しいかもしれない。我々は、この問題に対して実験的に適用可能性を調べる。また部分文字列増幅法ではグラフの視覚的判断による検出であるため、自動的な検出が難しい。我々は条件式を用いることで自動的な検出を可能にし、さらに総出現数ではなく異なり数を使った手法を提案する。

本研究では、英文中のアルファベット出現確率に従って生成した人工的データと実際のブログからした収集したデータを用いた実験を行うことによってブログスパムが自動的に検出できることを示す。

2. 部分文字列増幅法

池田らは大量の Web ページから多くのページに出現するテンプレートを検出するため部分文字列増幅法[10),11)]を開発した。テンプレートとは Web ページを特徴付ける枠組みであり、ある程度の長さと、ほとんどのページに出現するという特徴があり、部分文字列増幅法はこのような十分な長さと出現頻度を兼ね備えた部分文字列を検出することができる。

収集した全てのテキストページを入力とし、テキストに対して全ての部分文字列を数え上げる。部分文字列の数え上げには接尾辞配列[12)]を使うことで線形時間で数え上げることができる。数え上げた部分文字列に対して

☆ ブログ 1 サイトを考える場合。

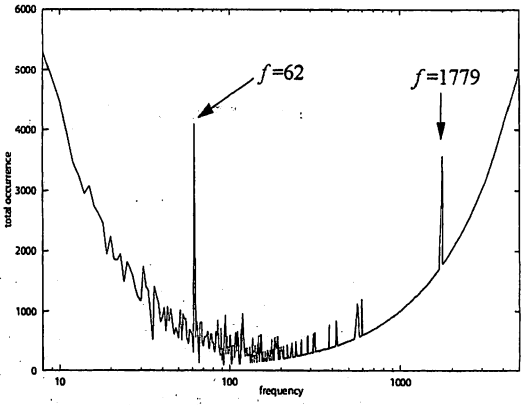

図 1 部分文字列増幅法によるグラフ。頻度を横軸、総出現数を縦軸にとる

出現頻度を横軸に、その頻度での部分文字列の総出現数を縦軸にしたグラフにする。このとき横軸は対数である。総出現数とは、ある文字列がテキスト中に出現した総数であり、出現頻度×異なり数で表すことができる。

図 1 は部分文字列増幅法によるグラフ☆☆の例である。本来、テンプレートが含まれない場合は滑らかな曲線になる。図 1 のグラフ中の $f = 62$ のところに表れているピークがテンプレート部分によるものである。しかし、$f = 1779$ のところに出現しているピークはテンプレートでなく出現頻度が高いことによって、総出現数が増加しただけであり、十分な長さは持っていない。

このように部分文字列増幅法をスパム検出に適用する場合、次の 3 つの問題が考えられる。(1) テンプレートに比べ、ブログスパムは全テキストに対する数が少ない。(2) テンプレートに比べ、ブログスパムの長さは短い。(3) 自動的に検出できない。

テンプレートは入力テキスト中の大部分に出現しているため、出現頻度が入力に対して相対的に高い。しかし、ブログスパムは 1 ブログサイト中、もしくは複数のブログサイト中であっても、全ての入力コメントエントリ数に比べると非常に少ないものである。 またテンプレートは HTML タグや URL、またタイトルやサイトマップなどのある程度長い同一の文字列が存在するのに対して、ブログスパムはコメントの表示上テンプレートほど長いものではなく、コメントの入力、表示などの制限によってはテンプレートに比べて非常に短いものになってしまう。

図 1 にもあるように、部分文字列増幅法ではグラフによる判断を要するため自動的な検出ができないだけでなく、$f = 1779$ に見られるように十分な長さがなく出現頻度が高いだけで視覚的にはピークに見えるようなものま

☆☆ 縦軸に総出現数、横軸に出現頻度をとり、横軸は対数スケールとする。

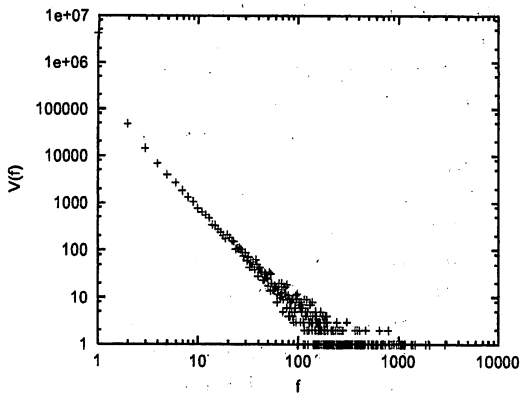

図 2 ジップの法則. 夏目漱石の "こころ" の中の文字列の出現頻度 f を横軸、異なり数 $V(f)$ を縦軸にとる

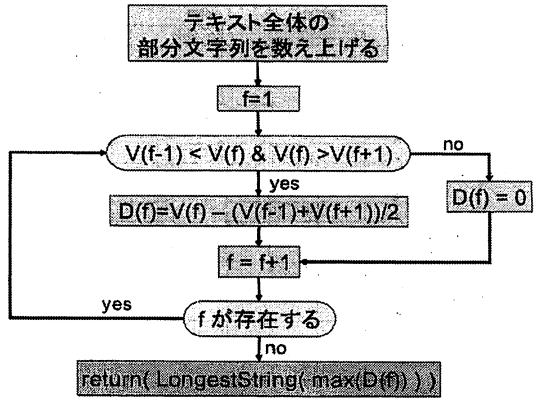

図 3 検出の流れ

でピークと誤認してしまうことがある。このグラフによる判断は視覚的な印象は大きいが自動検出を行うには不向きである。

本研究では、自動的にブログスパムを検出するための手法を提案する。また上記で述べた (1),(2) の問題については人工的なデータを用いることで実験的に部分文字列増幅法がスパム検出に適用できることを示す。

3. 提案手法

この章ではジップの法則に基づいた手法を提案する。

3.1 ジップの法則

ジップの法則は WWW のリンク [1] などでも見られる法則であり、サンプル S が与えられたとき、S 中の単語に関して、出現頻度 f を数える。各 f に対する異なり数を $V(f)$ とする。このとき, 部分文字列増幅法で用いる総出現数は $T(f) = f \times V(f)$ と書くことができる。

出現頻度 f と単語の異なり数 $V(f)$ にはジップの法則 [20),21)] が成り立つ。つまり $f, V(f)$ に関して次の式が成り立つ。

$$V(f) = bf^{-a} (a > 0)$$
$$\log V(f) = \log b - a \log f$$

図 2 は夏目漱石の "こころ" の中の文字列に関しての出現頻度 f と異なり数 $V(f)$ をグラフにしたものである。

このジップの法則は、単語単位だけではなく、部分文字列単位でも同様に成り立つことが知られている [10),11)]。我々は、単語を部分文字列に拡張してもこの法則が成り立つことを利用した異なり数 $V(f)$ を用いた手法を提案する。

3.2 条件式

2 章で述べた問題点を解決し、自動的にブログスパムを検出するため、部分文字列の出現頻度 f と異なり数 $V(f)$ を用いる。異なり数を用いることで、総出現数のように十分な長さがないものによるピークを検出することがなくなる。総出現数 $T(f)$ ではなく、異なり数でも部分文字列増幅法のように特異な値を示す。これを次に提案する条件式によって判断する。

$$D(f) = V(f) - \frac{V(f-1) + V(f+1)}{2}$$

ピーク (異常値) を検出するのに前後の平均との差分を考える。また前後の値よりも大きいピーク検出に関する条件式であるため各 f に対して、

$$V(f-1) < V(f) > V(f+1)$$

であることを調べ、この条件に従わない場合 $D(f) = 0$ とする。これにより $D(f)$ の値が最大値であるものの最長の文字列を検出する。図 3 はこの流れを表したものである。これにより、十分な長さを持ち、複数回出現する文字列を自動的に検出することができる。

3.3 複数検出

スパムは通常 1 種類だけではないため、複数のスパムを検出するための手法が必要となる。前節で述べた $D(f)$ による手法では 1 つのスパムだけしか検出ができず、また隣接した頻度に複数のスパムが存在する場合単純な $D(f)$ の値の高いものでは検出できない☆。また複数のスパムが同一の頻度で出現した場合、長さの長いものだけしか検出できない。

そこで検出した文字列をサンプルから全て削除し、再び $D(f)$ を計算しなおす。この操作を繰り返すことにより、複数のテンプレートやスパムなどの文字列を検出することができる (図 4 参照)。図 5 に複数スパム検出のアルゴリズムを示す。関数 `Count(T)` は、全ての部分文字列を数え上げる関数であり、接尾辞配列 (suffix array) を使うことによって線形時間で計算可能である [12)~14)]。

このアルゴリズムにより、スパムのような十分な長さ

☆ $V(f)$ と $V(f+1)$ の値がともに高いとき、$V(f) > V(f+1)$ ならば $D(f+1) = 0$ となる

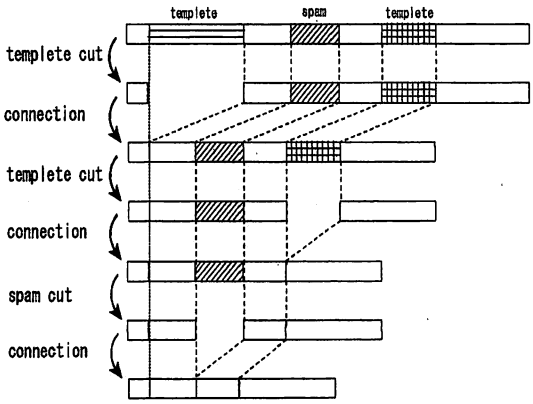

図 4　複数スパム検出の模式図

を持ち、複数回出現する文字列を完全に検出することができる。

4. 実　　験

前節で示したアルゴリズムがスパム検出に適応できることを示すためここでは 3 つの実験を行う。

4.1 長さと頻度

提案アルゴリズムがどのくらいの長さと頻度を持つスパムに対して適応できるのかを調べるために、人工的にスパムを埋め込んだランダムデータに対して実験を行う。

データに使う文字は a〜z とスペースを合わせた 27 種類のアルファベットで、各文字の出現確率は表 1 の通りである [16]。データは 1 サンプルを 100 ファイルとし、1 ファイルの文字数は 100 文字とする。また、各サンプルごとに埋め込むスパムの長さと数をパラメータとする。1 サンプル中に含むスパムの文字数を 4〜50 文字、スパムの数を 2,4,,100 個と変化させ、全部で $47 \times 50 = 2350$ のサンプルを用意する。これによりあらかじめ答えをわかったデータを作ることができる。例えば、1 サンプル中に 10 個のスパムを埋め込んだ場合、$D(10)$ の値が最大になると考えられる。

図 6 は提案手法によって検出できたサンプルを黒いセルで、検出できなかったサンプルを白いセルとして表している表である。ただし、ここで検出できたサンプルとは $D(f)$ が最大となるときの f と埋め込んだスパムの数が等しい事を言う。

提案手法では、全 2350 サンプル中、2054 サンプルに対してスパム検出が可能であった。長さが 10 文字程度あり、頻度が 10 程度あるような文字列であれば検出可能であることがわかる。また、同じデータに対して、部分文字列増幅法では 2140 サンプルに対してスパム検出が可能であった。これは、部分文字列増幅法ではグラフによる判断であるため、グラフ上でピークの存在する頻度が

```
function DiscoverString(var S: sample):sample
var
  OCC: hash(key:integer, value:integer);
  V: hash(key:integer,
          value:list of tuples of integers);
begin
  T:=w1$w2$...$w$; // S=w1, w2, ..., wn
  (OCC, V):=Count(T);
  for f in keys(V) do begin;
    for w in V{f} do begin;
    //w is denoted by (c, l, h) of integers
      if(|w-1|<|w| & |w|>|w+1|);
        D(f) = |w| - (|w-1|+|w+1|)/2;
      else;
        D(f) = 0;
    end;
  end;
  maxf = 0;
  maxD = 0;
  for f in (D(f)!=0) do begin;
    if(maxD < D(f))
      maxD = D(f);
      maxf = f;
  end;
  return(V(maxf));
end;

Algorithm SpamDetectionAlgorithm(Var S:sample)
begin
  while(spam ∈ S) do begin;
    string = DiscoverString(S);
    remove(string,S,newS);
    S = newS;
  end;
end;
```
図 5　スパム検出アルゴリズム

埋め込んだスパムの数付近にあるものでさえスパムとして誤検出した場合も考えられる。

部分文字列増幅法との大きな違いは、人手が要らず自動で高速にスパムを検出できたことである。

4.2 相対的な少なさ

前節の実験により、検出できるスパムの長さと数の下限がわかった。しかし、実際の Web 上のデータにおいてブログスパムは Web 全体や、コメント全体の数に比べ非常に少ない。そのため、相対的に非常に少ない場合でも検出が可能であるかを調べる。

データは前節と同様に人工的なデータを使う。文字の

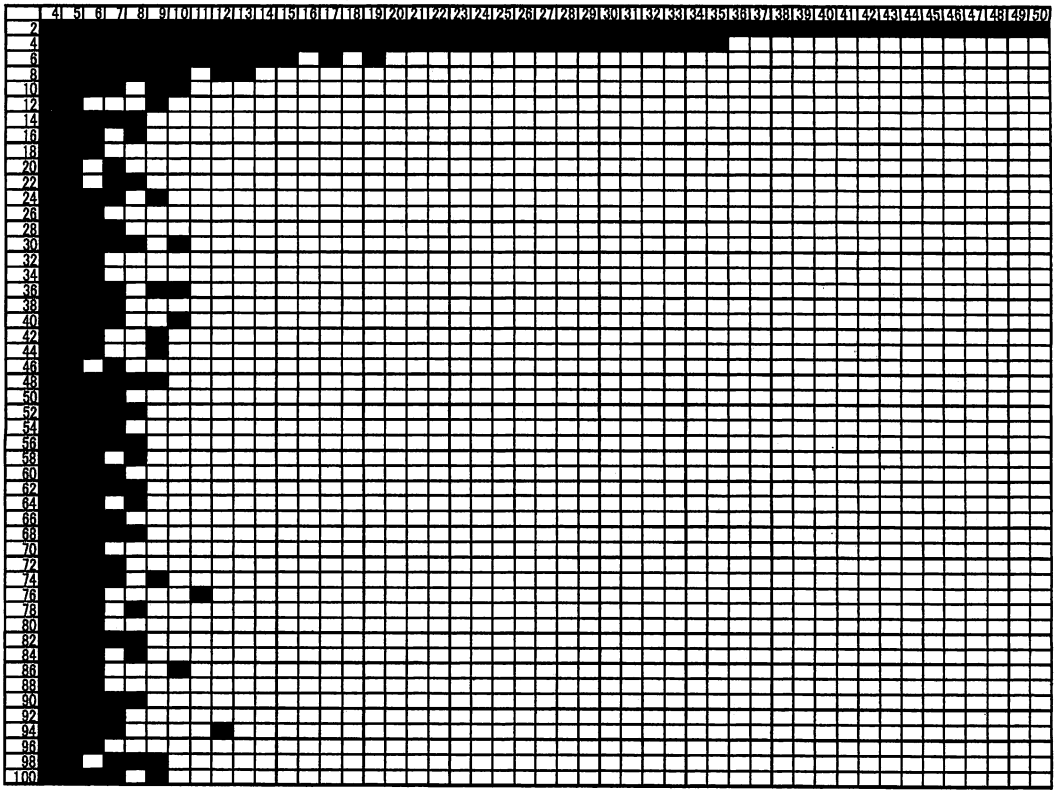

図 6 スパム長さ (横軸) と数 (縦軸) をパラメータとしたときの検出結果。黒いセルは検出可能、白いセルは検出不可能

表 1 アルファベットの出現確率

letter	probability	letter	probability
a	0.0668	o	0.0654
b	0.0118	p	0.0162
c	0.0226	q	0.0010
d	0.0310	r	0.0559
e	0.1073	s	0.0499
f	0.0239	t	0.0856
g	0.0163	u	0.0201
h	0.0431	v	0.0075
i	0.0519	w	0.0126
j	0.0011	x	0.0014
k	0.0034	y	0.0162
l	0.0278	z	0.0006
m	0.0208	space	0.1817
n	0.0581		

出現確率、1 ファイルの文字数は先ほどと同じとする。ここではパラメータとして 1 サンプル中のファイル数を考え、1000,10000,30000,50000,100000,200000 の 6 種類のサンプルを用意する。このとき、埋め込むスパムは (長さ, 数) =(20,50),(30,100),(40,101),(50,102),(30,150) の 5 つを全てのサンプルに等しく埋め込むものとする。

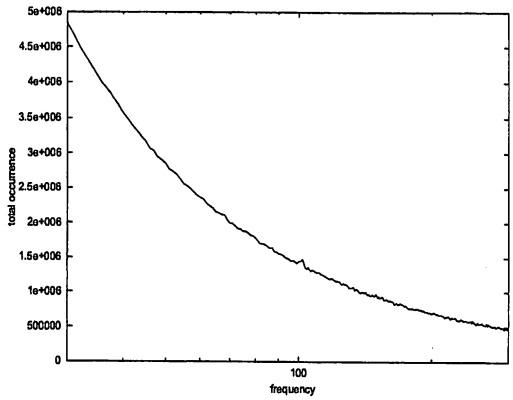

図 7 ファイル数 200000 のサンプルに対する部分文字列増幅法の結果。ただし $30 < f < 300$

図 7 は部分文字列増幅法による、ファイル数 200000 のサンプルに対する結果のグラフである。ただし、横軸は $30 < f < 300$ の範囲の対数スケールとしてある。これは、人工的にスパムを埋め込むことでピークが出現する

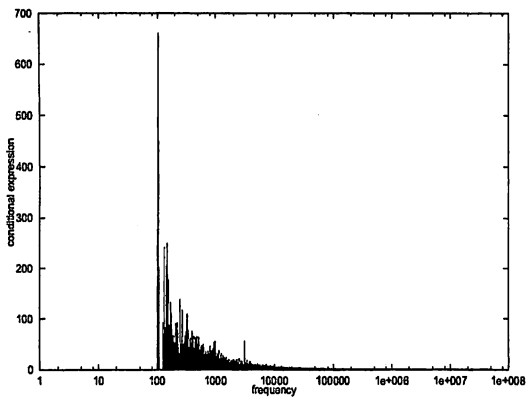

図 8　ファイル数 200000 のサンプルに対する本手法の結果。縦軸は $D(f)$、横軸は f、横軸は対数スケールとする

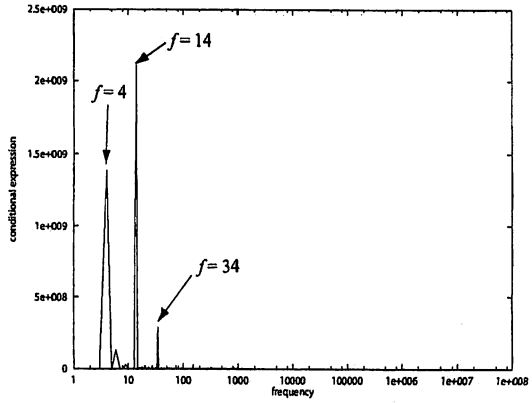

図 9　Arianna's Blog に対する本手法の結果のグラフ。縦軸は $D(f)$、横軸は f、横軸は対数スケールとする

頻度を知っているため、グラフの範囲を制限することで多少のピークでも検出できるが、実際未知のデータに対して全範囲でのグラフではこのような結果を出すことは不可能である。

図 8 は本手法による結果であり、縦軸を $T(f)$ ではなく $D(f)$ としたグラフである。ただし、横軸は対数スケールとする。これによると範囲を制限することなく検出できることは明らかである。このことから、本手法では相対的に非常に少ないスパムでも検出が可能であることがわかる。

4.3 実際のブログ

これまでの実験で、人工的データに対して、本手法がスパム検出に対して有効であることがわかった。そこで、実際のブログデータに対してスパム検出が可能であるかを調べる。

データは "Arianna's Blog☆" から取得したテキストデータ 27,190 ファイル (151.09MB) を対象とする。テンプレートの検出を防ぐため事前に HTML のタグは取り除いておく。このデータに対して、どれくらいのスパムが含まれているかはわからない。

図 9 は、本手法による結果によるグラフである。縦軸は $D(f)$、横軸は f である。ただし、横軸は対数スケールとする。いくつかの顕著なピークが見られる。詳細なデータは表 2 に示す。rank は $D(f)$ の値によるランクである。これより、総出現数 $T(f)$ と $D(f)$ では異なる結果であることがわかる。

rank が最大である、出現頻度 $f = 14$ の文字列は次のものである。

```
&#8226; @ &#8226; www &#8226; Reply
http://www.scorpion.my100megs.com
http://www.termites.ownsthis.com
```

☆ http://www.ariannaonline.com/blog/index.php

表 2　Arianna's Blog に対する本手法の結果

rank	f	T(f)	D(f)
1	14	29807776796	2113140742
2	4	5808770272	1383153658
3	34	9931043442	289234115
4	6	983693718	129228621
5	150	3784914450	25203203
6	9	629262801	24948258
7	27	723522051	21702500
8	74	1501518016	20141172
9	316	3574873872	11310990
10	23	352597452	10934478
11	840	9605687280	10778713
12	18	308156562	10279396
13	16	255922624	5830974
14	92	525336836	5687658
15	334	1601688650	4794884
16	799	4169810014	4749966
17	284	1269888960	4469877
18	40	194224560	4378191
19	31	185106239	4256154
20	21	227548650	3644976

```
http://bradpitt.freewebpage.org
http://www.quentin-tarantino.741.com
http://www.depeshe-mode.greatnow.com
http://www.mickeyrourke.esmartweb.com
http://www.crocodile.ownsthis.com
http://www.gitler.150m.com
http://www.titanic.741.com
http://www.chacknorrice.freewebpages.org
http://www.george-clooney.741.com
http://www.penelopecruz.batcave.net
```

中略

http://milfseeker.com-top.net/terra.html
http://milfseeker.com-top.net/devon.html
http://teensforcash.com-top.net/FREE.html
http://teensforcash.com-top.net/KATHLEEN.html
http://teensforcash.com-top.net/ALEXIS.html
http://teensforcash.com-top.net/VIDEO.html
http://teensforcash.com-top.net/NICOLE.html
http://teensforcash.com-top.net/RANDI.html
http://teensforcash.com-top.net/MEGAN.html
http://teensforcash.com-top.net/KIMMIE.html
http://teensforcash.com-top.net/BIANCA.html
• Trackback URL:
http://ariannaonline.com/blog/
bblog/trackback.php?tbpost=2

これは他サイトへの誘導の URL だけのコメントであり、URL のリンク先もブログの内容とは関係ないところに誘導してあり、内容的にもスパムであることは明らかである。文字数は 60,609 文字である。

また、本手法は複数のスパムを検出に成功した。特徴的なものには図 10 に示すような中国語のものもあった。図 10 は検出された文字列の一部である。これは、英語で書かれているブログには明らかに不自然であり、機械的に貼り付けられていることがわかる。内容も他のサイトの広告などが含まれており、スパムだと言える。このスパムの文字数は 5,494 文字であり、出現頻度 f は 4 である。

5. 結　論

本研究では、複数のブログスパムの出現頻度と長さに着目することで、自動的に検出するための手法を提案した。この手法では人工的なデータを用いることで実験的に長さ 10 文字程度のものや、相対的に低頻度のものでも検出できることを示した。また、内容的な解析をしないため自然言語に依存しないブログスパムを検出することができた。

提案手法では条件式 $D(f)$ を使うことによってスパムを判断し、検出したスパムを削除することで再び別のスパムを検出することができる。しかし、この手法にはいくつかの問題点が残る。

スパマーは全く同一のスパムを作るのではなく、内容を一部だけ変更したものを生成することもある。このような完全に一致しないようなスパムに対しては提案手法は検出が難しくなる。さらに、提案手法は部分文字列増幅法を基にしているため、スパムではないテンプレートも検出してしまう可能性がある。

また、今回は削除して数えなおすという操作を繰り返したが、この操作をどこまで繰り返すかということについて考えなければならない。全ての $D(f)$ が 0 になるまで

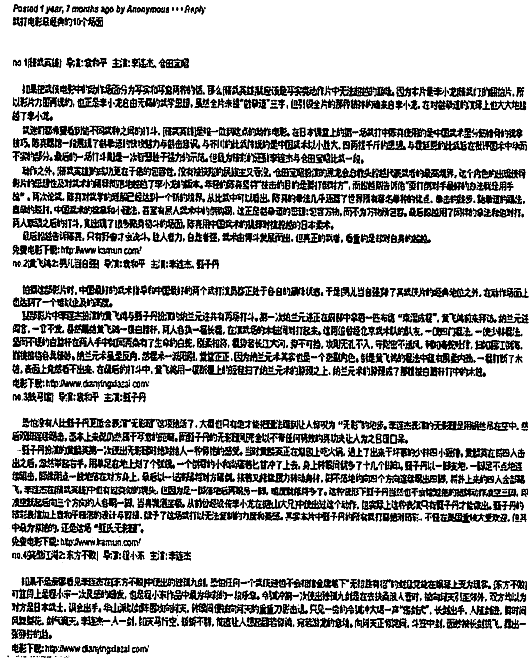
Posted 1 year, 7 months ago by Anonymous ··· Reply
免费电影下载: http://www.kamun.com/
电影下载: http://www.dianyingdazai.com/
免费电影下载: http://www.kamun.com/
电影下载: http://www.dianyingdazai.com/

図 10　Arianna's Blog の中の中国語スパムの一部

繰り返すと、短い文字列で明らかにスパムやテンプレートではないものまで検出されてしまう。

このような問題を解決していくことが今後の課題となるであろう。

参 考 文 献

1) R. Albert, H. Jeong, and A.-L. Barabási. Diameter of the World Wide Web. *Nature*, 401:130–131, 1999.
2) R. Baeza-Yates, C. Castillo, and V. López. Pagerank Increase under Different Collusion Topologies. In *Proceedings of the First International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, 2005. http://airweb.cse.lehigh.edu/2005/.
3) A. A. Benczúr, K. Csalogány, T. Sarlós, and M. Uher. SpamRank – Fully Automatic Link Spam Detection. In *Proceedings of the First International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, 2005. http://airweb.cse.lehigh.edu/2005/.
4) S.Chhabra, W.S. Yerazunis, and C.Siefkes. Spam Filtering using a Markov Random Field Model with Variable Weighting Schemas. In *Proceedings of the 4th IEEE International Conference on Data Mining*, pages 347–350, November 2004.
5) CNETNEWS.COM. Google Aims to Outsmart Search Tricksters, January 2005. http://news.com.com/2100-1024_3-5540740.html.

6) CNETNEWS.COM. Tempted by Blogs, Spam Becomes 'Splog', October 2005. http://news.com.com/2100-1032_3-5903409.html.

7) Z. Duan, Y. Dong, and K. Gopalan. DiffMail: A Differentiated Message Delivery Architecture to Control Spam. http://www.cs.fsu.edu/research/reports/TR-041025.pdf.

8) D. Gusfield. *Algorithms on Strings, Trees, and Sequences : Computer Science and Computational Biology.* Cambridge University Press, 1997.

9) Z. Gyöngyi and H. Garcia-Molina. Web Spam Taxonomy. In *Proceedings of the First International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, 2005. http://airweb.cse.lehigh.edu/2005/.

10) D. Ikeda. *Autoschediastic Text Mining Algorithms.* PhD thesis, Graduate School of Information Science and Electrical Engineering, Kyushu University, March 2004.

11) D. Ikeda and Y. Yamada. Gathering Text Files Generated from Templates. In *Proceedings of Workshop on Information Integration on the Web (IIWeb-04)*, pages 21–26, August 2004. http://cips.eas.asu.edu/iiweb.htm.

12) T. Kasai, G. Lee, H. Arimura, S. Arikawa, and K. Park. Linear-time Longest-Common-Prefix Computation in Suffix Arrays and Its Applications. In *Proceeding of the 12th Annual Symposium on Combinatorial Pattern Matching*, Lecture Notes in Computer Science 2089, pages 181–192. Springer-Verlag, 2001.

13) D.K. Kim, J.S. Sim, H.Park, and K.Park. Linear-time construction of suffix arrays. In *Proceeding of 14th Combinatorial Pattern Matching*, Lecture Notes in Computer Science 2676, pages 186–199, 2003.

14) G. Manzini and P. Ferragina. Engineering a Lightweight Suffix Array Construction Algorithm. *Algorithmica*, 40(1):33–50, 2004.

15) G. Mishne, D. Carmel, and R. Lempel. Blocking Blog Spam with Language Model Disagreement. In *Proceedings of the First International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, 2005. http://airweb.cse.lehigh.edu/2005/.

16) M. Nagao, S. Sato, S. Kurohashi, and T. Tsunoda. *Natural Language Processing.* IWANAMI KOZA Software Science 15. Iwanami Shoten, April 1996. (in Japanese).

17) M. Sahami, S. Dumais, D. Heckerman, and E.Horvitz. A Bayesian Approach to Filtering Junk E-mail. *AAAI Workshop on Learning for Text Categorization*, July 1998. ftp://ftp.research.microsoft.com/pub/ejh/junkfilter.pdf.

18) B. Wu and B. D. Davison. Identifying Link Farm Spam Pages. *In Proceedings of the 14th International World Wide Web Conference*, 2005.

19) K. Yoshida, F. Adachi, T. Washio, H. Motoda, and et al. Memory Management of Density-Based Spam Detector. *Symposium on Applications and the Internet (SAINT'05)*, pages 370–376, 2005.

20) G. K. Zipf. *The Psycho-Biology of Language: An Introduction to Dynamic Philology.* Houghton Mifflin, 1935.

21) G. K. Zipf. *Human Behavior and the Principle of Least Effort.* Addison-Wesley, 1949.