DBSJ Journal Vol.8, No.1
June 2009

ウェブサイト間の類似度を用いたウェブスパムの検出

Webspam Detection using Similarities of Websites

北村 順平 ♥　　青野 雅樹 ♦

Junpei KITAMURA　　Masaki AONO

様々なニュースサイトやブログの記事を流用することでウェブページを自動的に生成するスパムは検出の難しいウェブスパムの 1 つである。そのようなウェブスパムを検出するために、我々はウェブスパムが生成されるメカニズムに注目し、ウェブページ間の構造の類似度を求めることでウェブスパムを検出する方法を提案する。WEBSPAM-UK2007 データセット[1]を用いて実験を行った結果、提案手法を用いることでウェブスパム検出の精度を改善できることを確認した。また、ウェブスパム検出のワークショップである AIRWEB2008[2]の結果と比較したところ、AUC と F 値において良好な数値が得られた。

Among many types of web spams, it has been known to be very hard to detect a group of web spams that automatically generate web pages by duplicating existing news sites and blog articles. In this paper, we will focus on detecting this type of web spams and propose a method which takes care of the mechanism of how such a web spam is produced. Specifically, we attempt to detect this type of web spams by introducing new features for capturing structural similarities between web pages. We demonstrated our approach by using WEBSPAM-UK2007 datasets and showed that we could outperform the previously known methods appeared in AIRWEB2008, in terms of AUC and F-measure.

1. 序論

ウェブは近年の発達が著しい分野であり、人がウェブを通じてアクセス可能な情報は爆発的に増えている。検索エンジンは膨大な数のウェブページのインデックスを作成することで、検索クエリと関連性の高いウェブページを提示するシステムであり、ウェブの利用者が必要な情報にアクセスするための手段として重要性を増している。

検索エンジンの利用者のアクセスは表示位置が上位のウェブページ（ランキングの高いウェブページ）に集中する傾向にある。そのため、自分のウェブページに多くの顧客を呼び込むためには、検索結果の上位に表示されることが重要となる。e-commerce のウェブサイトを考えたとき、多くの顧客を呼び込むことが利益に直結するため、検索結果において高いランキングを得ることが求められる。

ランキングを意図的に高める方法としては SEO(Search Engine Optimization) がある。SEO は検索に頻繁に使用される検索語を考慮してウェブページを作成したり、HTML の最適化を図ることでランキングを高めようとする行為である。しかし、ウェブページの内容と関係のないキーワードを埋め込んだり、リンクファームを構築する等の過剰な SEO により高いランキングを狙うスパマーが存在する。過剰な SEO が施されたウェブページが高いランキングを獲得することは、検索エンジンの精度を低める要因となり望ましくない。そのため、検索エンジンは過剰な SEO が施されたウェブページをウェブスパムとして識別し、ウェブスパムに対してランキングを低下させるといったペナルティを与えている。

ウェブスパムを検出するための研究は活発になされており、自動的に検出することのできるウェブスパムは少なくない。そのような研究の多くは、ウェブページのテキストやウェブページ間のリンク構造を利用することでウェブスパムの検出を行っている。例えば、SpamRank は、spam page は多くの spam page にハイパーリンクを持つという知見に基づいてウェブスパムの検出を行っており、効率的にウェブスパムを検出できることが報告されている [1]。

しかし、テキストやリンク構造だけでは検出の難しいウェブスパムが存在する。そのようなウェブスパムの一例として、複数のニュースサイトやブログの記事を流用することでウェブページを大量に生成するタイプのウェブスパムがある。このタイプのウェブスパムは、流用したテキストをもとにウェブページを生成することで様々な検索クエリに対するスパミングを行っており、単純にテキストから検出することが難しい。そこで我々は、ウェブページのテキストだけではなく、ウェブページの構造からウェブスパムを検出する方法を提案した。提案手法は、これらのウェブスパムが自動的に生成されているという点に注目し、ウェブサイト間の類似度を計算することでウェブスパムの検出を行う。関連研究として Urvoy らの研究があるが [2]、我々は Javascript のコード間の類似度を考慮している点や、類似度をもとにした素性を提案し機械学習に応用している点で新規性がある。

本論文の構成は以下の通りである。2 章では、ウェブスパムの分類や種類などの基本的な事柄について説明する。3 章では、提案素性を求めるための前処理と LSH を用いてスケーラブルに文書間の類似度を求める方法を説明する。4 章では、提案素性の有効性の検証と関連研究との比較を行う。5 章では、結論と今後の課題を述べる。

♥ 非会員　豊橋技術科学大学大学院工学研究科情報工学専攻
kitamura@kde.ics.tut.ac.jp
♦ 正会員　豊橋技術科学大学情報工学系　aono@ics.tut.ac.jp

[1] UK2007, http://barcelona.research.yahoo.net/webspam/
[2] AIRWEB2008, http://airweb.cse.lehigh.edu/2008/

2. ウェブスパムの種類

Gyöngyi らはスパミングを「ウェブページのランキングを不当に高める全ての意図的な行為」と考え、スパミングに用いられる技術を Boosting Techniques と Hiding Techniques に分類した [3]。Boosting Techniques は検索エンジンのランキングに影響を与える技術であり、Hiding Techniques はスパミング行為そのものを検索エンジンや訪問者から隠蔽する技術である。Boosting Techniques は Term Spamming と Link Spamming に分類される。

■Term Spamming　TFIDF は検索エンジンがウェブページの順位を決めるために使う重要な指標の 1 つがである。あるウェブページ p_j が与えられたとき、p_j に含まれる単語 t_i の TFIDF は以下の式により定義される。

$$TFIDF(i,j) = TF(i,j) \cdot IDF(i) \tag{1}$$

TFIDF スコアを求めるための TF と IDF には様々なものがあるが、以下に一例を示す。

$$TF(i,j) = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{2}$$

ここで、$n_{i,j}$ はウェブページ p_j に含まれる単語 t_i の出現回数、$\sum_k n_{k,j}$ は p_j に含まれる全ての単語の出現回数を示す。

$$IDF(i,j) = \log \frac{|D|}{|d|} \tag{3}$$

ここで、$|D|$ はコーパスにおけるウェブページの総数、$|d|$ は単語 t_i を含むウェブページの総数である。つまり、TF が単語の出現頻度に比例したスコアであるのに対し、IDF は多くのウェブページに出現する一般的な単語に対して重要度を下げるスコアとして働く。

スパミングの観点から TFIDF スコアを考えると、IDF はスパマーにより制御できないスコアであるため、スパミングは TF を高めることで達成される。例えば、ウェブページの内容とは関係のないキーワードを追加することで、特定のクエリに対するウェブページの関連性を高める方法がある。Ntoulas らは、ウェブページの圧縮率や特定の検索クエリに対する適合率等を用いる方法を提案している [4]。

■Link Spamming　検索エンジンはウェブページの順位を決めるとき、TFIDF に加えリンク情報も重視する [3]。検索エンジンがリンクベースの重要度を決めるときに用いる指標の 1 つが PageRank である。PageRank はウェブページが外部から得るリンクの数はそのウェブページの重要度と関連性があると仮定し、ウェブページの重要度を決定する。これは、重要度の高い複数のウェブページからハイパーリンクを得ているウェブページの重要度は高いという考えに基づく。リンクベースのスパムは、この PageRank を高めることで達成される。Link Spam の一例として Link Farm が挙げられる。Gyöngyi らは、PageRank が最適化されたグループを識別することでウェブスパムを検出する方法を提案している [5]。

3. 提案手法

序論において説明した、他のウェブページの内容を流用することでウェブページを生成するタイプのスパミングは、Term Spamming に分類することができる。このタプのウェブスパムは、流用した記事をもとにウェブページを作成することで様々な検索クエリに対する TF スコアを高めている。ウェブページ自体は様々なニュースサイトやブログから流用しているため、ウェブスパムである証拠を見つけることが難しい。流用した記事をもとにウェブページを生成するタイプのスパミングは、自動化された手法で大量に生成されることが多い。そのため、ある構造を持つウェブページが既にウェブスパムと分かっているウェブページの構造と類似している場合、そのウェブページもウェブスパムである可能性が高いと考えられる。そこで我々は、ウェブサイト間の構造類似度を用いて、このようなタイプのウェブスパムを検出する方法を提案する。

3.1 前処理

各類似度を求めるためには HTML ファイルに対する適切な前処理が必要となる。HTML ファイルとは HTML により記述されたファイルのことであり、ウェブブラウザによりレンダリングされる前の状態である。前処理では、この HTML ファイルを構文解析し、類似度計算に必要な情報のみを抽出することで後の処理を効率的に行う。実際には以下に示す 3 種類のドキュメントを HTML ファイルから作成する。

- *words* ドキュメント（テキスト類似度）
- *tags* ドキュメント（タグ類似度）
- *scripts* ドキュメント（スクリプト類似度）

words ドキュメントはテキスト類似度、*tags* ドキュメントはタグ類似度、*scripts* ドキュメントはスクリプト類似度を計算するために用いられる。タグ類似度とスクリプト類似度がウェブページの構造的な類似度に相当する。以下、3 種類の前処理についてそれぞれ説明を行う。前処理の説明のために、表 1 に前処理を施す前の HTML ファイルの例を示す。

表 1　前処理を施す前の HTML ファイルの例

```
<html> <head> <title>Loan Service, Low Cost Loan at UK</title> <meta name="description" content="low cost loans"/> <script type="text/javascript"> var siteid=1; var trackmode='default'; </script> <script type="text/javascript" src="../js/code.js"></script> </head> <body> <h2>Low Cost Menu</h2> <p>See how much money our customers have saved.</p> <table width="700" border="0" cellspacing="2" bgcolor="#CCCCCC" align="center"> <tr><td>You can reduce your monthly outgoings.</td></tr> </table> </body> </html>
```

■*words*　*words* ドキュメントは HTML ファイルの HTML タグ、Javascript、コメントを除いたものである。多くのケースでウェブページの本文テキストに相当する部分と言い替えることができる。*words* ドキュメントを抽出することで、ウェブページ間の類似度を本文に基づいて計算することが可能となる。*words* ドキュメントを抽出する例を表 2 に示す。

表 2　*words* 処理を行った例

loan service low cost loan at uk low cost menu see how much money our customers have saved you can reduce your monthly outgoings

■*tags*　*tags* ドキュメントは HTML ファイルの HTML タグと HTML タグの属性のみを抽出するものである。但し、<body> より前に出現するタグはウェブページ間での差異が小さく、差別化要因にならないため処理の対象から除外する。*tags* を抽出する例を表 3 に示す。

表 3　*tags* 処理を行った例

body h2 h2 p p table width 700 border 0 cellspacing 2 bgcolor cccccc align center tr td td tr table body html

■*scripts*　*scripts* ドキュメントは HTML ファイルの <script> と <noscript> のみを抽出するものである。*scripts* を抽出する例を表 4 に示す。

表 4　*scripts* 処理を行った例

script type text javascript var siteid 1 var trackmode default script

3.2　類似度尺度

例として、*words* 処理により得られたドキュメント集合 D を考える。このドキュメント集合 D を n-grams によりフレーズ単位で抽出し、ユニークなフレーズ（以下、要素）に要素番号を付与する。この操作により、ドキュメント集合 D は要素番号を用いた集合 $S_D \in \{1, ..., n\}$ により表現することができる。また、n-grams を用いることで、単語の位置関係を部分的に保持した状態で要素集合を構築することが可能となる。n はドキュメント集合 D に含まれるユニークな要素数に等しい。このとき、任意のドキュメント A、B 間の類似度 $sim(A, B)$ は以下の式により定義される。

$$sim(A, B) = \frac{|S_A \cap S_B|}{|S_A \cup S_B|} \tag{4}$$

この方法を用いて全てのウェブページ間の類似度を計算することは不可能ではないが、大規模なデータセットに対して全てのパターンの類似度を計算することは現実的ではない。そこで我々は、Locality-Sensitive Hashing(LSH)[6] を適用することで計算量の削減を行った。

3.3　Locallity-Sensitive Hashing

LSH は近似近傍探索を行うアルゴリズムの一つである。LSH では、互いに類似したドキュメント間において衝突の発生する確率が高く、類似していないドキュメント間において衝突の発生する確率が低くなるハッシュ関数 $h \in \mathcal{H}$ を用いて、ドキュメントを複数のハッシュ値として表現する。LSH はこのような関数群 $\mathcal{H}$ により得られたハッシュ値を用い、k 個のハッシュの連結値を関数 g として L 個定義する。この L は Locality-Sensitive Hashing の L を表す。この関数 g により、全てのドキュメントはハッシュ値として表される。このとき、互いに類似しているドキュメントは同一のハッシュ値を持つ確率が高く、類似していないドキュメントは異なるハッシュ値を持つ確率が高い。

■MinHash　MinHash は LSH において Jaccard 係数を扱うための手法である [7]。ここで *words* 処理により得られた全てのドキュメントの要素から構成される集合 S_n を考え、π を S_n のランダム順列を得る関数として定義する。このとき、任意のドキュメント A、B 間の類似度は以下の式により与えられる [8]。

$$\mathbf{Pr}_{h \in \mathcal{H}}[min\{\pi(S_A)\} = min\{\pi(S_B)\}] = \frac{|S_A \cap S_B|}{|S_A \cup S_B|} \tag{5}$$

min はドキュメントに含まれる全ての要素のうち、ランダム順列において最初に出現する要素の要素番号を得る関数である。128 通りの π を定義したとき、ドキュメント D の要約は以下の式により表される。

$$\overline{S}_D = (min\{\pi_1(S_D)\}, min\{\pi_2(S_D)\}, ..., min\{\pi_{128}(S_D)\}) \tag{6}$$

この式を適用することで、ドキュメント D は 128 通りのハッシュ値の集合 $\overline{S}_D$ として扱うことができる。よって、任意のドキュメント A、B 間の Jaccard 類似度は $\overline{S}_A$、$\overline{S}_B$ において一致するハッシュ値の割合により求めることができる。

3.4　素性の定義

LSH を用いることで、任意のドキュメントの近似近傍のドキュメントの候補を得ることができる。LSH は *words* ドキュメント、*tags* ドキュメント、*scripts* ドキュメントのそれぞれに独立して適用する。次に LSH を用いてウェブページ間の類似度を *words*、*tags*、*scripts* のそれぞれについて計算し、ある閾値以上の類似度をもつウェブページの組合せを求める。閾値はハッシュの結合値が一致した数により与えられる。

ラベルの付与されているウェブページでウェブスパムのものを P_s、ウェブスパムでないものを P_{ns} としたときの素性を以下に式により定義する。

$$f = \frac{P_s - P_{ns}}{P_s + P_{ns}} \tag{7}$$

この式から、*words* に関する素性、*tags* に関する素性、*scripts* に関する素性を求めることができる。

4. 実験

実験は主に Java 言語と Python 言語を用いて行った。機械学習の SVM には LIBSVM[3]、ランダムフォレストには Orange[4]を利用した。計算の実行環境は、OS が Linux、CPU が Intel(R) Core(TM)2 Quad CPU Q6600 2.4GHz、メモリが 6GB である。データベースには Postgresql の 8.3 を利用した。

4.1 データセット

実験に用いるデータセットは広く公開されており、関連研究との比較が可能なものが望ましい。そのような条件を満たすデータセットとして、WEBSPAM-UK2007 を用いた。WEBSPAM-UK2007 はウェブスパム検出の研究のために Yahoo! Research Barcelona において公開されているもので、Universitá degli Studi di Milano において.uk ドメインを対象にクロールされたものである。ウェブページは WARC（Web ARChive）フォーマット[5]を用いてアーカイブされており、WARC フォーマットを扱うために Universitá degli Studi di Milano において公開されている LAW Library[6]を用いた。

データセットは 114,529 のウェブサイトから成る 105,896,555 のウェブページを含んでおり、圧縮された状態で 560GB の膨大なデータ量である。我々は各ウェブサイトのウェブページ数の上限を 400 とした要約バージョン（46GB）を実験に使用した。

データセットの一部のウェブサイトには有志によりラベルが付与されている。ラベルは 0〜1 の値をとり、0.5 より小さな値はウェブスパムではないウェブサイト（ノンスパム）を示し、0.5 より大きな値はウェブスパムを示す。0.5 はボーダーラインを示す。表 5 にラベルの分布を示す。このラベルのうち、2/3 が訓練用データ、1/3 がテスト用データとして機械学習用に提供されている。本研究ではこの訓練用データとテスト用データを使って実験と評価を行った。ラベルが undecide 若しくは borderline となっているものは評価の対象から除外した。

表 5 WEBSPAM-UK2007 データセットにおけるラベルの分布

ラベル	ラベル数	割合
Spam	344	5.3%
Non-spam	5709	88.1%
undecided or borderline	426	6.6%
合計	6479	100%

■データセットに付属の素性 データセットでは、それぞれのウェブサイトに対して 275 個の素性が計算されている。素性は内容ベース（テキストベース）の素性が 96 個、リンクベースの素性が 179 個である。内容ベースの素性はウェブページの圧縮率やクエリに対する適合率、リンクベースの素性は PageRank や TrustRank、Truncated PageRank 等が含まれる。これらの素性は Castillo[9]、Becchetti[10]、Ntoulas[4] らによって提案されたものである。この 275 個の素性を用いて機械学習を行ったものをベースライン検出器として後章での検討に用いる。

4.2 前処理、LSH の適用

まず、*words* ドキュメント、*tags* ドキュメント、*scripts* ドキュメントを抽出するために前処理を適用した。n-grams の n の値は、*words* が 5、*tags* と *scripts* が 7 である。*tags* と *scripts* は *words* に比べて共起する単語が多いため、n の値として *words* よりも大きな数値を選んだ。前処理の対象には、ラベルの付与されたウェブサイトと、それらのウェブサイトの近隣のウェブサイト（ラベル付きのウェブサイトへのハイパーリンクを持つウェブサイト）を選んだ。選ばれたウェブサイトの総数は 27,161 である。

次に LSH によりウェブページの近傍のウェブページ群を求め、素性を計算した。LSH のパラメータとしてハッシュ関数の数を 128、L を 42、k を 3 に設定した。ラベルは各ウェブサイトに付与されているため、ウェブサイトを構成しているウェブページの素性の平均値からウェブサイトの素性を求めた。LSH は、*words* ドキュメント、*tags* ドキュメント、*scripts* ドキュメントに独立して適用した。閾値の値は 8、16、32 を使い、それぞれの閾値について *words*、*tags*、*scripts* の素性を計算した。そのため、得られた素性の総数は 9 個となる

4.3 各素性の評価

提案した *words*、*tags*、*scripts* の 3 つの素性の評価を行う。評価尺度には機械学習の素性の評価に有効な F-score[11] を用いた。素性をラベルに対してベクトル表現したものを $x_k(k = 1, 2, ..., m)$ とし、ウェブスパムの数を n_+、ノンスパムの数を n_- としたときの i 番目の F-score は下式により表される。

$$F(i) \equiv \frac{(\bar{x}_i^{(+)} - \bar{x}_i)^2 + (\bar{x}_i^{(-)} - \bar{x}_i)^2}{\frac{1}{n_+ - 1}\sum_{k=1}^{n_+}(x_{k,i}^{(+)} - \bar{x}_i^{(+)})^2 \frac{1}{n_- - 1} + \sum_{k=1}^{n_-}(x_{k,i}^{(-)} - \bar{x}_i^{(-)})^2} \quad (8)$$

ここで、$\bar{x}_i$ は i 番目の素性の平均値、$\bar{x}_i^{(+)}$ は i 番目の素性でラベルがウェブスパムであるものの平均値、$\bar{x}_i^{(-)}$ は i 番目の素性でラベルが ノンスパムであるものの平均値を示す。$x_{k,i}^{(+)}$ は i 番目の素性でラベルがウェブスパムであるものの k 番目のウェブサイトの素性値、$x_{k,i}^{(-)}$ は i 番目の素性でラベルがノンスパムであるものの k 番目のウェブサイトの素性値を示す。この F-score が大きいほど、その素性はウェブスパムとノンスパムを大きく分け隔てており、機械学習において有用となる。

提案手法では、近傍のウェブサイト群にラベルが 1 つも付与されていない場合の素性を求めることができないため、そのウェブサイトを評価の対象から除外した。但し、評価の対象から外すのは素性間の評価のみで、次節においては評価の対象に含まれる。*words*、*tags*、*scripts* に閾値として 8、16、32 を用いたときの評価の対象となったウェブサイト数を表 6 に示す。閾値はハッシュ値の一致した関数の数を示す。割合 A はラベルの付与されている 6,053 のウェブサイトに対して各素性が求められた割

[3] LIBSVM, http://www.csie.ntu.edu.tw/~cjlin/libsvm/
[4] Orange, http://www.ailab.si/orange/
[5] http://bibnum.bnf.fr/WARC/warc_ISO_DIS_28500.pdf
[6] LAW Library, http://law.dsi.unimi.it/software/

合、割合 B はラベルの付与されている 6,053 のウェブサイトに対して *words*、*tags*、*scripts* のいずれかが求められた割合である。この表から、提案した素性は約 3 割のウェブサイトに対して計算されたことが分かる。但し、訓練データの数が増えれば、類似ページの見つかる確率が高まり、より多くのウェブサイトに対して素性を求めることが可能となる。

表 6　素性が計算されたウェブサイト数

閾値	素性	サイト数	割合 A(%)	割合 B(%)
8	*words*	914	15.1	35.78
	tags	798	13.2	
	scripts	1267	21.0	
16	*words*	909	15.0	34.89
	tags	730	12.1	
	scripts	1231	20.3	
32	*words*	875	14.5	32.68
	tags	509	8.4	
	scripts	1123	18.6	

表 7 に各素性の F-score を値を示す。この結果からタグ間の類似度である *tags* が最も高いスコアを示しており、HTML タグの類似度がウェブスパムの検出において有用な素性であることが確認できる。このことより、ウェブスパム同士は類似したタグ構造を持つ傾向があると考えられる。

表 7　各素性の F-score

素性	閾値	F-score
words	8	0.192
	16	0.180
	32	0.162
tags	8	0.462
	16	0.530
	32	0.502
scripts	8	0.213
	16	0.215
	32	0.216

4.4　機械学習による評価

機械学習による提案素性の評価実験について述べる。評価尺度には AUC(Area Under Curve) と F 値を用いる。表 8 はラベルのスパムとノンスパム、予測のスパムとノンスパムの対応を示している。AUC は ROC カーブの曲線下面積を求めたものである。AUC と F 値の計算は PERF[7]を用いて行う。

$$TruePositiveRate = \frac{TP}{TP + FN} \tag{9}$$

$$FalsePositiveRate = \frac{FP}{FP + TN} \tag{10}$$

[7] http://kodiak.cs.cornell.edu/kddcup/software.html

表 8　混同対照表

		ラベル	
		Spam	Non-spam
予測	Spam	TP	FP
	Non-spam	FN	TN

評価は上述した 275 個の素性と我々の計算した 9 個の素性を使って行う。275 個の素性のうち内容ベースの素性を C、リンクベースの素性を L、我々の計算した素性を Sims とおき、C と L の全ての素性を用いて学習を行った検出器 (C+L) をベースラインとする。それに対し、C と L に Sims を加えて学習を行った検出器 (C+L+Sims) の精度を比較する。機械学習にはサポートベクターマシン (SVM) とランダムフォレスト (RF) を用いた。

機械学習においては、データセットのスパムとノンスパムの割合が約 1:17 とアンバランスであるため、訓練に用いるノンスパムの数を調整し、精度の高い検出器を構築できる割合を探した。表 9、表 10 に AUC による提案手法の評価結果を示す。この結果より、提案素性を加えることで検出器の精度が向上することを確認できる。また、AUC が最も高かった RF の 1:5 の ROC カーブを図 4.5 に示す。この ROC カーブから、提案素性では False Positive Rate が 0 付近の状態で True Positive Rate が 0.4 であることが分かる。これはノンスパムをウェブスパムと誤検出することなく、4 割近いウェブスパムを検出できることを示しており、誤検出が重大な問題となるウェブスパムの検出において重要な改善である。

表 9　機械学習による評価 (SVM)

	1:1	1:3	1:5	1:10	1:15
C+L	0.784	0.762	0.765	0.727	0.710
C+L+Sims	0.823	0.817	0.806	0.823	0.796

表 10　機械学習による評価 (RF)

	1:1	1:3	1:5	1:10	1:15
C+L	0.802	0.816	0.812	0.796	0.768
C+L+Sims	0.829	0.857	**0.859**	0.838	0.813

4.5　関連研究との比較

ウェブスパム検出のワークショップである AIRWEB2008 との結果の比較を行う。AIRWEB2008 では、実験に使用した WEBSPAM-UK2007 データセットが用いられたため、前節の実験結果を使って比較することができる。表 11 に AIRWEB2008 の結果との比較を示す。これより、我々の検出器は AUC と F 値において最も良好な結果を示していることが分かる。但し、AIRWEB2008 の参加者はウェブスパム検出の予測結果を 2 通りしか提出できないルールだったため、我々の予測の仕方は不公

平だが、それを勘案しても比較的良い精度でウェブスパムを検出できていることが分かる。

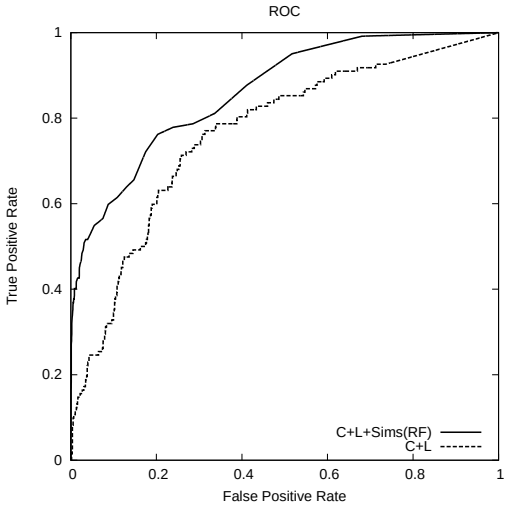

図 1　ROC カーブ

表 11　AIRWEB2008 との比較

	AUC	F 値
Geng et al.	0.848	0.470
Tang et al.	0.824	0.368
Abernethy and Chapelle	0.809	0.359
Siklosi and Benczur	0.796	0.318
Bauman et al.	0.783	0.257
Skvortsov	0.731	0.243
C+L+Sims(SVM)	0.823	**0.564**
C+L+Sims(RF)	**0.859**	0.533

5.　結論

本稿では、従来の方法では検出の難しい、他のウェブページの内容を流用することでウェブページを自動的に生成するタイプのウェブスパムを検出するための検討を行った。そのためにウェブページ間の構造に着目し、ウェブページ間のタグの類似度をもとに計算した素性がウェブスパムの検出に有効であることを実験により確認した。

今後の課題として、異なるデータセットや言語を用いた検討が期待される。また、今回の実験では *words* や *tags* といった類似度を計算する際に *n*-grams によるフレーズの抽出を行ったが、異なる方法でフレーズを抽出した際の検討が期待される。

[文献]

[1] A. A. Benczúr, K. Csalogány, T. Sarlós, and M. Uher. Spamrank: fully automatic link spam detection. In *Proceedings of the 1st International Workshop on Adversarial Information Retrieval on the Web*, 2005.

[2] T. Urvoy, E. Chauveau, P. Filoche, and T. Lavergne. Tracking web spam with html style similarities. In *ACM Transactions on the Web*, 2008.

[3] Z. Gyöngyi and H. Garcia-Molina. Web spam taxonomy. In *Proceedings of the 1st International Workshop on Adversarial Information Retrieval on the Web*, 2005.

[4] A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly. Detecting spam web pages through content analysis. In *Proceedings of the 15th international conference on World Wide Web*, pp. 83–92, 2006.

[5] Z. Gyöngyi and H. Garcia-Molina. Link spam alliances. In *Stanford University*, 2005.

[6] P. Indyk and R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pp. 604–613, 1998.

[7] A. Z. Broder, S. C. Glassman, M. S. Manasse, and G. Zweig. Syntactic clustering of the web. In *Elsevier Science Publishers Ltd*, pp. 1157–1166, 1997.

[8] A. Z. Broder, M. Charikar, A. M. Frieze, and M. Mitzenmacher. Min-wise independent permutations. In *Proceedings of the 30th annual ACM symposium on Theory of computing*, pp. 327–336, 1998.

[9] C. Castillo, D. Donato, A. Gionis, V. Murdock, and F. Silvestri. Know your neighbors: web spam detection using the web topology. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 423–430, 2007.

[10] L. Becchetti, C. Castillo, D. Donato, S. Leonardi, and R. Baeza-Yates. Link-based characterization and detection of web spam. In *Proceedings of the 2nd International Workshop on Adversarial Information Retrieval on the Web*, 2006.

[11] Y. W. Chen and C. J. Lin. Combining svms with various feature selection strategies. In *Taiwan University*, 2005.

北村 順平　Junpei KITAMURA
豊橋技術科学大学大学院工学研究科情報工学専攻博士前期課程。次世代光ネットワーク、IP ネットワークの信頼性構築、スパムブログ検出、ウェブスパム検出などの研究に従事。

青野 雅樹　Masaki AONO
豊橋技術科学大学情報工学系教授。1984 年東京大学理学系大学院情報科学専攻博士前期課程修了。Ph.D(Rensselaer Polytechnic Institute)。データベース、データマイニング、情報検索などの研究に従事。ACM、IEEE、情報処理学会、日本データベース学会、言語処理学会、人工知能学会、電子情報通信学会、各会員。