

## スプログ検出における HTML 構造の類似性の有効性の評価

片山 太一<sup>†1</sup> 芳中 隆幸<sup>†2</sup> 宇津呂 武仁<sup>†1</sup>  
河田 容英<sup>†3</sup> 福原 知宏<sup>†4</sup>

本研究では、ブログにおいてアフィリエイト収入を得ることを目的とするスパム (スパムブログ、スプログ) のうち、特に、同一のスパムブログ作成者が自動的に大量生成したと推測されるスプログの検出において、HTML 構造の類似性が効果的であることを示す。具体的には、ブログの HTML ファイルにおける DOM ツリーから、コンテンツの最小単位に相当するブロックを抽出し、複数のスプログの間でブロック構造の類似性を測定する。その結果、同一ブログホストにおけるスプログのうち、同一のスパムブログ作成者が自動的に大量生成したと推測されるスプログ同士では、ブロック構造が類似する傾向があることを示す。また、ブロック構造の類似性を素性として用いることにより、SVM によるスプログ検出の性能が向上する場合があることを示す。

### Evaluating Effects of Similarities of HTML Structures in Splog Detection

TAICHI KATAYAMA,<sup>†1</sup> TAKAYUKI YOSHINAKA,<sup>†2</sup>  
TAKEHITO UTSURO,<sup>†1</sup> YASUhide KAWADA<sup>†3</sup>  
and TOMOHIRO FUKUHARA<sup>†4</sup>

Spam blogs or splogs are blogs hosting spam posts, created using machine generated or hijacked content for the sole purpose of hosting advertisements or raising the number of inward of target sites. Among those splogs, this paper focuses on detecting a group of splogs which are estimated to be created by an identical spammer. We especially show that similarities of html structures among those splogs created by an identical spammer contribute to improving the performance of splog detection. In measuring similarities of html structures, we extract a list of blocks (minimum unit of content) from the DOM tree of a html file. We show that the html files of splogs estimated to be created by an identical spammer tend to have similar DOM trees and this tendency is quite effective in splog detection.

### 1. はじめに

ブログには個人の意見情報が記されており、市場の動向を推測するための手掛かりや製品についての意見調査をする上で有益であるとして、近年注目を集めている。そのため、従来からあるインデクシングのみを行う検索エンジンとは異なる、ブログ特有の情報検索サービスが出現している。

具体的には、ブログ解析サービスとして、*Techmorati*, *BlogPulse*<sup>5)</sup>, *kizasi.jp*, *blog-Watcher*<sup>16)</sup> などが存在する。多言語ブログサービスとしては、*Globe of Blogs* が言語横断ブログ記事検索機能を提供している。また *Best Blogs in Asia Directory* がアジア言語ブログの検索機能を提供している。*Blogwise* もまた多言語ブログ記事の分析を行っている。

一方で、ブログのウェブコンテンツの作成と配信は非常に容易になっており、そのことが引き金となって、アフィリエイト収入を得ることを目的とするスパムブログ (以下、スプログ) が急増している<sup>1),6),11),12),14)</sup>。スプログにおいては、通常、広告主への誘導または対象サイトの被リンク数を増加する目的のもとで、機械的な文書作成や他サイトの引用という手段を用いて自動的に記事を生成し、大量のリンクを有するブログを機械的に自動生成する。文献 12) は英語ブログにおいて、約 88% のブログサイトがスプログであり、それは全ブログポストの 75% を占めると報告している<sup>2)</sup>。このことから、文献 1), 13) に述べられているように、スプログは情報検索品質の低下やネットワークと格納資源の多大な浪費などといった問題を起こす要因となる。そのため、近年、スプログの分析や検出を目的とした研究が進められている<sup>11),12),14)</sup>。文献 14) では、TREC Blog06 データコレクションを用いて、スプログのピング時系列特性、入力度数/出力度数の分布特性、典型的な単語群を分析している。また、文献 11), 12) は、*BlogPulse* データセットを用いたスプログ分析の結果を報告している。一方、文献 7), 8), 10), 12), 13), 15) では、スプログを機械的に特定し、排除する技術について報告している。

ここで、本論文では、機械学習を用いたスプログ検出において、スプログの HTML 構造

<sup>†1</sup> 筑波大学大学院システム情報工学研究科 Graduate School of Systems and Information Engineering, University of Tsukuba

<sup>†2</sup> 東京電機大学大学院工学研究科 Graduate School of Engineering, Tokyo Denki University

<sup>†3</sup> (株) ナビックス Navix Co., Ltd.

<sup>†4</sup> 東京大学 人工物工学研究センター Research into Artifacts, Center for Engineering, University of Tokyo

表 1 スブログ/非スブログデータセット

(a) スブログ/非スブログ数					(b) 大量生成型スブログ数				
ブログホスト	CC 社	SS 社	その他	合計	ブログホスト	CC 社		SS 社	
スブログ数	198	293	277	768	ID	1	2	3	4
非スブログ数	210	259	2849	3318	スブログ数	163	26	31	33
合計	408	552	3126	4086					

の類似性を利用する手法を提案する。まず、本論文では、同一スパマーにより作成されているスブログは、HTML 構造が類似していることを示す。HTML 構造の類似度を測るために、HTML 文書の DOM ツリーから DOM 系列を抽出し、DOM 系列の類似度を測定する。そして、同一スパマーにより作成されているスブログ同士は、他のスブログや非スブログと比較すると DOM 系列が類似していることを示す。次に機械学習の一つである Support Vector Machines<sup>21)</sup>(SVMs) を用いた枠組みによって、HTML 構造の類似度の素性を加えることにより、SVM による検出性能が改善する場合があることを示す。

## 2. スブログ/非スブログデータセット

本論文では、2007 年 9 月～2008 年 2 月の期間において収集した日本語スブログ/非スブログデータセットを用いる。日本語スブログ/非スブログデータセットは、文献 17) で提案された基準によって、スブログ/非スブログを判定した結果が付与されている。また、日本語スブログ/非スブログデータセットの中で、テキストコンテンツやブラウザで見た際のフレーム構造が類似しているスブログを、同一作成者が自動生成している「大量生成型」のスブログ<sup>17)</sup>として同定し、大量生成型スパマー ID を付与している。それ以外のスブログを「単発」スブログとした。

本論文の評価のうち、特に大量生成型スブログを対象とした評価においては、スブログ・非スブログデータセット中において、一定数以上の大量生成型スブログを収集済の大量生成型スパマー ID を対象とする。具体的には、表 1 に示すように、ブログホスト CC 社におけるスパマー ID=1 の大量生成型スブログ、および SS 社におけるスパマー ID=2, 3, 4 の大量生成型スブログを対象とする。なお、文献 17) においては、2 社以上のブログホストにまたがって、同一の大量生成型スパマーによって自動生成されたと推測される大量生成型スブログも収集されている。しかし、異なるブログホストのスブログ・非スブログの間では、HTML 文書の構造が大きく異なることが多いため、本論文では、同一の大量生成型スパマーによって自動生成された大量生成型スブログのうち、ブログホストが同一のものに限

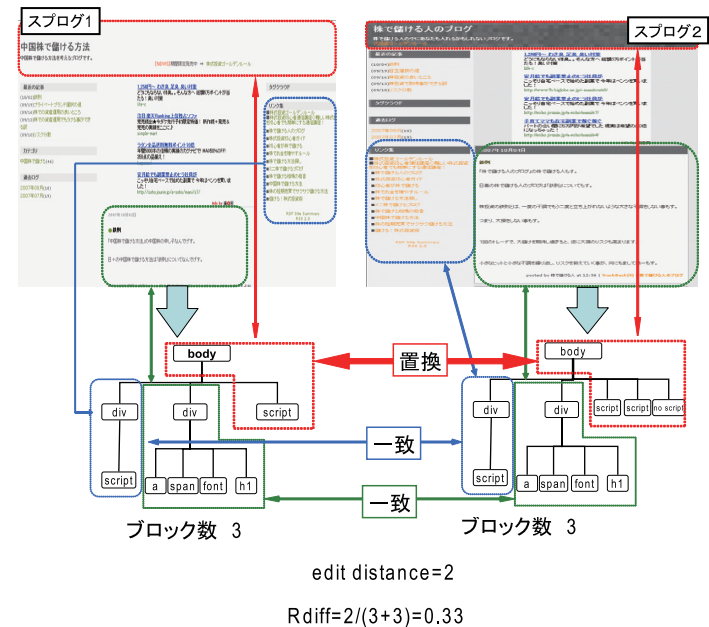


図 1 HTML 文書からの DOM 系列抽出および DOM 系列差分算出の例

定して評価を行う。

## 3. HTML 構造の類似度

### 3.1 HTML ファイルの DOM 系列の抽出

本論文では、文献 23) で提案されたブロック抽出の方式をふまえて、HTML 文書から DOM 系列を抽出する<sup>\*1 \*2</sup>。

\*1 文献 23) も含めて文献 3), 4) 等、HTML 文書からコンテンツを抽出する研究の多くは、自動で主要なコンテンツを抽出する手法に焦点をあてている。その中で、文献 23) の手法は、HTML 構造の差異を測る際に教師データを必要とせず、しかも主要でない補足的なコンテンツ間の差異の測定への適用も比較的容易であることから、本論文の目的に最も適していた。

\*2 スブログ検出において HTML 構造の類似度を使用するという基本的な考え方は、文献 20) においても用いられている。しかし、文献 20) は、HTML 構造の類似度の計算手法のみにとどまっており、類似度を用いたウェブスパムの検出の評価は行っていない。また、我々の手法と比較すると、HTML タグを用いた類似度尺度の粒

まず、図 1 に示すように、HTML 文書  $s$  中の全ての HTML タグを木構造で表現する。次に、この HTML タグの木構造に対して、ブロックレベル要素として用いられるタグのうち、P タグおよび DIV タグによって木構造を分割し、これらのタグの下位にあるタグを取り込むことによって、個々のブロックを構成する。ここで、一般に、ブロックレベル要素としては、P タグおよび DIV タグ以外のタグも用いられるが、本論文では、簡単化のために、P タグおよび DIV タグに限定する。また、文献 23) と同様に、BODY タグも、P タグおよび DIV タグと同様に扱い、BODY タグの位置において、HTML タグの木構造の分割を行う。さらに、文献 23) では、ブラウザにレンダリングされない SCRIPT と STYLE の二タグ及びその下位ノードはブロック内に含まないとしているが、本論文では、ブロックの中身の詳細を区別するために、これらのタグ以下もブロック内に含める。

次に、ブロックにまとめあげられた HTML タグの木構造を横型探索することにより、ブロックのリスト構造を形成し、HTML 文書  $s$  の DOM 系列  $dm(s)$  とする。

3.2 DOM 系列の差分の分布

HTML 文書  $s$  および  $t$  に対して、それぞれから抽出された DOM 系列  $dm(s)$ 、および  $dm(t)$  の差分を DP マッチングによって求める。DP マッチングの際、挿入および削除のコストを 1、置換のコストを 2 として、DP マッチングにより求まる編集距離 (レーベンシュタイン距離) を  $edit\ distance\ (dm(s), dm(t))$  とする。次に、抽出された DOM 系列  $dm(s)$  の要素数を  $|dm(s)|$  とし、以下の式で  $s, t$  の DOM 系列の差分の割合  $Rdiff(s, t)$  を計算する。

$$Rdiff(s, t) = \frac{edit\ distance\ (dm(s), dm(t))}{|dm(s)| + |dm(t)|}$$

図 1 に、二つのプログラムの HTML 文書から DOM 系列を求めた後、差分の割合を模式的に算出する様子を示す。

3.3 DOM 系列の差分の割合の分布

次に、プログラムもしくは非プログラムの HTML 文書の集合  $S$  および  $T$  の間で、HTML 文書  $s \in S$  および  $t \in T$  の間の DOM 系列の差分の割合を求め、その分布を分析する。具体的には、大量生成型のプログラム同士、大量生成型のプログラムと単発プログラムの間、大量生成型のプログラムと非プログラムの間、および、単発プログラム・非プログラム同士の間で DOM 系列の差分の割合の分布を比較する。そのためにまず、HTML 文書  $s \in S$  に対して、HTML 文書集合  $T$  の要素  $t \in T$  との間で、差分の割合  $Rdiff(s, t)$  が最も小さい 10 個を求め、そ

度が相対的に粗いと言える。

の差分の割合の平均値を  $AvMinDF_{10}(s, T)$  とする\*1。

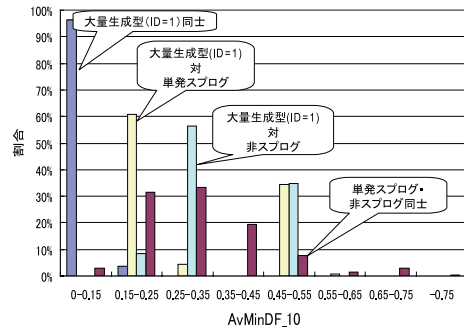
$$AvMinDF_{10}(s, T) = T \text{ 中で, } Rdiff(s, t \in T) \text{ の値が最も小さい } 10 \text{ 個の } t \text{ に対する } Rdiff(s, t) \text{ の平均値}$$

次に、大量生成型プログラム (ID=1, CC 社)、および、大量生成型プログラム (ID=2, 3, 4, SS 社) を対象として、大量生成型のプログラム同士、大量生成型のプログラムと単発プログラムの間、大量生成型のプログラムと非プログラムの間、および、単発プログラム・非プログラム同士の間で DOM 系列の差分の割合の分布を求め、これを図 2 において比較する。ここで、大量生成型のプログラム同士の場合には、同一の大量生成型プログラムの ID (例えば、ID=1) を持つ大量生成型プログラムの集合を  $S$  および  $T$  として、 $S$  の要素  $s$  に対して  $AvMinDF_{10}(s, T)$  の分布を求める。一方、大量生成型のプログラムと単発プログラムの間の場合には、例えば、大量生成型プログラム (ID=1, CC 社) を対象とする場合には、大量生成型プログラムの ID=1 を持つ大量生成型プログラムの集合を  $S$ 、プログラムホスト CC 社における全単発プログラムの集合を  $T$  として、 $S$  の要素  $s$  に対して  $AvMinDF_{10}(s, T)$  の分布を求める。大量生成型のプログラムと非プログラムの間の場合も同様である。一方、単発プログラム・非プログラム同士の間の場合には、プログラムホスト SC 社およびプログラムホスト SS 社のそれぞれについて分布を求める。例えば、図 2(a) の場合は、プログラムホスト CC 社中の単発プログラム・非プログラムの和集合を  $S$  および  $T$  として、 $S$  の要素  $s$  に対して  $AvMinDF_{10}(s, T)$  の分布を求める。図 2(b), (c), (d) の場合は、プログラムホスト SS 社中の単発プログラム・非プログラムを対象として同様の分布を求める (図 2(b), (c), (d) とも全く同一の分布を示している。) 図 2(a),(c),(d) のいずれの分布を見ても、大量生成型のプログラム同士の分布に対して、それ以外の、大量生成型のプログラムと単発プログラムの間の分布、大量生成型のプログラムと非プログラムの間の分布、および、単発プログラム・非プログラム同士の分布がほぼ分離される傾向にあることが分かる。例えば、図 2(a) では、大量生成型のプログラム同士では、ほぼ全ての大量生成型に対する  $AvMinDF_{10}(s, T)$  の値が 0 から 0.15 の範囲に分布しており、それ以外の三種類の分布とはほぼ分離される。

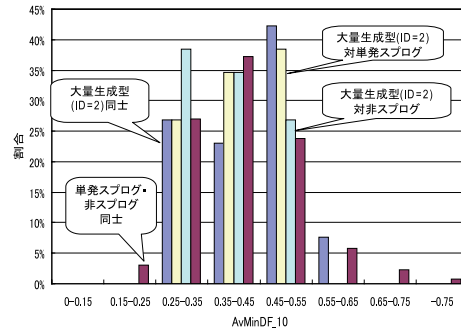
また、図 2(b) の ID=2 の大量生成型プログラムにおいて、大量生成型のプログラムは、他のプログラムサイトに比べて類似しているとはいえない。これらにおいては、プログラムの本文部分は極めて類似しているが、サイドバー部分が大きく違うことにより、 $AvMinDF_{10}(s, T)$  が大きくなっていった。

以上の結果から、同一の大量生成型プログラムの ID を持つ大量生成型プログラムの HTML 文

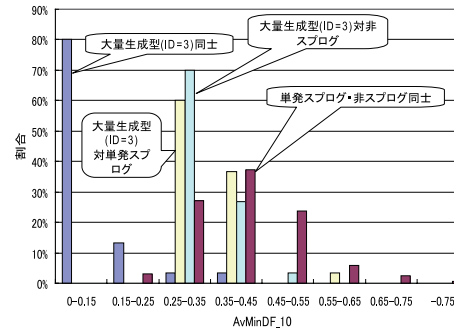
\*1  $AvMinDF_k(s, T)$  について、 $k = 1, \dots, 25$  の性能を比較した結果、 $k = 10$  の場合が最も高い性能を示した。



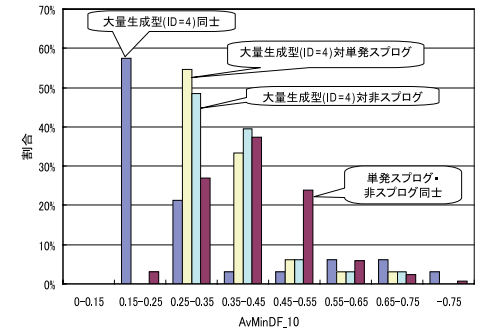
(a) 大量生成型スプログ (ID = 1, CC 社)



(b) 大量生成型スプログ (ID = 2, SS 社)



(c) 大量生成型スプログ (ID = 3, SS 社)



(d) 大量生成型スプログ (ID = 4, SS 社)

図 2 スプログ・非スプログの DOM 系列の差分の割合の分布

書から抽出した DOM 系列は、相互に高い類似性を持つものが多く、この類似性を用いることにより、同一スプログホストにおける単発スプログや非スプログとの識別において利用できる可能性があることがわかった。

#### 4. スプログ検出のための素性

本節では SVM によるスプログ判定において用いる素性について述べる。

##### 4.1 DOM 系列の差分素性

本論文では、3 節で述べた手法により、HTML 文書から DOM 系列を抽出し、それらの差分の割合を求め、素性値とする。以下では、訓練・評価事例となるスプログまたは非スプログを  $s$  とし、3.3 節の手順により  $AvMinDF_{10}(s, T)$  を求めた後、その対数  $\log AvMinDF_{10}(s, T)$  を素性値とする<sup>\*1</sup>。この手順において、差分を求める対象の集合  $T$  を二通り用意することにより、以下の二種類の素性を設定する。

- (1) 集合  $T$  を、訓練事例中の全スプログの集合とする。この素性を、DOM 系列差分 (スプログ) 素性とよぶ。
- (2) 集合  $T$  を、訓練事例中の全大量生成型スプログの集合とする。この素性を、DOM 系

\*1 対数なしの場合、とり得る値の範囲が大きく、値のばらつきが大きくなるので有効に機能しないと考え対数をとることにした。

列差分 (大量生成型) 素性とよぶ。

#### 4.2 従来からの素性

文献 9) で使用した素性を従来素性とする。

##### 4.2.1 ブラックリスト/ホワイトリスト URL 素性

訓練事例として、スプログ/非スプログが与えられると、その HTML ファイルからアウトリンクとなっている URL を抽出する。その中から以下の条件を満たすものを選定し、ホワイトリスト URL とした。

- i) 訓練事例中のスプログの HTML ファイルのいずれにも含まれない URL である。
  - ii) 訓練事例中の非スプログの HTML ファイルの中で、2 回以上出現する URL である。
- 次に、各ホワイトリスト URL  $u$  に以下のように重みづけを行い、ホワイトリスト URL 素性の値を算出した。

$$\log \sum_u \left( \frac{\text{訓練事例全体中の非スプログにおける } u \text{ の総出現頻度}}{\text{訓練事例全体中の非スプログにおける } u \text{ の総出現頻度}} \right) \times \left( \frac{\text{評価事例における } u \text{ の出現頻度}}{\text{評価事例における } u \text{ の出現頻度}} \right)$$

一方、ブラックリスト URL についても、同様の手順で選定した。

##### 4.2.2 名詞句素性

文献 17), 22) の知見より、スプログおよび非スプログ中における単語の分布には異なりがあり、特定の種類の単語は非スプログよりもスプログに現れやすいということがわかって

いる．そこで，特定の名詞句とスプログ，非スプログとの間の相関をとらえるために，名詞句素性を導入する．

具体的には，スプログ/非スプロクの本文テキストを形態素解析<sup>\*1</sup>した結果から名詞句を抽出し，以下の分割表にしたがって，訓練データ中のスプログ/非スプログにおける名詞句  $w$  の出現頻度を用いて，スプログと名詞句  $w$  との間の  $\phi^2$  統計量を求めた．

	$w$	$\neg w$
訓練データ中のスプログ	$freq(\text{スプログ}, w) = a$	$freq(\text{スプログ}, \neg w) = b$
訓練データ中の非スプログ	$freq(\text{非スプログ}, w) = c$	$freq(\text{非スプログ}, \neg w) = d$

$$\phi^2(\text{スプログ}, w) = \frac{(ad - bc)^2}{(a + b)(a + c)(b + d)(c + d)}$$

また，評価事例に対しては，この名詞句素性の値として以下の式を用いた．

$$\log \sum_w \phi^2(\text{スプログ}, w) \times (\text{評価事例における } w \text{ の出現頻度})$$

#### 4.2.3 アンカーテキスト名詞句・リンク URL 素性

ブラックリスト/ホワイトリスト URL 素性および名詞句素性よりもより詳細な条件を設定することにより，より有効な性能を示す素性として，アンカーテキストの名詞句およびそのリンク先 URL の (緩い) 組み合わせを用いる．以下，まず，名詞句  $w$  およびブログサイト  $s$  に対して，以下の尺度  $AncfB(w, s)$  および  $AncfW(w, s)$  を定義する．

$$AncfB(w, s) = \begin{pmatrix} \text{ブログサイト } s \text{ 中で名詞句 } w \text{ がアンカーテキストに含まれ} \\ \text{そのリンク先がブラックリスト URL もしくは訓練事例中の} \\ \text{スプログとなっている回数} \end{pmatrix}$$

$$AncfW(w, s) = \begin{pmatrix} \text{ブログサイト } s \text{ 中で名詞句 } w \text{ がアンカーテキストに含まれ} \\ \text{そのリンク先がホワイトリスト URL もしくは訓練事例中の} \\ \text{非スプログとなっている回数} \end{pmatrix}$$

そして，訓練事例中のスプログ全体の中での総出現頻度  $\sum AncfB(w, s)$  が 2 以上であるものを選定し，「ブラックリスト URL へのアウトリンクを持つスプログアンカーテキスト名詞句」とする．次に， $w$  を「ブラックリスト URL へのアウトリンクを持つスプログ

アンカーテキスト名詞句」として，評価事例  $t$  に対して以下の重みを算出し，評価事例  $t$  に対する「ブラックリスト URL へのアウトリンクを持つアンカーテキスト名詞句素性」の値とする．

$$\log \sum_w \left( \sum_{\text{訓練事例中のスプログ } s} AncfB(w, s) \right) \times AncfB(w, t)$$

同様に，訓練事例中のスプログ全体の中での総出現頻度  $\sum AncfW(w, s)$  が 2 以上であるものを選定し，「ホワイトリスト URL へのアウトリンクを持つスプログアンカーテキスト名詞句」とする．次に， $w$  を「ホワイトリスト URL へのアウトリンクを持つスプログアンカーテキスト名詞句」として，評価事例  $t$  に対する以下の重みを，評価事例  $t$  に対する「ホワイトリスト URL へのアウトリンクを持つアンカーテキスト名詞句素性」の値とする．

$$\log \sum_w \left( \sum_{\text{訓練事例中のスプログ } s} AncfW(w, s) \right) \times AncfW(w, t)$$

## 5. スプログ検出および信頼度尺度

### 5.1 SVM を用いたスプログ検出

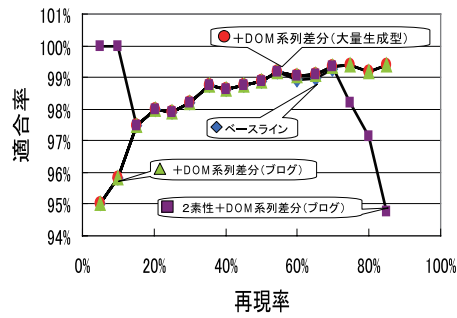
SVM 機械学習を行うためのツールとして，TinySVM (<http://chasen.org/~taku/software/TinySVM/>) を用いた．カーネル関数としては，線形および 2 次多項式を比較し，2 次多項式の方が性能が良かったため，6 節においては，2 次多項式カーネルを用いた場合の結果を示す．また，全ての素性に値がないものは訓練データから除外する．

### 5.2 信頼度尺度

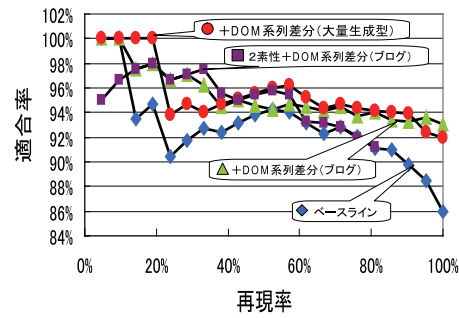
SVM 機械学習での信頼度尺度として，分離平面から各評価事例への距離を用いた<sup>19)\*2</sup>．具体的には，スプログとして判定される事例に対する分離平面からの距離の下限  $LBD_s$ ，および，非スプログとして判定される事例に対する分離平面からの距離の下限  $LBD_{ab}$  をそれぞれ設定する．

\*1 日本語形態素解析器 茶筌 (<http://chasen-legacy.sourceforge.jp/>) および ipadic 辞書を用いた．

\*2 機械学習および統計的自然言語処理の分野における能動学習手法の研究事例においては，未知事例のうちで信頼度の低い事例を選別して訓練事例に追加する過程において，信頼度尺度が利用される．

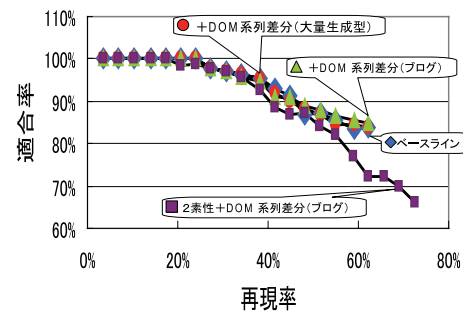


(a-1) スプログ検出性能 (CC 社)

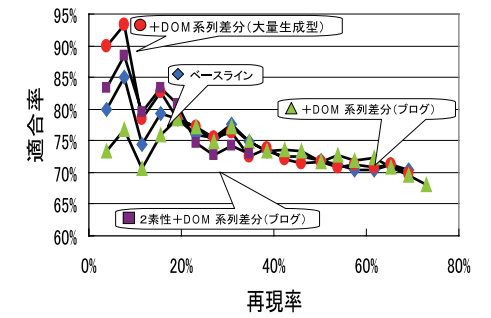


(a-2) 非スプログ検出性能 (CC 社)

CC 社



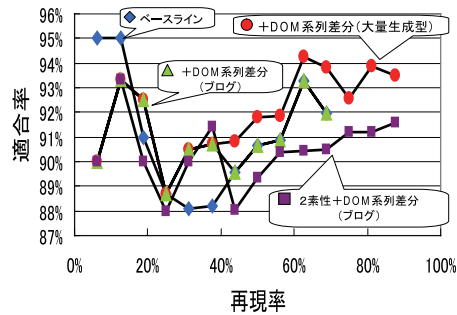
(b-1) スプログ検出性能 (SS 社)



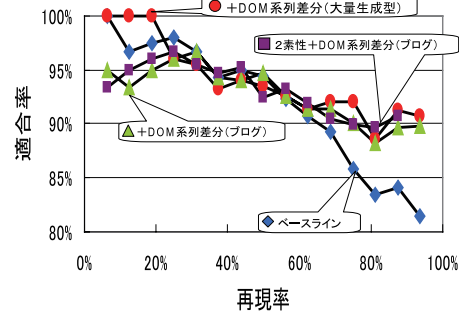
(b-2) 非スプログ検出性能 (SS 社)

SS 社

図 3 スプログ/非スプログ検出性能

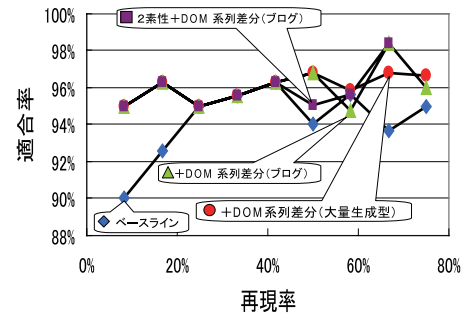


(a-1) 大量生成型スプログ検出性能 (CC 社)

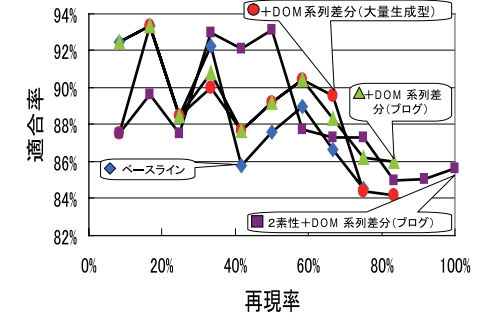


(a-2) 単発スプログ・非スプログ検出性能 (CC 社)

CC 社



(b-1) 大量生成型スプログ検出性能 (SS 社)



(b-2) 単発スプログ・非スプログ検出性能 (SS 社)

SS 社

図 4 「大量生成型スプログ」/「単発スプログ・非スプログ」検出性能

## 6. 評価

### 6.1 評価手順

本節では、表 1(a) に示すデータセットのうち、CC 社および SS 社のスプログ・非スプログデータセットを用いて、プログホスト別に、スプログ/非スプログ検出性能 (図 3)、および「大量生成型スプログ」/「単発スプログ・非スプログ」検出性能 (図 4) の評価を行った。スプログ/非スプログ検出性能の評価においては、CC 社を対象とした評価では全 408 プログサイトを、SS 社を対象とした評価では全 552 プログサイトを、それぞれ用いた。「大量生成型スプログ」/「単発スプログ・非スプログ」検出性能の評価においては、CC 社の場合は、大量生成型スプログ 163 サイトおよび単発スプログ・非スプログ計 163 サイトを用い、SS 社の場合は、大量生成型スプログ 90 サイトおよび単発スプログ・非スプログ計 90 サイトを用いた。評価においては、これらのデータセットを用いて、10 分割交差検定を行った。

### 6.2 評価尺度

以下の説明においては、スプログ/非スプログ検出性能の評価においては、スプログを正例、非スプログを負例とする。また「大量生成型スプログ」/「単発スプログ・非スプログ」検出性能の評価においては、大量生成型スプログを正例、単発スプログ・非スプログを負例とする。

以下では、評価用事例集合、および、正例として判定される事例に対する分離平面からの距離の下限  $LBD_p$  を用いる。分離平面からの距離が  $LBD_p$  以上となる評価事例に対して、正例として判定した場合の再現率、適合率を測定する。そして、 $LBD_p$  を変化させた場合の再現率、適合率の推移をプロットする。同様に、評価用事例集合、および、負例として判定される事例に対する分離平面からの距離の下限  $LBD_n$  を用いて、分離平面からの距離が  $LBD_n$  以上となる評価事例に対して、負例として判定した場合の再現率、適合率を測定する。そして、 $LBD_n$  を変化させた場合の再現率、適合率の推移をプロットする。

### 6.3 評価結果

図 3 および図 4 において、「ベースライン」のプロットは、従来からの素性だけを用いて訓練した分類器の性能を示す。「+DOM 系列差分 (大量生成型)」のプロットは、従来からの素性に DOM 系列差分 (大量生成型) 素性を追加して訓練した分類器の性能を示し、「+DOM 系列差分 (プログ)」のプロットは、従来からの素性に DOM 系列差分 (プログ) 素性を追加して訓練した分類器の性能を示す。図 3 および図 4 の (a-1)、(a-2) において、「2 素性 + DOM 系列差分 (プログ)」のプロットは、ホワイトリスト URL 素性、名詞句素性、DOM

系列差分 (プログ) 素性を用いて訓練した分類器の性能である。一方、図 3 および図 4 の (b-1)、(b-2) において、「2 素性 + DOM 系列差分 (プログ)」のプロットは、ホワイトリスト URL 素性、ブラックリスト URL へのアウトリンクを持つアンカーテキスト名詞句素性、DOM 系列差分 (プログ) 素性を用いて訓練した分類器の性能である。

図 3 において、(a-2) の CC 社非スプログ検出性能では、DOM 系列差分 (プログ) 素性および DOM 系列差分 (大量生成型) 素性を用いた場合に、高再現率点においてベースラインとの適合率の差は統計的に有意であった。しかし、(a-1)、(b-1) および (b-2) では、DOM 系列差分 (プログ) 素性および DOM 系列差分 (大量生成型) 素性を加えても、ベースラインとの適合率の差は統計的に有意ではなかった。これにより、HTML 構造の素性を加えてもスプログ/非スプログ検出性能が必ず改善されるとは言えないことが分かった。ただし、図 3 において、DOM 系列差分 (プログ) 素性および DOM 系列差分 (大量生成型) 素性を用いた場合、ベースラインと比べてサポートベクター数が減少していることが分かった。少ないサポートベクター数で同等の性能を達成できていることから、タスクのモデル化においてより適切な素性が導入できたことが示唆される。今後は、データ量を増やすことにより、ベースラインに対して統計的に有意な改善が達成できるかどうかの検証を進める必要がある。

一方、図 4 においては、(a-2) の SS 社「単発スプログ・非スプログ」検出性能では、DOM 系列差分 (大量生成型) 素性を用いた場合のみ、高再現率点におけるベースラインとの適合率の差は統計的に有意であった。しかし、(a-1)、(b-1) および (b-2) では、DOM 系列差分 (プログ) および DOM 系列差分 (大量生成型) 素性を用いても、ベースラインとの適合率の差は統計的に有意ではなかった。これにより、HTML 構造の素性を加えても「大量生成型スプログ」検出性能および「単発スプログ・非スプログ」検出性能も必ず改善されるとは言えないことが分かった。しかし、図 4 において、DOM 系列差分 (プログ) 素性および DOM 系列差分 (大量生成型) 素性を用いた場合、ベースラインと比べてサポートベクター数が減少していることが分かった。この場合も、少ないサポートベクター数で同等の性能を達成できていることから、タスクのモデル化においてより適切な素性が導入できたことが示唆される。また、前述と同様に、データ量を増やすことにより、ベースラインに対して統計的に有意な改善が達成できるかどうかの検証を進める。

また、図 3 および図 4 を通して、「DOM 系列差分 (プログ)」は「2 素性 + DOM 系列差分 (プログ)」との適合率の差が統計的に有意であったものはほとんどなかった。この結果から、DOM 系列差分 (プログ) 素性を用いれば、それ以外の従来からの素性を全て用いなくとも、従来からの素性 2 種類を適切に選定さえすれば十分であることが分かる。

## 7. おわりに

本論文では、同一のスプログ作成者が自動的に大量生成したと推測されるスプログの検出において、HTML 構造の類似性が効果的であることを示した。機械学習のひとつである SVM を用いた枠組みによって、スプログの判定を行うタスクを設定し、HTML 構造の類似性を素性として、SVM を用いたスプログ検出を行った結果において、スプログ検出の性能が向上する場合があることを示した。今後は、本論文の DOM 系列差分の定式化において、DOM 系列の余弦尺度<sup>24)</sup>、木構造カーネルを用いる方式<sup>18)</sup>等、他の定式化との比較を行う。

大量生成したと推測されるスプログの HTML 構造は類似性があるという特徴を用いれば、訓練事例となるスプログを用意しなくても、DOM 系列の差分の割合が極端に小さいプログの組を自動収集することにより、教師なしスプログ収集を実現できる可能性がある。現在、この考え方に基づいて、数百万件のブログをクロールした結果から、スプログの候補を収集し、人手によるスプログ・非スプログ判定作業を進めており、DOM 系列の差分の割合が極端に小さいプログの組の中にスプログが含まれることを確認済みである。この結果の詳細については、別の機会に報告する予定である。

## 参 考 文 献

- 1) : *Wikipedia, Spam blog*. [http://en.wikipedia.org/wiki/Spam\\_blog](http://en.wikipedia.org/wiki/Spam_blog).
- 2) : *Wikipedia, Ping (blogging)*. [http://en.wikipedia.org/wiki/Ping\\_\(blogging\)](http://en.wikipedia.org/wiki/Ping_(blogging)).
- 3) Bing, L., Wang, Y., Zhang, Y. and Wang, H.: Primary Content Extraction with Mountain Model, *Proc. 8th IEEE CIT*, pp.479–484 (2008).
- 4) Debnath, S., Mitra, P., Pal, N. and Giles, C.L.: Automatic Identification of Informative Sections of Web Pages, *IEEE Transactions on Knowledge and Data Engineering*, Vol.17, No.9, pp.1233–1246 (2005).
- 5) Glance, N., Hurst, M. and Tomokiyo, T.: BlogPulse: Automated Trend Discovery for Weblogs, *Proc. WWW 2004 Workshop on the Weblogging Ecosystem: Aggregation, Analysis and Dynamics* (2004).
- 6) Gyöngyi, Z. and Garcia-Molina, H.: Web Spam Taxonomy, *Proc. 1st AIRWeb*, pp. 39–47 (2005).
- 7) 石田和成：スパムブログの推定と抽出，日本データベース学会 Letters，Vol.6, No.4, pp.37–40 (2008).
- 8) 石田和成：共起クラスターシードと連鎖的抽出にもとづくスパムブログのフィルタリング，Web とデータベースに関するフォーラム (WebDB Forum)2008 論文集，情報処理学会 (2008).
- 9) Katayama, T., Sato, Y., Utsuro, T., Yoshinaka, T., Kawada, Y. and Fukuhara, T.: An Empirical Study on Selective Sampling in Active Learning for Splog Detection, *Proc. 5th AIRWeb*, pp.29–36 (2009).
- 10) Kolari, P., Finin, T. and Joshi, A.: SVMs for the Blogosphere: Blog Identification and Splog Detection, *Proceedings of the 2006 AAAI Spring Symposium on Computational Approaches to Analyzing Weblogs*, pp.92–99 (2006).
- 11) Kolari, P., Finin, T. and Joshi, A.: Spam in Blogs and Social Media, *Tutorial at ICWSM* (2007).
- 12) Kolari, P., Joshi, A. and Finin, T.: Characterizing the Splogosphere, *Proc. 3rd Ann. Workshop on the Weblogging Ecosystem: Aggregation, Analysis and Dynamics* (2006).
- 13) Lin, Y.-R., Sundaram, H., Chi, Y., Tatemura, J. and Tseng, B.L.: Splog Detection using Self-similarity Analysis on Blog Temporal Dynamics, *Proc. 3rd AIRWeb*, pp. 1–8 (2007).
- 14) Macdonald, C. and Ounis, I.: The TREC Blogs06 Collection : Creating and Analysing a Blog Test Collection, Technical Report TR-2006-224, University of Glasgow, Department of Computing Science (2006).
- 15) Mishne, G., Carmel, D. and Lempel, R.: Blocking Blog Spam with Language Model Disagreement, *Proc. 1st AIRWeb* (2005).
- 16) Nanno, T., Fujiki, T., Suzuki, Y. and Okumura, M.: Automatically Collecting, Monitoring, and Mining Japanese Weblogs, *WWW Alt. '04: Proc. 13th WWW Conf. Alternate Track Papers & Posters*, pp.320–321 (2004).
- 17) Sato, Y., Utsuro, T., Fukuhara, T., Kawada, Y., Murakami, Y., Nakagawa, H. and Kando, N.: Analysing Features of Japanese Splogs and Characteristics of Keywords, *Proc. 4th AIRWeb*, pp.33–40 (2008).
- 18) Suzuki, J., Isozaki, H. and Maeda, E.: Convolution Kernels with Feature Selection for Natural Language Processing, *Proc. 42nd ACL*, pp.110–126 (2004).
- 19) Tong, S. and Koller, D.: Support Vector Machine Active Learning with Applications to Text Classification, *Proc. 17th ICML*, pp.999–1006 (2000).
- 20) Urvoy, T., Lavergne, T. and Filoche, P.: Tracking Web Spam with Hidden Style Similarity, *Proc. 2nd AIRWeb*, pp.25–30 (2006).
- 21) Vapnik, V.N.: *Statistical Learning Theory*, Wiley-Interscience (1998).
- 22) Wang, Y., Ma, M., Niu, Y. and Chen, H.: Spam Double-Funnel: Connecting Web Spammers with Advertisers, *Proc. 16th WWW*, pp.291–300 (2007).
- 23) 吉田光男，山本幹雄：教師情報を必要としないニュースページ群からのコンテンツ自動抽出，日本データベース学会論文誌，Vol.8, No.1, pp.29–34 (2009).
- 24) 余 耕イ：HTML タグ系列の類似性を用いたスパムブログの自動検出，修士論文，東京工業大学 大学院総合理工学研究科 知能システム科学専攻 (2009).