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

2004－ＦＩ－ 76　(3)
2004－ＮＬ－163　(3)
2004／9／16

部分文字列の頻度分布に基づいた同一テンプレートを持つ Web ページの自動収集

池田大輔[†]　山田泰寛[††]
田中省作[†††]　松本英樹[††]

クローラー等が収集した大量の Web ページから、テンプレートを共有する Web ページ群を発見する**データ発見問題**について考察する。各テンプレートから生成されたページ群は情報抽出やラッパー生成アルゴリズムの入力として利用できる。本稿では、この問題に対し**部分文字列増幅法**と呼ばれる線形時間アルゴリズムを利用し、実データを用いた実験により有効性を示す。この手法はコンテンツを記述する言語の頻度分布を利用するが、実際に 9ヶ国の言語に対し分布を調べ、言語非依存性も示す。さらに、ノイズが混入する場合はノイズを分離する境界値の設定が必要だが、自然言語に普遍的な特徴を用いることで、設定の一部を自動化する方法を提案する。

Collecting Web Pages with Templates using Distributions of Substring Frequencies

Daisuke Ikeda [†], Yasuhiro Yamada [††], Shosaku Tanaka [†††]
and Hideki Matsumoto[††]

We consider the *data discovery problem* to find sets of Web pages, each of which share some template, among many pages collected by a Web crawler. A found set is a potential input for information extraction and wrapper generation algorithms. We propose a linear time algorithm, called the *substring amplification*, and show its effectiveness by experiments using real data. The algorithm exploits distributions of substring frequencies for natural languages, which describe contents. Therefore, we examine distributions for 9 languages, and show its independence from natural languages. The algorithm requires a threshold value if noise files abound. We discuss how to decide an appropriate value for it using properties of natural languages.

1. はじめに

Web 上には種々雑多なデータが膨大に混在しており、この中から求める情報や有用な知識を効率よく発見する手法が求められる。混沌とした Web 空間だが、特定のサイトやデータの製作者が作成したデータ内に限れば、共通する構造が存在する場合も多い。例えば、検索エンジンは検索結果が繰り返し並んだリストにして返す。新聞記事やプレスリリースのページは、個々のページ内には共通の構造はないが、ページ間では共通のテンプレートが利用されている。

これらのデータは、物理的なレイアウトにより人間にとって意味のある属性を付与しているだけであり、実際のデータは HTML ファイルである。このことは、得られる HTML ファイルが構造化されたデータベースから生成される場合も同様である。このようなデータを構造化し、データの検索や加工を容易にする研究が行なわれている[1)~5),7),11)~13)]。これらの研究では、基本的に対象とするデータは同じテンプレートから生成されたものと仮定しているが、これらのデータをどのように発見・収集するかについて言及したものはほとんど存在しない。しかし、Web 上のデータを対象としているため、データベースを対象とするデータマイニング等と異なり、まずデータを収集する必要がある。

本稿では、ラッパー生成や情報抽出アルゴリズムの入力となる Web ページ群を発見する問題を**データ発見問題**と呼び、これを解くアルゴリズムを提案する。これは、ロボットやクローラー等により収集された大量の HTML ファイルを入力として与えられた時、同一のテンプレートから生成されたファイルごとにクラスタリングする問題である。各クラスタ内のファイルが、ラッパー生成や情報抽出アルゴリズムの入力となる。

データ発見問題が対象とするデータは膨大であり、クラスタの数も不明である。また、クラスタには分類されないデータも多く含まれると考えられる。そのため、データ発見問題を解くアルゴリズムは高速で規模耐性を持ち、かつ、ノイズ耐性も有する必要がある。さらに、データ発見問題では、事前に使えるデータが存在しないため、

† 九州大学附属図書館
Kyushu University Library
†† 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
††† 九州大学情報基盤センター
Computing and Communications Center, Kyushu University

訓練例や適切な閾値等の設定は困難である。そのため、訓練例等が不要でも動くアルゴリズムであることが望ましい。

本稿では、**部分文字列増幅法**と呼ぶアルゴリズムを利用し、データ発見問題を解く。このアルゴリズムは、本来他のラッパー生成や情報抽出アルゴリズムと同じく、すでにクラスタリングされていて、同一テンプレートから生成されたファイルが与えらるという仮定のもとに、頻度の高い部分を共通するテンプレートとして特定するものであった [8),15),16)]。テンプレートは、文法やラッパー等のように厳密な形式ではなく、より表現力の緩い部分文字列の集合として定義される。

部分文字列増幅法は入力長 n に対し $O(n)$ 時間で動く高速なアルゴリズムである。実際の入力は、1 本の文字列ではなく、ファイル（文字列）の集合として与えられるが、部分文字列増幅法はファイルが 1 つ与えられるごとに計算を進めることができ、その意味でオンラインである。さらに、収集したファイルを与えるだけでよく、あらかじめ訓練例やクラスタ数を指定する必要がない。

部分文字列増幅法では、各ファイルはテンプレートとコンテンツから構成されると仮定する。その上で、テンプレートと主に自然言語で記述されるコンテンツ内の部分文字列の生起確率が異なることを利用し、テンプレートの分離を行なう。具体的には、f 回出現する部分文字列の総出現数 $F(f)$ を計測し、この値が特異的に大きな f_p を探し、f_p 回現われる部分文字列を探すべきテンプレートの一部とする。つまり、自然な分布であるコンテンツの分布と比較して、規則的に現われるテンプレートの出現頻度は特異的であり、この頻度の差を増幅することでテンプレートの発見を容易にしている（図 1 参照）。

そのため、無規則という意味ではこちらも自然な分布であるノイズファイルが混入しても問題なく、さらに、複数のテンプレートから生成された場合でも問題なくテンプレートを発見できる [9)]。部分増幅法のこのような性質から、テンプレート発見だけではなく、データ発見問題にも適用可能であると考える。ラッパー生成アルゴリズムをクラスタリングに利用するという点では文献 6) の手法が近いが、あらかじめクラスタリングするためのラッパーが分かっていないといけない点が、部分文字列増幅法とは根本的に異なる。

テンプレートは規則的に現われてさえいればよいので、テンプレート部分を構成するマークアップ言語は HTML でなくてもよく、自然言語が混在していても問題ない。しかし、コンテンツ部分を記述する言語の頻度分布は自然である必要がある☆。文献 14) では、英語と日本語の場合に限り実験を行なっているが、任意の自然言語を扱えるかどうかは未確認であった。通常、クローラー等はリンクをたどりながらファイルを収集するため、どのような言語のコンテンツが収集されるか未知である。そのため、コンテンツを記述する言語は仮定できず、この点を明確にすることは重要な問題である。

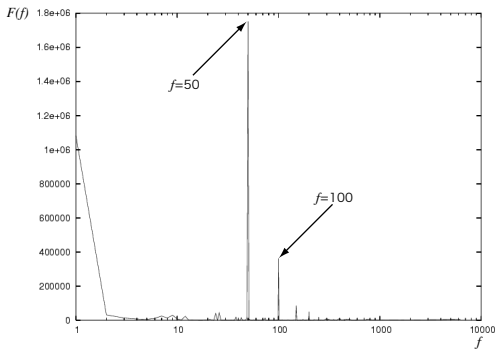

図 1 産経新聞の記事ファイル 50 個から生成した $F(f)$ のグラフ。横軸は頻度 f で、縦軸は総出現数 $F(f)$。特異的な値を持つ f によりテンプレート内の部分文字列が特定される

本稿では、チェコ語、ドイツ語、オランダ語、フランス語、ドイツ語、スペイン語、トルコ語、イタリア語、日本語の 9 言語約 17 万強の文書について $F(f)$ を計測し、部分文字列の頻度分布について調べ、言語非依存性を示す。

データ発見問題では、ノイズが多いため、どこまでを発見すべきクラスタとし、どこからをノイズとするかの境界値が必要である。そこで、最適な境界値を自動で定めるための試みの一つとして、自然言語に普遍的な性質を用いた手法について考察する。

2. 部分文字列増幅法

本節では、文献 8), 10) に従って部分文字列増幅法について簡単に説明する。

$\mathcal{D}$ を文書集合とする。$V(f)$ と $F(f)$ を、それぞれ、$\mathcal{D}$ 中にちょうど f 回出現する部分文字列の異なり数と総出現数とする。つまり、$F(f) = f \times V(f)$ である。

部分文字列増幅法の基本的なアイデアを、具体的な例を用いて説明する。図 1 は、産経新聞☆☆から取得した 50 個の記事ファイルを $\mathcal{D}$ とした時の $F(f)$ のグラフである。各ファイルは基本的に日付、見出し、サブ見出し、記事本文の 4 種類のコンテンツから構成されるが、サブ見出しの数はファイルにより 0〜3 個までと様々であった。各ファイルは空白文字を含め取得したままで $F(f)$ を計測する（図 2 参照）。

図 1 において、f が増加するにつれ $F(f)$ は急激に小さくなっているが、その中に特異的に大きな $F(f)$ を持つ箇所がある。まず、$f = 50$ に特異的に高い $F(f)$ が存在することが分かる。この数はファイル数と同じであり、あ

☆ 「自然である」ための必要十分条件の特定は未解決だが、例えば Zipf の法則で知られるように、ベキ分布に従う場合は問題ないことが分かっている。

☆☆ http://www.sankei.co.jp/

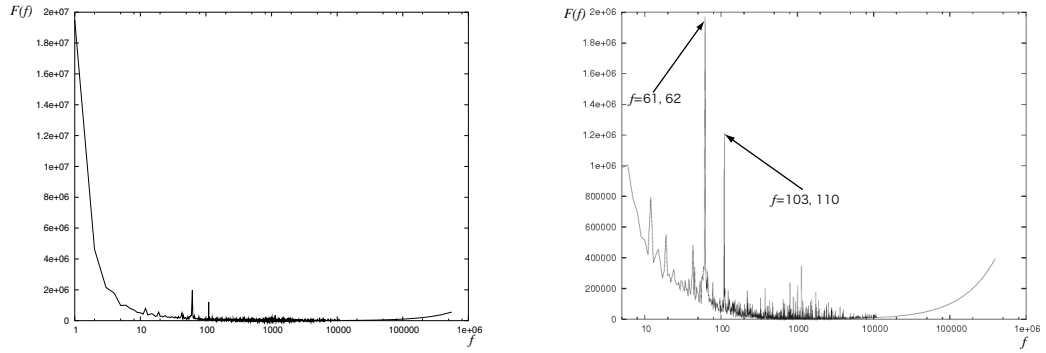

図 3 九州大学から収集したファイルから生成した $F(f)$ のグラフ。左が $f \geq 1$ で、右は $f \geq 5$ に拡大したもの

```
</b></i><br><P>
<center>
<table width=467>
<tr>
<td>

<!---------ヘッダ情報終了---------->

<!------------★★ここから入れ替えてね ‥ ---------------->
<font color="#8b0000">■</font><b>
```

図 2 コメントや空白記号を含んだ産経新聞のテンプレートの一部

る程度長い部分文字列がちょうど 50 回出現していることを示している。$F(f)$ は、全ての部分文字列の出現回数を足しあわせて得られるので、何度も繰り返し出現する文字列から生成される $F(50)$ は非常に大きい値となる。そのため、この手法を部分文字列**増幅**法と呼ぶ。

$\mathcal{D}$ 中にちょうど 50 回出現する部分文字列をつなげて得られる文字列は多くの部分文字列を含んでいる。例えば、図 2 の文字列は 72 文字であり、テンプレートの一部であった。この中に含まれる部分文字列の総出現数は $72 \cdot (72-1)/2 = 2556$ となり、また、すべての記事がこの文字列を含んでいることから少なくとも $F(50)$ は 2556×50 となることが分かる。さらに、$f = c \times 50\ (c = 2, 3, \ldots)$ にも高い値が見える。これらは、50 回出現するテンプレートの中で 2 度、3 度と出現する部分文字列による。

以上より、$F(f)$ が特異的に大きな値を持つ f の値が分かれば、ちょうど f 回出現する部分文字列からテンプレートが見つけられることが分かる。そこで、例えば、$G(f)$ を

$$G(f) = \frac{F(f)}{(F(f-1) + F(f+1))/2}$$

と定義し、$G(f)$ が境界値 δ を越えるもの f を**ピーク**と定義する。ただし、$F(f)$ が未定義の場合 $F(f) = 1$ と約束する。

部分文字列増幅法の入力は文字列の集合 $\mathcal{D}$ と境界値 δ であり、出力は $\mathcal{D}$ に現われる部分文字列の集合である。まず、アルゴリズムは接尾辞木を用いて各ノードに対する頻度をカウントし、$F(f)$ を計算する。次に各 f に対し、$G(f)$ と δ からピークであるかどうか判断し、ピークと同じ回数出現する分岐語を出力する。

接尾辞木におけるノードの数から、次の補題を示すことができる。

補題 1 $\mathcal{D}$ 中に現われる部分文字列の出現回数の異なり数は $O(n)$ である。ただし、n は $\mathcal{D}$ 中の文字列長の総和である。

この補題を用いて、部分文字列増幅法が線形時間アルゴリズムであることが示される。

定理 1 部分文字列増幅法の時間計算量は $O(n)$ である。ただし、n は $\mathcal{D}$ 中の文字列長の総和である。

入力として与える境界値 δ によって、部分文字列増幅法の結果は変わる。しかし、人間の評価により実行結果の意味が変化することと同じであり、機械的に計算できる結果や計算量が境界値に依存するわけではない。つまり、どれだけ $F(f)$ が大きければテンプレートとして検出/非検出とするかという判断する値というだけである。その意味で、類似のアルゴリズムである頻出パタンマイニングの最小サポートやクラスタリングにおけるクラスタ数の閾値とは本質的に異なる。これらのアルゴリズムの閾値は、判断のための値でもあるが、判断の前の計算にも閾値が影響を与える。つまり、これらの閾値を変化させると、一般にアルゴリズムは計算をやりなおす必要がある。一方、部分文字列増幅法では、境界値の値を変化させても、枝刈りなどはしないため $F(f)$ そのものの分布は影響は受けず、また、計算量もオーダーの意味では変化しない。そのため、この値を閾値ではなく境界値と呼ぶことにする。

3. データ発見問題への応用

本節では、部分文字列増幅法をデータ発見問題へ応用し、実際のデータを用いた実験によりその有効性を示す（節 3.1 参照）。また、様々な言語に対する部分文字列の総出現数のグラフを用いて言語に依存しないことを示し（節 3.2 参照）、最後にノイズの分離のための境界値の自動的な設定について考察する（節 3.3 参照）。

図 5 62 回出現するテンプレートから生成された HTML ファイル。左側のナビゲーション用リンクと上方の検索フォームがこれらのファイルに共通である

Australia

【University Level Academic Exchange Counterpart Institutions】

Institution	Beginning Date
The University of Queensland*	1993.12.15

【Faculty Level Academic Exchange Counterpart Institutions】

Institutions	Faculties	Beginning Date
Faculty of Law and Arts,Monash University*	Faculty of Law	1996.6.14
IPS Radio and Space Services, Australian Department of Administrative Services	Faculty of Science	1997.3.24
School of Mathematical Sciences,Australian National University	Graduate School of Mathematics	1997.12.15

〔BACK〕

ベルギー

【大学間学術交流協定】

大　学　名	所　在　地	締結年月日
ルーヴァン・カトリック大学*	ベルギー王国 ルーヴァン・ラ・ヌーブ市	1984.1.25
レウヴェン・カトリック大学*	ベルギー王国 ルーヴァン市	1984.2.14

〔戻る〕

図 4 57 回出現するテンプレートから生成された HTML ファイル。上は英語で下は日本語で記述されている

3.1 九大ページを対象としたデータ発見

入力ファイルは、九州大学のトップページ☆からリンクを 3 段階までたどって収集した 598 個の HTML ファイル（約 5.6M バイト）である。単一サイトではあるが、後述するように様々なテンプレートが見つかった。

図 3 が、これらのファイルを入力とした $F(f)$ のグラフである。図 1 と比較すると、$F(f)$ の変化は不規則でピークと考えられるものも小さい。以下にいくつかのピークのテンプレートを具体的に見ていく。

57 回出現するテンプレートは交換留学の Web ページに用いられる（図 4 参照）。これらのページの内容は英語か日本語で記述されている。また、各ページは表を含んでいるが、表の数はページにより異なる。

62 回出現するテンプレートは 62 個のファイルで用いられている。これらのうちの一つは九大のトップページであり（図 5 の一番左側）、右の 2 つの図も同じスタイルである。

上述したテンプレートはページ全体の見栄えに影響を与えているが、“HOME” と “Back” ボタンを共有するだけの見映えとしては局所的なテンプレートも問題なく発見できた。

3.2 様々な言語における $F(f)$ 分布

本節では、自然言語における $F(f)$ 分布の傾向について考察する。

Zipf の法則は単語の頻度に関する法則であり、ある程度短い長さの部分文字列に対する法則と言える。一方、部分文字列増幅法では、自然言語に関する知識を用いず、単なる文字列（文字 n グラム）を対象にしている。ある程度小さい n に対しては、その頻度分布は単語のそれと似たものになり、言語毎の特徴をある程度包含していると予想される。

しかし、部分文字列増幅法においては任意の長さ☆☆の部分文字列の頻度を足しこむため、文法的な意味など無視した大きな n に対する部分文字列の頻度が支配的になる。つまり、$F(f)$ のグラフは言語に依存しない普遍的な性質を示すのではないかと予想される。

本節ではこの予想を確かめるために、ECI/MCI 多言語コーパス☆☆☆より得られた UTF-8 でエンコードされたチェコ語、ドイツ語、オランダ語、フランス語、ドイツ語、イタリア語、スペイン語、トルコ語および毎日新聞☆4より日本語、9 言語約 17 万強の文書について次のような調査を行った。文書の内訳は、チェコ語 2207 文書、オランダ語 854 文書、英語 3049 文書、フランス語 8963 文書、ドイツ語 50374 文書、イタリア語 1033 文書、日本語 111497 文書、スペイン語 374 文書、トルコ語 754 文書である。

$F(f)$ は文書サイズに依存するので、次のように正規化し、その分布を $F'(f)$ とする。

$$F'(f) = \frac{F(f)}{\sum_f F(f)}$$

このような正規化した分布 $F'(f)$ を各文書毎に算出し、標本平均と標本分散を求めた。文書 d に対する $F'(f)$ を、$\langle d \rangle$ を肩に添え $F'^{\langle d \rangle}(f)$ と表すと、$F'(f)$ に対する標本平均 $\overline{F'(f)}$ と標本分散 s^2 は次式のように求まる。

☆ http://www.kyushu-u.ac.jp/

☆☆ 実際には最も短いファイルの長さや、10 や 30 といった適当に大きな数で固定しても構わないことが実験的に分かっている。

☆☆☆ http://www.elsnet.org/eci.html

☆4 http://cl.aist-nara.ac.jp/lab/resource/cdrom/Mainichi/MS.html

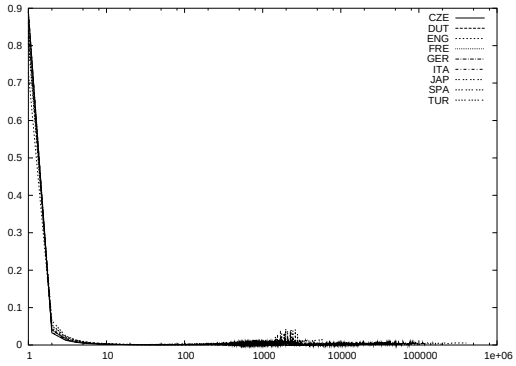

図 6 言語毎の $f \times \overline{F'(f)}$

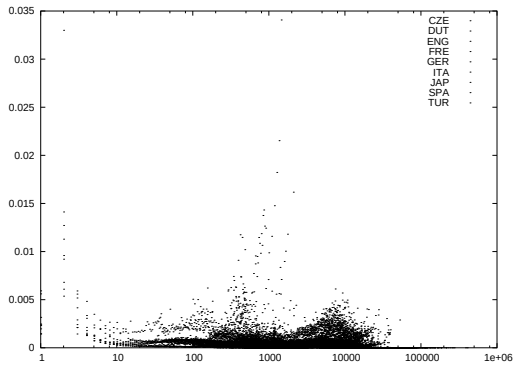

図 7 言語毎の $f \times s^2/\overline{F'(f)}$

$$\overline{F'(f)} = \frac{1}{|\mathcal{D}|} \sum_{d \in \mathcal{D}} F'^{\langle d \rangle}(f)$$

$$s^2 = \frac{1}{|\mathcal{D}| - 1} \sum_{d \in \mathcal{D}} \{F'^{\langle d \rangle}(f) - \overline{F'(f)}\}^2$$

ただし、$\mathcal{D}$ は文書の集合である。

図 8 に各 f における $f \times \overline{F'(f)}$ を $\pm s (= \sqrt{s^2})$ の範囲付きで示す。 大雑把な傾向を述べると、$\overline{F'(f)}$ は大体 $f = 40$ 周辺で底となり、それ以降は若干の振幅を繰り返しながら緩やかな増加傾向となる。また $f = 1000 \sim 1500$ までは極めて激しい振幅を描いており、これはちょうど空白文字や欧州系の言語であれば冠詞や前置詞、日本語で言えば助詞や連体詞といった、内容に依存しない表現、いわゆる機能的表現の断片（n グラムパタン）が存在する領域である。

各言語毎に $\overline{F'(f)}$ および s^2 を計算し、各言語における $f \times \overline{F'(f)}$ と $f \times s^2/\overline{F'(f)}$ を図 6 と図 7 に示す。9 言語とも上記した範囲は若干のばらつきがあるものの、$\overline{F'(f)}$ は酷似しており、標本分散を標本平均で割り込んだ $s^2/\overline{F'(f)}$ も、言語に応じてばらつきは多少あるものの、絶対的な値としては高々0.0035 を下回る程度である。したがって、文書中に混在するコンテンツとテンプレート部分の区別という点では、図 8 は言語にはほぼ無関係の自然言語の普遍的な分布と見なしても良いであろう。

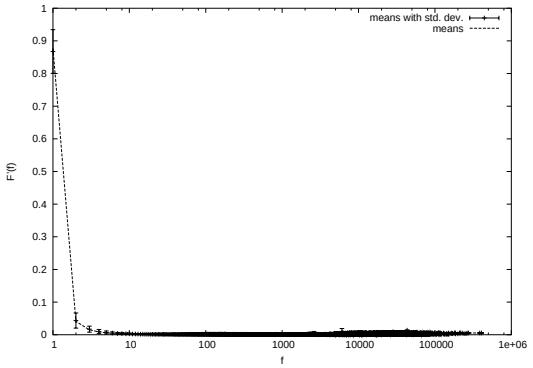

図 8 9 言語の文書に対する $f \times \overline{F'(f)} \pm s$

3.3 自然言語の分布に基づく境界値の設定

本稿では、文書中に混在しているコンテンツ/テンプレート部分を峻別することが目的の一つである。テンプレート部分を含む文書における $F(f)$ を描くと、明らかに先の図とは乖離した $F(f)$ が幾つか見て取れる。コンテンツ部分における $F(f)$ が自然言語のそれに従うと仮定すれば、結局、自然言語における $\overline{F'(f)}$ から乖離したものは排他的にテンプレート部分として検出することができる。

例えば、各 f において $\overline{F'(f)}$ が正規分布に従うと仮定すると、片側検定の考え方によって次のようにテンプレートの検出/非検出の境界値を設定することができる。本手法を適用する問題に応じて危険率（自然言語部分であるにもかかわらず、非自然言語部分と判定する確率）が適宜に設定できるとし、これを α とする。$f(x)$ を正規分布 $N(0, 1)$ に従う確率密度関数、z^*_α を

$$\int_{z^*_\alpha}^{\infty} f(x)dx = \alpha$$

となる定数であれば、境界値は $sz^*_\alpha + \overline{F(f)}$ となり、d において $F'^{\langle d \rangle}(f)$ がこの値以上を取る f には、危険率 α で非自然言語部分では無いパタンが含まれている、と判定する。

4. ま と め

本稿では、部分文字列増幅法を用いて情報抽出やラッパー生成アルゴリズムの入力となるテンプレートを共有するファイル群を見つけることができることを示した。この手法は、言語の頻度分布を利用するが、9 つの言語で書かれたコーパスを用いて言語に依存しないことを示した。

本来のテンプレートが構成するピークと、偶然コンテンツ中に頻出したためピークのように見える $F(f)$ とを峻別する境界値の設定が必要である☆。本稿では、幾つかの

☆ ただし、部分文字列増幅法は境界値による枝刈り等は行なっていないので、$F(f)$ の計測後に動的に δ を変更することも可能である。

自然言語文書から $\overline{F'(f)}$ を求めることで、統計的に境界値を設定する手法を示した。今後、実験的にその有効性についても検討していく予定である。

参 考 文 献

1) A. Arasu and H. Garcia-Molina. Extracting Structured Data from Web Pages. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, pp. 337–348, 2003.
2) R. Baumgartner, S. Flesca, and G. Gottlob. Visual Web Information Extraction with Lixto. In *Proceedings of the 27th International Conference on Very Large Data Bases*, pp. 119–128, September 2001.
3) C.-H. Chang and S.-C. Lui. IEPAD: Information Extraction Based on Pattern Discovery. In *Proceedings of the 10th International World Wide Web Conference*, pp. 4–15, 2001.
4) W. W. Cohen and L. S. Jensen. A Structured Wrapper Induction System for Extracting Information from Semi-structured Documents. In *Proceedings of IJCAI 2001 Workshop on Adaptive Text Extraction and Mining*, 2001. `http://www.smi.ucd.ie/ATEM2001/proceedings/`.
5) V. Crescenzi, G. Mecca, and P. Merialdo. RoadRunner: Towards Automatic Data Extraction from Large Web Sites. In *Proceedings of the 27th International Conference on Very Large Data Bases*, pp. 109–118, September 2001.
6) V. Crescenzi, G. Mecca, and P. Merialdo. Wrapping-Oriented Classification of Web Pages. In *Proceedings of the 2002 ACM symposium on Applied Computing*, pp. 1108–1112, 2002.
7) D. W. Embley, Y. S. Jiang, and Y.-K. Ng. Record-Boundary Discovery in Web Documents. In *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, pp. 467–478, 1999.
8) D. Ikeda. *Autoschediastic Text Mining Algorithms*. PhD thesis, Graduate School of Information Science and Electrical Engineering, Kyushu University, March 2004.
9) D. Ikeda and Y. Yamada. Gathering Text Files Generated from Templates. In *Proceedings of Workshop on Information Integration on the Web (IIWeb-04)*, August 2004. `http://cips.eas.asu.edu/iiweb.htm` (to appear).
10) D. Ikeda, Y. Yamada, and S. Hirokawa. A Pattern Discovery Algorithm by Substring Amplification. *IPSJ Transactions on Mathematical Modeling and Its Applications*, 2004. (in Japanese, to appear).
11) N. Kushmerick, D. S. Weld, and R. B. Doorenbos. Wrapper Induction for Information Extraction. In *Proceedings of the 15th Intenational Joint Conference on Artificial Intelligence*, pp. 729–737, 1997.
12) I. Muslea, S. Minton, and C. A. Knoblock. STALKER: Learning Extraction Rules for Semistructured Web-based Infomation Sources. In *Proceedings of AAAI-98 Workshop on AI and Information Integration*, pp. 74–81, 1998.
13) Y. Yamada, D. Ikeda, and S. Hirokawa. Automatic Wrapper Generation for Multilingual Web Resources. In *Proceedings of the 5th International Conference on Discovery Science*, Lecture Notes in Computer Science 2534, pp. 332–339. Springer-Verlag, November 2002.
14) 山田, 池田, 廣川. 構造的類似性を持つ半構造化文書における頻度分析. 第 2 回情報科学技術フォーラム一般講演論文集 2 分冊, pp. 59–60, September 2003.
15) 池田, 山田, 廣川. 文字列の頻度分布による共通パタン発見. 第 72 回情報処理学会情報学基礎研究会, pp. 25–32, September 2003.
16) 池田, 山田, 廣川. 文字列増幅法による共通パタン発見アルゴリズム. 第 47 回情報処理学会数理モデル化と問題解決研究会, pp. 45–48, December 2003.