九州大学学術情報リポジトリ
Kyushu University Institutional Repository

FFTを用いた繰り返しパターン発見手法の提案

中藤, 哲也
九州大学情報基盤センター

廣川, 佐千男
九州大学情報基盤センター

https://hdl.handle.net/2324/2964

出版情報：情報処理学会研究報告 ： データベースシステム. 2003 (71), pp.311-318, 2003-07. 情報処理学会
バージョン：
権利関係：ここに掲載した著作物の利用に関する注意 本著作物の著作権は（社）情報処理学会に帰属します。本著作物は著作権者である情報処理学会の許可のもとに掲載するものです。ご利用に当たっては「著作権法」ならびに「情報処理学会倫理綱領」に従うことをお願いいたします。

KYUSHU UNIVERSITY

社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

FFTを用いた繰り返しパターン発見手法の提案

中藤　哲也†　　廣川佐千男†

† 九州大学情報基盤センター　〒 812-8581 福岡市東区箱崎 6-10-1
E-mail: †{nakatoh,hirokawa}@cc.kyushu-u.ac.jp

あらまし　半構造テキスト中から自明でない情報を取り出す取り出す技術である，データマイニング，あるいはテキストマイニングは，拡大する WWW 上の情報を取り扱う上で非常に重要である．その技術の一つとして，対象のデータに繰り返し出現するパターンを発見する問題がある．発見されたパターンを用いることで，そのデータを加工する，あるいはデータから新たな情報を抽出する事が可能となる．繰り返しパターンを発見する方法として，対象となるデータをそれ自身のコピーと位置をずらして重ね，一致部分を見つける素朴な方法が考えられる．しかしこの方法は，テキストのサイズ n に対して計算量が $O(n^2)$ となり，大きなデータに対しては現実的でない．本研究では，我々が提唱している FFT を用いた効率的な近似文字列照合アルゴリズムを適用し，$O(n \log n)$ の計算量で繰り返しパターンを発見する手法について提案する．

キーワード　繰り返しパターン発見，マイニング，半構造データ，近似文字列照合，検索エンジン，FFT

Finding Repetitive Patterns Using FFT

Tetsuya NAKATOH† and Sachio HIROKAWA†

† Computing and Communications Center, Kyushu University
Hakozaki 6-10-1, Higashi-ku, Fukuoka, 812-8581 Japan
E-mail: †{nakatoh,hirokawa}@cc.kyushu-u.ac.jp

Abstract　Data-Mining or Text-Mining, that is technique to extract non-obvious information from semi-structured texts, has been very important technologies when we handle expanding information in WWW. One of them is to discover patterns that appear in the data repetitively. Using the patterns, we can process the data and can extract from the data. To discover them, we can think about the naive method, i.e. the method of aligning data with that own shifted copy data, and compare them. However, when the size of the text is n, time complexity of this method becomes $O(n^2)$, and it isn't efficient for big data. In this paper, we propose the technique to reduce time complexity of the method to $O(n \log n)$ using our string matching algorithm with mismatches.

Key words　Finding Repetitive Patterns, Mining, Semi-structured Text, String Matching with Mismatches, Search Engine, FFT

1. はじめに

WWW 上の膨大な情報は，それを利用しようとする人間にとって非常に有用な情報になりうるが，情報を整理し必要なものを抽出する作業もその膨大さゆえに容易なものではない．このため，計算機自体に情報抽出の作業を行なわせる各種試みが行なわれているが，本来人間が扱う事を目的としている情報である為に，非均質でありまた構造化されておらず，自動的な情報抽出も容易ではない．

このような構造を持たないテキストや半構造テキスト中から自明でない情報を取り出す事を，データマイニング，あるいはテキストマイニングと呼び，特に対象を Web ページとするものを Web マイニングと呼ぶ．これらは拡大する WWW 上の情報を取り扱う上でも非常に重要な技術である．

マイニングの重要な技術の一つに，データに繰り返し出現するパターンの抽出がある．明確な構造を持たないデータに構造を与える事で，変換や比較，マージなどの操作を行なう事が可能となる．あるいは自明でない新しい情報を抽出する事ができる．

Web データ上からパターンを発見する研究は，これまでも盛んに行なわれてきた．Kushmerick ら [14] は HTML ファイルのタグ列に対するラッパー帰納問題として定式化し，効率的なアルゴリズムを与えた．また，坂本ら [21] は，HTML による WebPage を木構造と捉えて木構造から情報を抽出する

シフト量	0	1	2	3	4	5	6	7	8	9							
テキスト	a	c	b	a	d	b	a	c	b	d							
テキスト	a	c	b	a	d	b	a	c	b	d							
		a	c	b	a	d	b	a	c	b	d						
			a	c	b	a	d	b	a	c	b	d					
				a	c	b	a	d	b	a	c	b	d				
					a	c	b	a	d	b	a	c	b	d			
						a	c	b	a	d	b	a	c	b	d		
							a	c	b	a	d	b	a	c	b	d	
								a	c	b	a	d	b	a	c	b	d
一致数		10	0	0	4	0	1	3									

図 1　テキスト acbabbaccb のシフト量と一致数

TreeWrapper を提唱している．HTML に繰り返し出現するパターンに注目する研究としては，PAT を使って繰り返しを抽出するものが Chang ら [6] によって提唱されている．有村ら [2] は，パターンサイズを k に制限し，その中に d 個のフレーズが出現する近接語相関パターンを提唱し，効率的なアルゴリズムを与えている．

繰り返しパターンを発見する方法として，対象となるデータをそれ自身のコピーと位置をずらして重ね，一致部分を見つける素朴な方法を考える事ができる．データ中に同じパターンが現れているならばそのパターンが上手く一致する位置があり，重ねる位置を変えながらチェックする事で繰り返しパターンを見いだせる．少ないデータを対象にする場合，人間の視覚によるチェックを行なえば，この方法でパターンを発見する事が可能であり，実際，大量のデータに対して計算機による補助を用いながらも，人間の高い視覚処理能力に頼って遺伝子の繰り返しパターンを発見する試みが存在する [20]．

しかし，この方法を計算機を用いて単純に実装すると，テキストのサイズ n に対して計算量は $O(n^2)$ となり，大きなデータに対しては現実的でない．本研究では，我々が提唱している効率的な近似文字列照合アルゴリズムを用い，$O(n \log n)$ の計算量で繰り返しパターンを発見する手法について提案する．

2. 繰り返しパターンの発見手法

Σ を有限のアルファベットとし，Σ^* の要素を文字列と呼ぶ．文字列 w の長さを $|w|$ で表す．以後では，長さ n のテキスト $T = t_1 t_2 \cdots t_n \in \Sigma^*$ について考える．

T に何らかの繰り返しパターンが存在する場合，T とそれ自身のコピーを重ねて徐々にシフトして行くと，パターン同士がちょうど一致する位置が見つかる．シフト毎のシンボルの一致の数をカウントすると，パターンが重なる特定のシフトでのシンボルの一致の数は必然的に多くなる．このシンボルの一致数を比較すれば，パターンが高い頻度で一致するシフト量を割り出す事ができる．図 1 にシフト量と一致数の概要を示す．

このようなシフト量が決まれば，このシフト量における繰り返しパターンを発見するのは容易である．テキスト全体を一度スキャンすれば，全てのパターンを収集できる．すなわち，計算量は $O(n)$ である．これについては後述する．

この手法における問題点は，全てのシフト量におけるシンボルの一致の数を求める計算である．1 シフトする毎に一致の数をカウントする単純な方法では，$O(n^2)$ の計算量が必要となる事は明らかである．我々は，この計算量を削減するために FFT を用いた近似文字列照合アルゴリズムを用いる事を提案する．計算量は $O(n \log n)$ である．

次節以降において，我々の提案している効率的な近似文字列照合アルゴリズムについて概説する．それを用いた全シフト量におけるシンボルの一致数の計算方法を示し，更にそれを用いてパターンを取り出す方法を説明する．

3. 不一致を許す文字列照合アルゴリズム

文字列照合問題は，$T = t_1 \cdots t_n$ と $P = p_1 \cdots p_m$ をそれぞれテキストとパターンと呼ばれるアルファベット Σ 上の文字列とする時，テキスト T に現れるパターン P の出現位置を全て見つける問題である．この問題の解は $O(n)$ で得られる事が既に知られている．これに対し，パターン P を編集したものをテキスト T 中に見つける問題を**近似文字列照合問題**と言い，単なる文字列照合問題より難易度は高い．編集の種類としては，代入や挿入，削除なとがあり，その違いは編集距離として定義される．

近似文字列照合問題の特殊な場合として，編集に代入のみを許した**不一致を許す文字列照合問題**があり，その距離はハミング距離として定義される．パターン長とハミング距離の差がマッチングのスコアとなる．スコアはテキスト T 上の全ての位置で得られるので，この問題はテキスト T 上の全ての位置におけるパターン P とのマッチングのスコアのベクトル $C(T, P) = (c_1, \ldots, c_{n-m+1})$ を求める問題とみなす事ができる．ここで各 c_i は，T の部分文字列 $t_i \cdots t_{i+m-1}$ と P の間のシンボルの一致の数であり，$c_i = m$ は，テキスト中の i 番目の位置にパターンそのものが現れている場合である．

スコアベクトルを求めるには，例えば単純にシンボルの比較を m 回ずつ $n - m + 1$ 個の i について行なえば良く，この素朴なアルゴリズムの計算量は $O(mn)$ であることが容易に分かる．この問題に関しては，これまでに多くの研究が行なわれている [7] [9]．Fischer ら [8] は，文字列の比較に畳み込みが使えることを見いだした．その原理に基づき高速フーリエ変換 (FFT) を用いた計算量 $O(|\Sigma| n \log m)$ のアルゴリズム [9] が示されている．また，Abrahamson [1] は，一般化された文字列照合を $O(n\sqrt{m \log m})$ で得るアルゴリズムを示した．ハミング距離の近似を得る方法としては，Karloff [11] が $O(\frac{n}{\epsilon^2} \log^c m)$ のアルゴリズムを示している．c は定数項，ϵ は近似の良さを表している．不一致数を k に制限した k-mismatch 問題にもさまざまな解決が図られており，Amir ら [4] は $O(n\sqrt{k \log k})$ と $O((n + \frac{nk^3}{m}) \log k)$ の二つのアルゴリズムを示した．

近年，Atallar らはスコアベクトルを求めるためのモンテカルロ型確率アルゴリズムを提案した [3]．これは，スコアベクトルをテキストとパターンに対応する二つの写像関数の畳み込みで表し，FFT を利用して計算する方法である．この時，正確な値を求めるために必要となる非常に多くの写像についてその全てを計算する代わりに，ランダムに選んだ k 個の写像標本の平均によってスコアベクトルの推定を確率的に行なう事で，計

算量を $O(kn\log m)$ に抑えている．また，馬場ら [5] は Atallar らと同様の手法において写像数をより少なくする写像の改良を行なっている．

我々は，馬場ら [5] と同様の手法において，写像の数を大きく削減する手法を示した [15]．計算量は k 個の標本について馬場らと同じく $O(kn\log m)$ であるが，標本の元となる総写像空間は充分に小さく，k は $1 \leqq k \leqq |\Sigma|$ である．$k = |\Sigma|$ とする事で決定性アルゴリズムとして扱うことも可能である．k の取り方で計算量と精度を自由にコントロールできるので，入力の状況に応じた処理が可能である．

3.1 準　　備

Σ を有限のアルファベットとし，Σ^* の要素を文字列と呼ぶ．文字列 w の長さを $|w|$ で表す．また集合 S の要素数を $|S|$ によって表す．

シンボルの一致を表す関数 δ:$\Sigma \times \Sigma \to \{0,1\}$ を次式のように定義する．

$$\delta(a,b) = \begin{cases} 1 & (a = b \text{ のとき}) \\ 0 & (a \neq b \text{ のとき}) \end{cases}$$

但し，$a, b \in \Sigma$ である．

テキスト $T = t_1 t_2 \cdots t_n \in \Sigma^*$ とパターン $P = p_1 p_2 \cdots p_m \in \Sigma^*$ について，

$$c_i = \sum_{j=1}^{m} \delta(t_{i+j-1}, p_j) \qquad (1 \leqq i \leqq n-m+1) \qquad (1)$$

として得られるベクトル $C(T,P) = (c_1, c_2, \ldots, c_{n-m+1})$ を T と P の間のスコアベクトルと呼ぶ．つまり，c_i は P の先頭のシンボルが T の i 番目の位置にあるときの一致の数である．

3.2 FFT を使うための条件

c_i は式 (1) の定義より $\sum_{j=1}^{m} \delta(t_{i+j-1}, p_j)$ である．この δ 関数を何らかの方法で，t の関数と p の関数の積へ変換することができれば，ベクトル $C(T,P)$ の計算は，T と P の畳み込み (convolution) に変換可能である．畳み込みで表現されたベクトル $C(T,P)$ は高速フーリエ変換 (FFT) によって，$O(n\log n)$ で計算できる．加えて, [3] と同様に [7] の技法を用いる事で，計算量を $O(n\log m)$ に落とす事ができる．

シンボルの一致を表す δ 関数を，t の関数と p の関数の積で直接表現する方法は無い．次に紹介する Atallar らの方法 [3]，馬場らの方法 [5]，本稿で提案する方法はいずれも，アルファベットの集合をまず複数の写像に変換し，各写像に対する結果の総和が δ 関数と一致するような関数を導入して，δ 関数を間接的に表現している．

3.3 決定性アルゴリズム

アルファベット Σ のある一つのシンボルのみを 1 とし，その他のシンボルを -1 とする $\{1, -1\}$ への写像の集合を Ψ_Σ とする．このとき，明らかに $|\Psi_\Sigma| = |\Sigma|$ である．文脈から明らかな場合，Ψ_Σ を Ψ と略記する．

あるシンボル $x \in \Sigma$ を 1 に変換する写像 $\psi_x \in \Psi$ は次式のように定義される．

$$\psi_x(a) = \begin{cases} 1 & a = x \text{ のとき} \\ -1 & a \neq x \text{ のとき} \end{cases}$$

このとき，二つのシンボル $a, b \in \Sigma$ について，それらの一致を表す δ 関数は，次式で表される．

$$\delta(a,b) = \frac{1}{4} \sum_{\psi_x \in \Psi} \psi_x(a)\psi_x(b) - \frac{|\Sigma|}{4} + 1 \qquad (2)$$

式 (1) と式 (2) より，スコア c_i に関して次式が得られる．$1 \leqq i \leqq n-m+1$ である，あらゆる i について，

$$c_i = \frac{1}{4} \sum_{x \in \Psi} \sum_{j=1}^{m} \psi_x(t_{i+j-1})\psi_x(p_j) - \frac{m|\Sigma|}{4} + m \qquad (3)$$

よってスコアベクトル $C(T,P)$ は，

$$\sum_{j=1}^{m} \psi_x(t_{i+j-1})\psi_x(p_j)$$

の計算を $|\Sigma|$ 回行ない，線形変換を行なうことで計算できる事が分かる．

［定理 1］ スコアベクトル $C(T,P)$ は $O(|\Sigma| n\log m)$ で正確に計算できる．

3.3.1 確率アルゴリズム

前節のアルゴリズムは効率的ではあるが，それでもなお，アルファベットサイズ $|\Sigma|$ と同じ回数の FFT 計算が必要であるので確率化を試みる．

二つのシンボル $a, b \in \Sigma$ について，それらの一致を表す δ 関数は，写像 Ψ_Σ から k 個を取り出して作った部分集合 Ψ_k を用いて次の期待値で表される．

$$\delta_k(a,b) = \frac{|\Sigma|}{4k} \sum_{\psi_x \in \Psi_k} \psi_x(a)\psi_x(b) + \frac{4 - |\Sigma|}{4}$$

この δ 関数により，スコア c_i に関して次式が得られる．$1 \leqq i \leqq n-m+1$ である，あらゆる i について，

$$c_i = \frac{|\Sigma|}{4k} \sum_{\psi_x \in \Psi_k} \sum_{j=1}^{m} \psi_x(t_{i+j-1})\psi_x(p_j) + \frac{m(4-|\Sigma|)}{4}$$

よってスコアベクトル $C(T,P)$ は，

$$\sum_{j=1}^{m} \psi_x(t_{i+j-1})\psi_x(p_j)$$

の計算を k 回行ない，一度の線形変換を行なうことで確率的に求まる．

［定理 2］ スコアベクトル $C(T,P)$ は k 個のサンプルを用いて確率的に計算できる．得られた値の期待値は正しい値と一致する．計算量は，$O(kn\log m)$ である．

得られたアルゴリズムを図 2 に示す．

本アルゴリズムは，Atallar ら [3] や馬場ら [5] によるアルゴリズムと同様，FFT による畳み込みを用いるものであるが，より少ない写像をベースとした確率アルゴリズムであり，正確なスコアを求める決定性アルゴリズムとしての動作を含め，精度と計算量を自由に設定できる．計算量は $O(kn\log m)(1 \leqq k \leqq |\Sigma|)$ である．

Procedure ComputeScore
入力：Σ^* 上のテキスト $T = t_1 \cdots t_{(1+\alpha)m}$
入力：Σ^* 上のパターン $P = p_1 \cdots p_m$
出力：スコアベクトル $C(T, P)$
for $\psi_\ell \in \Psi_k$ **do begin**
　$\psi_\ell(T)$ を求め T_ℓ とする．
　　(T_ℓ は長さ $(1+\alpha)m$ の $\{0,1\}$ 上の列である．)
　$\psi_\ell(P)$ の後ろに αm 個の 0 を付加し P_ℓ とする．
　　(P_ℓ は長さ $(1+\alpha)m$ の $\{0,1\}$ 上の列である．)
　P_ℓ の列を反転する．これを P_ℓ^R とする．
　T_ℓ と P_ℓ^R の畳み込み c_ℓ を FFT により計算する．
end
$\sum_{\psi_\ell \in \Psi'_\Sigma} c_\ell$ を計算し，
必要な一次変換を行なって $C(T, P)$ として出力する．

図 2　アルゴリズム

4. 繰り返しパターンの抽出

4.1 スコアベクトルの計算

不一致を許す文字列照合アルゴリズムを，パターン発見の対象としているテキストに適用する．同一のテキスト間での本アルゴリズムの計算量は，$O(n \log n)$ である．

4.2 パターンの選択

テキスト中には一般に複数の繰り返しパターンが存在する．本手法により計算されたスコアベクトルは繰り返しパターンの存在を良い確率で示唆するが，必ずしも最頻繰り返しを示すものではない．このため，スコアベクトル中から値の大きなスコアを持つシフト量を選び，そのシフト量に関する繰り返しパターンを抽出する．

抽出方法には，抽出したいパターンの性質によって複数の方法が考えられるが，本稿では連続する繰り返しを見つけるために次の方法を用いた．

テキスト：T
シフト量：s
位置を示す変数：i
注目中のパターンを保持するキュー：Qp
Qp の各シンボルのカウントを保持するキュー：Qn

1) 前方（左側）からテキストをスキャン（位置は i）する
2) Qp からシンボルを一つ取り除き，$T(i)$ と比較
　2-1) 一致する場合，
　　Qn から 1 つ取り除く
　　その値に 1 を加えて Qn に追加．
　　その値以上の Qn の要素位置を調べる
　　その位置の Qp からなるシンボル列を構成
　　例：Qp が {a,b,c},Qn が {3,2,3} → シンボル列 “a*c”
　2-2, 不一致の場合，Qn から 1 つ取り除き，Qn に 0 を追加
　　構成したシンボル列をハッシュから探す．
　　登録があればその値を 1 増やす．
　　登録がなければ，値 1 としてハッシュに新規に登録する．
3, 全テキストをスキャン後，
　ハッシュからパターンと値を取り出す．
4, パターン中ワイルドカードを除くシンボル数に
　ハッシュに登録された値（シンボルの繰り返し数）
　を乗じて，そのパターンのスコアとする．

以上により，部分的な不一致を許容した高頻度の繰り返しパターンが得られる．テキストは一度スキャンするだけで良いので，計算量は $O(n)$ である．

5. 実　　験

一般に検索エンジンの返す検索結果は，あるパターンの繰り返しである事が多い．しかしながら，その繰り返しを特定して結果を切り分けるのは容易ではない．

我々は，WWW 上から必要な情報を的確に得る手段の一つとして自動構築型メタサーチエンジン DAISEn（後述）を開発している．このシステムにおいても，検索結果のパターン抽出は主要な技術である．

本稿で提案する繰り返しパターン発見手法を，この DAISEn システムにおける新しいパターン抽出法をして応用する事を考えており，今回，幾つかの例に適用してその有効性や問題点を調べる事とした．

まず，DAISEn システムの概要について説明した後，検索結果からのパターン抽出についての実験結果を示す．

5.1 自動構築型メタサーチシステム DAISEn について

Web 上の膨大な情報から必要な情報を集める為に，一つ一つのページを調べて回るには，WWW はあまりにも膨大すぎる．そこで，我々は Yahoo!や google などのような WWW 全体を対象とする検索エンジンを利用することで，必要な情報にたどり着く．このような検索エンジンでは，検索結果の表示方法やランキングに様々な工夫がなされているにも関わらず，求める情報と無関係なページが多く含まれることがあり，検索結果の質が問題となる．

一方，企業や学会などの Web サイトにおいては，自サイト内の情報やデータベースについて検索機能をつけているところが増えている．本稿ではこのようなサイトを検索サイトと呼ぶ．検索サイトは，そのサイト内の情報を効率良く提供することを目的としており，一般の検索エンジンより高品質の情報が得られる．

しかしながら，検索サイト自身が持つ情報量は一般の検索エンジンに比べると少ない傾向にあるために，利用者が必要な情報を集めるためには多くの検索サイトを調べる事が必要になる．また，そのような複数ある検索サイトについて一つ一つに対して個別に検索を行なわなければならない．

この問題に対しては，このような検索サイトを目的に応じて一括して検索するための統合システムを考えることができる．例えば DVD プレーヤーについて調べたいとき，まず Sony, Panasonic, Victor, Hitachi, AOpen などのメーカーを選択し，それらの検索サイトに一括して検索をおこなう．これによ

り，効率良く高品質の情報を入手することができる．

我々は，このような検索サイトの統合を実現するシステムとして，自動構築型メタサーチシステム DAISEn を構築している [22], [23]．本システムは，非常に多く存在する専門検索サイトから目的に合ったサイトを選択し，それらを利用して検索を行ない，統合した検索結果をユーザに提示する．非常に多い検索サイトを統一的な枠組みで扱うには，人手でサイトの情報を抽出しシステムに組み込むのでは手間がかかり過ぎる．検索サイトの自動的な取り扱いが必要である．

我々はこれまでに，検索サイトへ検索クエリーを送るためのクエリーフォームの自動生成 [18]，複数クエリーを用いた検索のための検索論理式の自動推定 [16], [19]，得られた結果を統合しユーザに提示するためにサイト毎の検索結果一覧を一件毎のデータに切り分けるためのラッパーの自動生成 [12], [23]，自動生成したラッパーの精度推定のための検索結果一覧における表示件数の推定手法の提案 [13] とそれを用いた検索サイトエディタ [24] を開発した．検索結果の HTML ページそのものを使うよりも，ラッパーで切り出した部分を用いた方がより適切な特徴ベクトルを得られることを実験的に示し [10]，それを用いた検索サイトの分類法を提案している [17]．

本稿で提案するパターン発見手法は，このシステムの主要な一部である検索サイトに対するラッパーの自動生成部に応用可能である．また，この手法は，数学的な計算に基盤を置きながら，これまで本システムで提案してきた手法 [12], [23] の自然な拡張になっている．

5.2 実験用データ

今回実験に用いた検索サイトを表 1 に示す．これらのサイトは，現 DAISEn に登録されており，現在のラッパーで切り分けの出来るサイトである．従って本実験は甘い条件下のものとなるが，新ラッパーの問題点の洗い出しの意味もあり，これらのサイトを用いた．より多くの一般的な検索サイトでの評価実験は，改めて行なう予定である．

135 のサンプリングクエリーを用いて，これらのサイトに対して自動的にデータ（検索結果）の収集を行なった．この収集には DAISEn システムを用いている．コンテンツの傾向に差があるので，サイト毎にもっとも情報量の多かった検索結果を本実験の対象とした．

前処理として，得られた HTML に対して，各タグをシンボルにし，タグ以外のテキストはタグで囲まれた領域を一つのシンボルとした．またヒューリスティックスの一つとして，クエリーを強調するタグを無視した．

5.3 実験結果と考察

各サイトのシンボル列に対して，本稿で提案した手法を適用し，パターンの抽出を行なった．抽出されたパターンについて，人手で検索結果の HTML との比較を行い，適切にパターンが切り出せているかを確認した．結果，23 サイト中 14 サイトでは第 1 候補のシフト量から求められたパターンが最適なものであった．

更に第 5 候補までのシフト量についてパターンを求め，それぞれ最大のスコアを持つものについて，スコアの比較を行なった．結果を表 2 に示す．表中，適切なパターンを持つものにアンダーラインを付けている．この表から分かるように，先の 14 サイトを含む 15 サイトは適切なパターンがこの手法により自動的に選ばれる事が分かる．No.5 のサイトは，スコアの比較でも第 2 候補が目的のパターンであった．他の候補はリンク集などの繰り返しパターンであった．

表 2 実験結果

サイト番号	第 1 候補スコア	第 2 候補スコア	第 3 候補スコア	第 4 候補スコア	第 5 候補スコア
1	162	144	108	8	11
2	532	504	420	448	420
5	16	25	14	9	16
6	243	216	162	28	30
8	81	133	396	99	96
9	380	176	160	128	144
14	558	496	72	48	268
20	30	18	29	47	23
21	76	38	54	39	40
24	984	924	882	840	100
27	2280	2112	2016	1920	1916
32	200	126	84	126	107
33	147	99	80	28	30
34	323	306	255	272	24
37	105	138	55	60	80
38	216	228	216	192	180
39	589	570	560	532	475
40	14	9	11	19	16
42	408	324	390	336	360
44	418	100	285	42	216
47	420	192	70	25	90
49	399	392	378	364	350
53	40	50	36	140	83

次に適切なパターンが得られなかった 7 サイトについて詳細に調べた．その結果，適切なパターンが得られなかった理由は次のようなものであった．

- 目的としているもの以外（リンク集など）の繰り返しパターンが得られているサイト：16, 21, 37, 38, 42
- 検索結果が小さく，明確な繰り返しが認められなかったサイト：40
- その他：20, 53

6. まとめと今後の課題

FFT 計算を基礎とした不一致を許す文字列照合アルゴリズムを利用する，繰り返しパターン発見手法を提案した．また実験により，実際に繰り返しパターンが得られることを示した．

本手法の特徴は，全テキストを数学的な計算を用いて網羅的に調べる事にある．このため，事前にパターンのサイズを制限する必要がなく，テキスト全体に及ぶような大きなパターンも理論的には抽出可能である．これは他の手法にはない特徴である．また複数のパターンが，そのパターンの有効性の指標となるスコアを持って複数得られる点も，特徴の一つである．更にシンボルに対する各種のヒューリスティックスの導入が容易な点があげられる．ヒューリスティックスを用いたパターン，用いないパターン，いずれも同じ形式とスコアを持つので，同じ指標で比べる事が可能である．

表 1　実験に用いたサイトの URL

1	http://www.clayzee.com/education/classes_workshops/index.html
2	http://www.osaka-kyoiku.ac.jp/educ/index-e.html
5	http://dir.yahoo.com/Education/Statistics/
6	http://www.ael.org/eric/ned.htm
8	http://www.doe.mass.edu/
9	http://www.mcrel.org/resources/currdata/
14	http://www.ncbe.gwu.edu/
20	http://www.brassring.com/contests.html
21	http://www.growingnd.com/search/default.asp?sectionID=4&subSectionID=0&pageID=41
24	http://www.Headhunter.net/JobSeeker/Index.htm?zbid=2000121516493ab3wgwgb2
27	http://www.jobs.com/
32	http://dir.yahoo.com/News_and_Media/Television/Shows/Science_Fiction__Fantasy__and_Horror/Babylon_5/Events/
33	http://search.news.com.au/search.htm
34	http://NewsSynthesis.com/search/searchbox.htm
37	http://cnnfn.cnn.com/resources/search/
38	http://searchenginewatch.internet.com/links/News_Search_Engines/Regional_News_Search_Engines/index.html
39	http://mentalhelp.net/guide/trtsrch.htm
40	http://dir.yahoo.com/Reference/Libraries/Library_and_Information_Science /Organizations/Association_of_College_and_Research_Libraries__ACRL_/
42	http://www.ninds.nih.gov/find_people/index.htm
44	http://allhealthnet.com/Eye+Care/
47	http://www.crimes-of-persuasion.com/news.htm
49	http://www.junkbusters.com/ht/en/cookies.html
53	http://www.osOpinion.com/

本手法の問題点としては，シフトによって得られるスコアベクトルの各要素が，パターンの存在を高い確率で示しているものの，明確な物理量を持たない点があげられる．強いて言うならば，パターンの頻度とパターン長の積に相当する値であるが，同じ長さを持つ複数のパターンが混在する欠点がある．この点が実際にどの程度影響するかの評価が今後必要である．

今回は，新しいパターン発見手法をテストする目的で小規模な実験を行なったが，今後 1000 件規模の検索サイトに対する実験を行なう予定である．また，その他の応用についても検討して行く予定である．

文　献

[1] K. Abrahamson, Generalized string matching, SIAM Journal on Computing, vol.16, no.6, pp.1039-1051, 1987.

[2] H. Arimura, S. Shimozono, and S. Arikawa, Efficient Discovery of Optimal Word-Association Patterns in Large Text Databases, New Generation Computing, Vol.18, pp.49-60, 2000.

[3] M. J. Atallah, F. Chyzak, and P. Dumas, A Randomized Algorithm for Approximate String Matching, Algorithmica 29, pp.468-486, 2001.

[4] A. Amir, M. Lewenstein, and E. Porat, Faster algorithms for string matching with k mismatches, Symposium on Discrete Algorithms, pp.794-803, 2000.

[5] K. Baba, A. Shinohara, M. Takeda, S. Inenaga, and S. Arikawa, A Note on Randomized Algorithm for String Matching with Mismatches, Proceedings of the Prague Stringology Conference '02, pp.9-17, 2002.

[6] C. H. Chang, and S. C. Lui, IEPAD: Information Extraction Based on Pattern Discovery, Proc. of 10th World Wide Web Conference, 2001.

[7] M. Crochemore, and W. Rytter, Text Algorithms, Oxford University Press, New York. 1994.

[8] M. J. Fischer, and M. S. Paterson, String-matching and other products, In Complexity of Computation (Proceedings of the SIAM-AMS Applied Mathematics Sy mposium, New York, 1973), pp.113-125, 1974.

[9] D. Gusfield, Algorithms on Strings, Trees, and Sequences, Cambridge University Press, New York. 1997.

[10] S. Hirokawa, S. Watanabe, Y. Koga, and T. Taguchi, Automatic Feature extraction of Search sites, Proc. SSGRR2001(CD-ROM), 2001.

[11] H. Karloff, Fast algorithms for approximately counting mismatches, Information Processing Letters 48, 2, pp.53-60. 1993.

[12] 古賀康則，田口剛史，廣川佐千男, "検索サイト統合のためのラッパー生成法," DEWS2001 CD-ROM:6b-1, 2001.

[13] 古賀康則，酒井美由紀，廣川佐千男, "統合検索のための検索結果件数推定方法," 第 63 回情報処理学会全国大会, 2001.

[14] N. Kushmerick, D. Weld, and R. Doorenbos, Wrapper induction for information Extraction, IJCAI'97 pp.729-737, 1997.

[15] 中藤哲也，馬場健介, "近似文字列照合のための効率的なアルゴリズム," 日本データベース学会 Letters, Vol.2, No.1, pp.87-90, 2003.

[16] T. Nakatoh, Y. Koga, A. Uhl, S. Hirokawa, Automatic Estimation of Query Form for Search Sites, Proc. PYIWIT'02, pp.329-332, 2002.

[17] 中藤哲也，古賀康則，廣川佐千男, "検索統合のための検索サイト分類法," Proc. DBWeb2001, pp. 225-228, 2001.

[18] T. Nakatoh, M. Sakai, Y. Koga, S. Hirokawa, Generation of Query URL for Search Sites, Proc. SSGRR2002w(CD-ROM), 2002.

[19] 中藤哲也，酒井美由紀，廣川佐千男, "検索サイトのための集合演算子の自動推定," 情報科学技術フォーラム 一般講演論文集 第 2 分冊, pp.9-10, 2002.

[20] 大澤研二, 吉田徹彦, 宇野民幸, 尾畑伸明, "色彩パターン化を用いたゲノム情報解析：ヒト 21 番染色体中のタンデムリピートの探索," 第 24 回日本分子生物学会年会, 3P-076, 2001.

[21] H. Sakamoto, Y. Murakami, H. Arimura, S. Arikawa, Extracting Partial Structures from HTML Documents, Proc. the 14the International FLAIRS Conference. pp.264-268, 2001.

[22] 田口剛史, 古賀康則, 廣川佐千男, "検索サイトの統合システム," 第 14 回人工知能学会全国大会, pp.414-415, 2000.

[23] T. Taguchi, Y. Koga, S. Hirokawa, Integration of Search Sites of the World Wide Web, Proc.Intern. Forum cum Conf. on Information Technology and Communication, Vol.2, pp.25-32, 2000.

[24] 山田泰寛，廣川佐千男, "専門検索サイトの動的統合による次世代検索システム DAISEN における検索サイトエディタの開発," 情報科学技術フォーラム 一般講演論文集 第 2 分冊, pp.11-12, 2002.