情 報 学 基 礎　66－20
デジタル・ドキュメント　32－20
（２００２．３．１５）

極大共通生垣を用いた情報抽出手法の提案

福田 賢治　石野 明　竹田 正幸　松尾 文碩

九州大学大学院 システム情報科学府 情報理学専攻

本稿ではHTML文書を生垣(hedge)として扱い, 複数のHTML文書に対して反単一化(anti-unification)[2][3]を行なうことでそれらの共通構造を求め, その共通構造を用いることで個々のHTML文書の情報を抽出する手法を提案する. 本手法では生垣の極小共通汎化を複数のHTML文書の共通パターンとし, 共通パターンとHTML文書とマッチングを行うことにより情報を抽出する. 本稿では生垣のクラスを単純で正則なクラスと正則なクラスの二つのクラスに限定し, それぞれのクラスで極小共通汎化を求め, 実際のWebサイトから情報を抽出する実験を行った. 特に, 正則な生垣のクラスにおいては極小共通汎化の一つとして極大共通部分生垣 (maximal common subhedge, MCH) を定義した.

A Proposal of Information Extraction Based on Maximal Common Hedge

Kenji FUKUDA　Akira ISHINO　Masayuki TAKEDA　Fumihiro MATSUO

Department of Informatics, Kyushu University

In this paper, we propose the information extraction method treating HTML documents as hedges and using by anti-unification for hedges. We create a common pattern replaced the different parts in each HTML document with variables using by anti-unification for HTML documents, and extract information after matching the common pattern and a HTML document. In this paper, we define maximal common hedge(MCH) as one of the minimal general generalization for hedges. We treat MCH as the common pattern among the HTML documents. As an experiment, we extract information from actual HTML documents.

1　はじめに

インターネットの急速な普及は, 社会を大きく変化させた. 現在, 社会・経済・科学などさまざまな分野の情報が大量にインターネット上に蓄積されている. また, 今後もより多くの情報が, インターネット上で公開され, 広く利用されるものと予測される.

これらインターネット上に蓄積された情報の多くは, 人間である利用者が直接見たり読んだりするためのものであり, 計算機が自動的にこれらのデータからデータへアクセスし, 情報を自動的に取り出すためのものではない. しかし, 将来的には計算機の支援のもとに, これらの大量のデータを解析して, 人間である利用者に提示したり, 計算機同士で情報を交換したりすることが考えられる. そのためには, これらの大量の情報から計算機を用いて有用な情報を取り出すための技術が必要となっている.

また, インターネット上に蓄積されている情報の多くはHTML文書として公開されているが, これらHTML文書に含まれる情報は一定の共通構造の中に含まれていることがある. 例えば本のタイトルと著者の情報はamazon.com (`http://www.amazon.com/`) などのWebサイトにあり, 天気予報に関する情報ならばYahoo! (`http://www.yahoo.co.jp/`) の天気予報のページなどに含まれている.

そこで本稿ではHTML文書を生垣 (hedge) とし

て扱い, 複数の HTML 文書に対して反単一化 (anti-unification)[2][3] を行なうことで, 一定の共通構造の中に含まれている HTML 文書の情報を抽出する手法を提案する. 2 つの生垣に対して反単一化を行なうことによりそれらに共通する生垣が得られることは [2][3] で提案されているが, 本稿ではそれを実際の HTML 文書に対して適用し評価を行なった.

本手法は HTML 文書を生垣として扱い, 反単一化を行なって極小共通汎化を求め, それを複数の HTML 文書に共通するパターンとする. この極小共通汎化は個々の HTML 文書ごとに異なる部分が変数に置き換えられており, 極小共通汎化と情報抽出を行ないたい HTML 文書とのマッチングを行なうことで変数とマッチした文字列及び文字列と HTML 要素の列を抽出する. 極小共通汎化は生垣のクラスを限定することにより求めることができ, 単純で正則なクラスの極小共通汎化を求めるアルゴリズムは [2][3] に示した. また,[1] において単純で正則なクラスの極小共通汎化を共通パターンとして実際の Web サイトから情報を抽出する実験を行った. その結果, このクラスの極小共通汎化では表現能力が低いことがわかった.

そこで本稿では生垣のクラスを正則な生垣のクラスに拡張し, このクラスの極小共通汎化を共通パターンとして情報抽出を行う. 特に, 正則な生垣のクラスの極小共通汎化の一つとして極大共通生垣を定義した. この極大共通生垣は正則な生垣の極小共通汎化のうち最もスコアの高い生垣として定義される. また, 極大共通生垣を求めるアルゴリズムも与えた. このアルゴリズムは最長共通部分列 [11] を求める動的計画法を生垣に対して拡張したものである.

HTML 文書からの情報を抽出する他の手法としては, LRWrapper[6], WISK[9], T-Wrapper[7][8], WHIRL[10] などさまざまな手法があるが, これらの手法はラベル付けによって個々の HTML 文書ごとにラッパーを記述したり, 特別なラッパー言語を用いてラッパーを作成したりする手法である. 一方, 本手法は HTML 文書を生垣として扱い, 複数の HTML 文書に対して反単一化を行なうことでそれらに共通のパターンを生成し, 情報抽出を行なう.

2 節では本手法の基礎となる生垣と単純で正則なクラスでの極小共通汎化について説明する. 3 節では正則なクラスの極小共通汎化の一つである極大共通生垣についてその定義とアルゴリズムを与え, 4 節で極大共通部分生垣を用いた実験について述べる. 最後 5 節でまとめを行う.

2 生垣

2.1 生垣の定義

生垣の定義を以下に示す. 文字の有限集合を Σ, 名前の有限集合を N とし, Σ と N は互いに素であるとする. また変数の無限集合を X とする.

定義 2.1 生垣を次のように再帰的に定義する.

1. 空列 ε は生垣である.
2. Σ の要素である文字は生垣である.
3. X の要素である変数は生垣である.
4. 名前 n と生垣 u に対して,$n(u)$ は生垣である.
5. u,v を生垣とするとき, 列 uv は生垣である.

タグのバランスの取れた半構造化文書は生垣と一対一に対応することが [2][3] に示されている. 故にタグのバランスの取れた半構造化文書は生垣であるが, HTML 文書は終了タグが不要なタグを含むため, タグのバランスの取れていない文書である. そのため本研究では前処理として HTML Tidy[5] を用いることでタグのバランスの取れた半構造化文書として扱い HTML 文書を変数を含まない生垣 (基礎生垣, ground hedge) として表現する.

定義 2.2

1. ルートにある文字, 名前, 変数の間の関係
2. 同一の親を持つ文字, 名前, 変数の間の関係

のことを**兄弟** (sibling) という.

例 2.1 HTML 文書における表を表現するための名前は, TABLE,TR,TD,CAPTION である. これらの名前を用いた HTML 文書 D_1 を以下に示す. (読みやすさのため, 適当に改行が挿入してある).

XML 関連

タイトル	著者名	値段
最新 XML がわかる	池田実	1980 円
XML アプリケーションズ	フランク・バムフリー	4600 円

D_1 =
TABLE (CAPTION (XML 関連)
TR (TD (タイトル)
TD (著者名) TD (値段))
TR (TD (最新 XML がわかる)
TD (池田実) TD (1980 円))
TR(TD (XML アプリケーションズ)
TD (フランク・バムフリー) TD (4600 円)))

図 1: HTML 表の例

次に生垣に対する汎化 (generalization, または反単一化 anti-unificaiton) について述べる.

定義 2.3 p, q を生垣とし $p\theta = q$ となるような代入 θ が存在するとき, p は q より一般的であるといい $p \geq q$ と書く. また, $p \geq q$ かつ $q \not\geq p$ ならば, p は q より真に一般的であるといい $p > q$ と書く. $p \geq q$ $(p > q)$ であるとき, p は q の汎化 (真に汎化) であるという. さらに $p \geq q$ かつ $q \geq p$ であるとき p と q は等価であるといい, $p \equiv q$ と書く.

一般に生垣の最小共通汎化 (lgg, または最小反単一化 lca) は存在しない. そこで本手法では極小共通汎化 (mgg, または極小反単一化 mca) を情報抽出のための共通パターンとして用いる. まず共通汎化の定義を与える.

定義 2.4 q_1, q_2 を生垣とする. このとき生垣 p がすべての $i = 1, 2$ に対して $p \geq p_i$ をみたすならば p を q_1,q_2 の共通汎化という.

定義 2.5 生垣 p を生垣 q_1,q_2 の共通汎化であるとする. このとき, p が q_1 と q_2 の任意の共通汎化 p' に対して, $p \not> p'$ をみたすとき, p を q_1 と q_2 の極小共通汎化であるという.

2.2 単純で正則な生垣の極小共通汎化

与えられた生垣のペアからすべての極小共通汎化を求めることは困難である [2]. そこで極小共通汎化のうち一つを求める問題に限定し, そのようなアルゴリズムを与える単純で正則なクラスと呼ばれるクラスを以下に述べる.

定義 2.6 生垣 h において, 各兄弟の集合に変数が高々一つしかないとき, h を**単純であるという**.

定義 2.7 生垣 h において,h 中の全ての変数が異なるとき, h を**正則であるという**.

例 2.2 $\Sigma = \{a, b, c\}$, $X = \{x, y, z, ...\}$, $N = \{f, g\}$ とする. このとき $h_1 = (a\ x\ f(y\ g(b)))$, $h_2 = (g(f(x)\ f(y)\ f(z)))$, $h_3 = (x\ f(y\ f(a\ z)))$ は単純で正則な生垣である. しかし $h_4 = (a\ x\ f(b)\ y\ a)$, $h_5 = (x\ f(x))$ は単純で正則な生垣ではない. なぜなら h_4 は同じ兄弟に二つの変数 x, y が含まれており, h_5 は同じ変数 x が繰り返し出現しているからである.

単純で正則な生垣の極小共通汎化を多項式時間で計算可能なアルゴリズムは [2] に与えられている. 一般に, 単純で正則な生垣の極小共通汎化は一意に決定できない. そこで本手法では生垣の兄弟の左側の一致を優先した極小共通汎化を HTML 文書の共通パターンとした [1][3]. この共通パターンと情報を取り出したい HTML 文書とマッチングを行うことで情報が抽出できる.

例 2.3 例 2.1 と同様に HTML 表 D_2 と, 例 2.1 の D_1 と D_2 から求めた極小共通汎化を P とする.

JAVA 関連

タイトル	著者名	値段
独習 Java	ジョゼフ・オニール	3600 円
分散オブジェクト入門	中山茂	2800 円

$D_2 =$
```
TABLE (
   CAPTION (JAVA 関連)
        TR (TD (タイトル) TD (著者名) TD (値段))
        TR (TD (独習 Java) TD (ジョゼフ・オニール)
            TD (3600 円))
        TR (TD (分散オブジェクト入門) TD (中山茂)
            TD (2800 円)))
```
$P =$
```
TABLE (
   CAPTION (X1 関連)
        TR ( TD (タイトル) TD (著者名) TD (値段))
        TR ( TD (X2) TD (X3) TD (X40 円))
        TR ( TD (X5) TD (X6) TD (X700 円)))
```

図 2: 単純で正則な極小共通汎化の例

例 2.2 において D_2 と P のマッチングを行うと X_1 は文字列"JAVA", X_2 および X_5 はそれぞれ文字列"独習 Java"と"分散オブジェクト入門"とマッチし, その結果, HTML 文書 D_2 から本の分野とタイトルが抽出されることになる.

2.3 単純で正則なクラスの極小共通汎化を用いた実験

単純で正則なクラスの極小共通汎化を共通パターンとして, 実際の Web サイトから情報を抽出する実験を行った.

Web サイト名	URL
Yahoo!天気情報	http://weather.yahoo.co.jp/
Yahoo!ファイナンス	http://quote.yahoo.co.jp/
Yahoo!ミュージック	http://music.yahoo.co.jp/
goo[天気]	http://channel.goo.ne.jp/weather/
goo[ニュース]	http://channel.goo.ne.jp/news/
Excite:ニュース	http://www.excite.co.jp/News/
Excite:ミュージック	http://www.excite.co.jp/music/
Yahoo!weather	http://weather.yahoo.com/
Yahoo!Finance	http://quote.yahoo.com/
washingtonpost	http://www.washingtonpost.com/
cdnow	http://www.cdnow.com/
amazon	http://www.amazon.com/

表 1: 実験に利用した Web サイト

表 1 のそれぞれの Web サイトごとに極小共通汎化を求め共通パターンを生成し, 同一 Web サイトの HTML 文書からマッチングを行って情報を抽出した. このとき各 Web サイトの内容を考慮し, その Web サイトで重要であると考えられる情報を抽出項目 (正解集合) として選んだ. この抽出対象の項目とは, 例えば Yahoo!天気情報ならば, 都市名, 日付, 天気, 最高気温, 降水確率 (6 時間おき), 風向き, 波の高さの 10 項目である. また, これら抽出対象の各項目がきちんと抽出されたかど うかを評価するために再現率 (recall) を求めた. 再現率の定義は以下である.

$$recall = \frac{\text{抽出に成功した項目数}}{\text{抽出対象の項目数}}$$

共通パターンの生成に用いた HTML 文書数, 共通パターン内の変数の数, 抽出対象の項目数及びその再現率を表 2 に示す. 表 2 からほとんどの Web サイトが高い再現率を示しており, 抽出項目が取り出せていることがわかる. しかし, 情報抽出に失敗した Web サイト (Yahoo!ファイナンス他) もあった. 詳しい考察は [1] を参照.

Web サイト名	HTML 文書数	変数の数	項目数	再現率
Yahoo!天気情報	293	76	10	100%
Yahoo!ファイナンス	80	4	22	0%
Yahoo!ミュージック	8	165	162	100%
goo[天気]	107	91	68	93.6%
goo[ニュース]	10	35	28	100%
Excite:ニュース	10	104	26	100%
Excite:ミュージック	23	104	81	100%
Yahoo!weather	75	51	14	78.6%
Yahoo!Finance	129	37	14	100%
washingtonpost	14	55	9	11.1%
cdnow	5	117	75	98.7%
amazon	5	149	50	100%

表 2: 各 Web サイトごとの抽出項目数および再現率

3 極大共通生垣 (maximal common hedge, MCH)

前節まで単純で正則な生垣のクラスの極小共通汎化を考えてきたが, 実験の結果, このクラスの極小共通汎化は情報抽出のための共通パターンとして適さない場合があった. そこで本節では生垣のクラスを正則なクラスに拡張し, このクラスでの極小共通汎化を考える. 本節では正則な生垣に対する極小共通汎化として極大共通生垣を定義し,MCH を求めるアルゴリズムを示す.

極大共通生垣は最長共通部分列 (longest common subsequence, LCS) を生垣に拡張したものである. 最長共通部分列とは与えられた二つの文字列 $A = a_1 a_2 \cdots a_n$ と $B = b_1 b_2 \cdots b_m$ に共通する部分文字列のうち, その長さが最大の部分文字列のことである. 例えば二つの生垣 $A = a\ b\ c\ d\ c$ と $B = a\ b\ d\ c\ a\ e$ に対して, A と B の最長共通部分列 $a\ b\ d\ c$ である. また, 文字列 A と B で一致しなかった文字の部分を変数に置き換えた列 $a\ b\ *\ d\ *\ c\ *$ を正則パターンと呼ぶ. 極大共通生垣は二つの生垣が与えられたとき, 生垣に含まれる名前が持つ子供に対しても正則パターンを求めたものである. 極大共通生垣の例を以下に示す.

例 3.1 例 2.1 の HTML 文書 D_1 と HTML 文書 D_3 について D_1 と D_3 の極大共通生垣 $mch(D_1,\ D_3)$ と, 単純で正則なクラスでの極小共通汎化 P を以下に示す.

インターネット・通信技術関連

タイトル	著者名	出版社	値段
最新 TCP/IP ハンドブック	若林宏	秀和システム	2400 円
情報流通アプリケーション技術	小谷昭	電気通信協会	2000 円

$D_3 =$
TABLE (
 CAPTION (インターネット・通信技術関連)
  TR (TD (タイトル) TD (著者名) TD (出版社)
   TD (値段))
  TR (TD (最新 TCP/IP ハンドブック) TD (若林宏)
   TD (秀和システム) TD (2400 円))
  TR (TD (情報流通アプリケーション技術)
   TD (小谷昭) TD (電気通信協会) TD (2000 円)))

$mch(D_1, D_3) =$
TABLE (
 CAPTION (Y_1 関連)
  TR (TD (タイトル) TD (著者名) Y_2 TD (値段))
  TR (TD (最新 Y_3) TD (Y_4) Y_5 TD ($Y_6$0 円))
  TR (TD (Y_7 アプリケーション Y_8)
   TD (Y_9) Y_{10} TD (Y_{11}00 円)))

$P =$
TABLE (
 CAPTION (X_1 関連)
  TR (TD (タイトル) TD (著者名) TD (X_2) X_3)
  TR (TD (最新 X_4) TD (X_5) TD (X_6) X_7)
  TR(TD (X_8) TD (X_9) TD (X_{10}) X_{11}))

図 3: 極大共通部分生垣の例

例 3.1 の極大共通生垣 $mch(D_1,\ D_3)$ は生垣 P と異なり, "TD (Y_7 アプリケーション Y_8)"の部分に二つの変数を含んでいることがわかる.

極大共通生垣の定義を与える前に, まず生垣に対する**スコア**を定義する. Σ を文字の集合, N を名前の集合 (Σ と N は互いに素), X を変数の集合, H を生垣の全体集合, R を正の整数とする. また, k, l を正の定数とする.

定義 3.1 スコア関数 $score{:}H \longrightarrow R$

1. $score(\varepsilon) = 0$
2. $score(a) = k,\ a \in \Sigma$
3. $score(x) = 0,\ x \in X$
4. $score(n(u)) = l + score(u),\ n \in N, u \in H$
5. $score(uv) = score(u) + score(v),\ u, v \in H$

生垣のスコア (定義 3.1) を用いて極大共通生垣は次のように定義される.

定義 3.2 任意の生垣 p_1, p_2 の極小共通汎化の集合を H_{mgg} とする. スコアの最大値 S_m を

$$S_m = max(\{score(p) \mid p \in H_{ca}\})$$

としたとき**極大共通生垣** $MCH(p_1,\ p_2)$ は,

$$MCH(p_1, p_2) = \{p \mid score(p) = S_m,\ p \in H_{mgg}\}$$

である.

次に極大共通生垣を求めるアルゴリズムを与える.

アルゴリズム $MCH(A, B)$
入力 生垣 $A = a_1a_2\cdots a_m$, $B = b_1b_2\cdots b_n$
ここで a_i, b_j $(1 \le i \le m,\ 1 \le j \le n)$ は Σ あるいは X の要素, または $a_i = f(u),\ b_j = f(v)$ の形をした生垣. また, S はスコアのための配列, V は走査の際のベクトルの配列, T は極大共通生垣の要素を格納する配列である.

```
1  S[0,0] := 0; V[0,0] := NULL;
2  T[0,0] := NULL;
2  for i := 1 to m do begin
3       S[i,0] := S[i-1,0]; V[i,0] := V[i-1,0];
4       T[i,0] := T[i-1,0];
5  end;
6  for j := 1 to n do begin
7       S[0,j] := S[0,j-1]; V[0,j] := V[0,j-1];
8       T[0,j] := T[0,j-1];
9  end;
10 i := 1; j := 1;
11 while j ≤ n; do
12      while i ≤ m; do
13           if a_i = b_j かつ a_i, b_j ∈ Σ then
14                S[i,j] := S[i-1,j-1] + k;
15                V[i,j] := ↖; T[i,j] := a_i;
16           else if a_i = b_j かつ
17                     a_i = f(u) かつ b_j = f(v) then
18                (s,w) := MCH(u,v);
19                S[i,j] := S[i-1,j-1] + l + s;
20                V[i,j] := ↖; T[i,j] := w;
21           else if S[i,j-1] ≤ S[i-1,j] then
22                S[i,j] := S[i-1,j];
23                V[i,j] := ←; T[i,j] := NULL;
24           else
25                S[i,j] := S[i,j-1];
26                V[i,j] := ↑; T[i,j] := NULL;
27      end;
28 end;
29 while i > 0 かつ j > 0 do
30      if V[i,j] = ↖ then
31           C := T[i,j]C;
32           i := i-1, j := j-1;
33      else if V[i,j] = ← then
34           i := i-1;
35           if c_1 ∉ X then C := xC; //c_1 は先頭.
36      else
37           j := j-1;
38           if c_1 ∉ X then C := xC; //c_1 は先頭.
39 end;
40 if i ≠ 0 または j ≠ 0 then C := xC;
41 output (S[m,n], C); //スコアと MCH のペア.
```

アルゴリズム $MCH(A, B)$ は [11] の動的計画法 (dynamic programming) を生垣に対して拡張したものである. 動的計画法は文字列をラベルに持つ二次元配列を用いるが, アルゴリズム $MCH(A,\ B)$ では, 生垣に含まれる名前の子供に対してもこの配列を生成し, 一致しなかった部分を変数に置き換えている.

補題 3.1 アルゴリズム $MCH(A, B)$ の配列 S の要素 $S[i, j]$ は正則な生垣 $A = a_1a_2\cdots a_i$ 及び $B = b_1b_2\cdots b_j$ の共通汎化のうち最大のスコアを表す.

補題 3.2 アルゴリズム $MCH(A, B)$ によって得られる生垣 C は正則な生垣 A, B の極小共通汎化である.

定理 3.1 補題 3.1, 補題 3.2 よりアルゴリズム $MCH(A, B)$ が極大共通生垣を求めるアルゴリズムである.

極大共通生垣は与えられた二つの生垣に対して複数存在することがある. 例えば, 生垣 A =

$a\ b\ f(c\ b)\ d\ f(c)$ と $B = a\ b\ f(c\ d)$ の極大共通生垣は $P_1 = a\ b\ f(c\ X)\ Y$ と $P_2 = a\ b\ X\ f(c\ Y)$ ($a,\ b,\ c,\ d$ は文字, f は名前, $x,\ y$ は変数) の二つが存在する. アルゴリズム $MCH(A, B)$ は極大共通生垣が複数存在する場合, 左側の一致を優先した極大共通生垣を返すようなアルゴリズムである. つまりこの例では P_1 が本手法で用いる極大共通生垣となる.

アルゴリズム $MCH(A, B)$ は二つの正則な生垣 A, B の極大共通生垣を求めるものであるが, 本手法では複数の基礎生垣から求めた極大共通生垣を求める必要がある. ここで,n 個の生垣 $h_1,\ h_2,\ h_3,\ \cdots, h_n$ から同時に求めた極大共通生垣を p とする. また, 以下のように n 個の生垣について順に極大共通生垣を求め, 最終的に得られる生垣を q_{n-1} とすると,

$$\begin{aligned} q_1 &= mch(h_1,\ h_2) \\ q_2 &= mch(q_1,\ h_3) \\ q_3 &= mch(q_2,\ h_4) \\ &\vdots \\ q_{n-1} &= mch(q_{n-2},\ h_n) \end{aligned}$$

一般に, p と q_{n-1} について $p \neq q_{n-1}$ である. しかし生垣 p は計算資源の点から求めることは困難なので, 本研究では q_{n-1} を n 個の生垣から得られる極大共通生垣とした.

例 3.1 において単純で正則な極小共通汎化 P と極大共通部分生垣 $mch(D_1, D_3)$ について, それぞれ D_1, D_3 とマッチングを行うと, P と D_1 とのマッチングの結果, 変数 X_6 からは本の値段"1980 円"が抽出されるが, P と D_3 のマッチングを行うと変数 X_6 からは出版社"秀和システム"が抽出される. この例は単純で正則なクラスでは同じ変数に異なる種類の情報がマッチしてしまう例である. 一方この例において極大共通部分生垣 $mch(D_1, D_3)$ を共通パターンとしてマッチングを行うと本の値段は変数 $X_6,\ X_{10}$ から出版社は変数 $X_7,\ X_{11}$ から抽出されることがわかる.

4 極大共通生垣を用いた実験

極大共通生垣を HTML 文書の共通パターンとしてさまざまな分野の Web サイトから情報を抽出する実験を行った. 実験には表 1 と同じ Web サイトを用いた.

単純で正則なクラスの場合と同様, 表 1 のそれぞれの Web サイトごとに極大共通生垣を求め, 同一 Web サイトの HTML 文書とマッチングを行うことで情報を抽出する実験を行った. このときスコアの付け方を変えることで 3 パターンの極大共通生垣を求めた (表 3). 表 3 中の変数 k, l はスコアの定義 3.1 に現れる k, l と同じであり, それぞれ文字が一致したときのスコア, 名前が一致したときのスコアである.

タイプ	スコアの付け方
A	文字が一致: $k = 1$, 名前が一致: $l = 1$
B	文字が一致: $k = 1$, 名前が一致: $l = 10$
C	文字列が一致: $k = 1$, 名前が一致: $l = 1$

表 3: スコアの付け方

また正解集合として抽出の対象となる項目を各 Web サイトごとに選び, それらが抽出されたかどうかを再現率 (recall) を求めて評価した. ここでの抽出対象の項目は 2.3 節の項目とは異なり, 新たに選んだものである. 極大共通生垣の生成に用いた HTML 文書数, 極大共通生垣に含まれる変数の数, 抽出対象の項目数および Web サイトごとの再現率を表 4 に示す.

Web サイト名		HTML 文書数	変数の数	項目数	再現率
Yahoo!天気情報	**A**	146	126	67	92.5%
	B	146	126	67	92.5%
	C	146	101	52	94.2%
Yahoo! ファイナンス	**A**	30	41	25	68.0%
	B	30	41	25	68.0%
	C	30	25	22	100%
Yahoo! ミュージック	**A**	35	193	204	64.2%
	B	35	193	204	64.7%
	C	35	167	162	100%
goo[天気]	**A**	113	93	79	69.6%
	B	113	93	79	69.6%
	C	113	72	70	72.9%
goo[ニュース]	**A**	52	60	39	89.7%
	B	52	60	39	89.7%
	C	52	36	33	96.7%
Excite: ニュース	**A**	52	92	30	77.0%
	B	52	92	30	77.0%
	C	52	80	28	89.2%
Excite: ミュージック	**A**	29	105	80	98.8%
	B	29	105	80	98.8%
	C	29	80	60	100%

表 4: Web サイトごとの抽出項目数および再現率

表 4 において, 単純で正則なクラスで共通パターンを求めたとき情報抽出に失敗していた Yahoo!ファイナンスは極大共通生垣を共通パターンとした今回の実験では高い再現率を示していることがわかる. Yahoo!ファイナンスについて HTML 文書を図 4 に, その極大共通生垣を図 5 に示す. なお, 図 5 は極大共通生垣に含まれる変数を画像に置き換えてブラウザで表示できるようにしたものである. 図 5 を見ると, この

Web サイトで抽出対象の項目となっていた平均株価やジャスダック指数などの数字の部分が変数に置き換わっていることがわかる.

図 4: Yahoo!ファイナンスの HTML 文書

図 5: Yahoo!ファイナンスの極大共通生垣

また, 図 6 は Excite:ニュースの HTML 文書から求めた単純で正則なクラスにおける極小共通汎化 (左) と極大共通生垣 (右) である.

Excite:ニュースの HTML 文書には共通する部分があるにもかかわらず, その共通部分が変数に置き換わっていたものがあった (図 6 左). しかし, 極大共通生垣では単純で正則なクラスの極小共通汎化に比べてこの Web サイトの共通部分が多く残っていることがわかる. つまり極大共通生垣は単純で正則なクラスの極小共通汎化よりも表現能力が高いといえる.

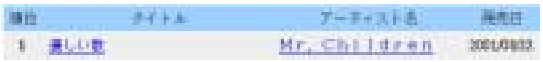
図 6: Excite:ニュースの極小共通汎化

また表 4 から, 一文字ずつ一致を見る A, B のスコアの付け方で極大共通部分生垣を求めた場合よりも, 文字列が一致したときにスコアを付ける C の場合の方が良い結果が得られていることがわかる.

図 7 は Excite:ミュージックの HTML 文書 D_1, D_2 と A タイプのスコアの付け方によって得られる極大共通部分生垣 $mch(D_1, D_2)$ である. この Web サイトでは CD のタイトルとアーティスト名が抽出の対象となっていた. これらの情報を抽出するには以下のような共通パターンが望ましい.

$$
\begin{aligned}
P = \texttt{TABLE}\ (\ &\texttt{TR}\ (\texttt{TD}\ (\text{順位})\ \texttt{TD}\ (\text{タイトル}) \\
&\quad \texttt{TD}\ (\text{アーティスト名})\ \texttt{TD}\ (\text{発売日})) \\
&\texttt{TR}\ (\texttt{TD}\ (1)\ \texttt{TD}\ (Y_1)\ \texttt{TD}\ (Y_2) \\
&\quad \texttt{TD}\ (2001/8/Y_3)))
\end{aligned}
$$

しかし, 得られる極大共通部分生垣は D_1 のアーティスト名に含まれる”re”という 2 文字と, D_2 のタイトルに含まれる”re”の一致をとる (スコアが高くなる) ため, この場合マッチングを行ってもタイトルとアーティスト名が抽出できなかった. このことが実験に用いた Web サイトすべてにおいて起っていたため, 一文字ずつ一致を見る A, B のスコアの付け方で極大共通部分生垣を求めた場合よりも, 文字列が一致したときにスコアを付ける C の場合の方が良い結果が得られている理由である.



$D_1 =$
TABLE (TR (TD (順位) TD (タイトル)
TD (アーティスト名) TD (発売日))
TR (TD (1) TD (優しい歌)
TD (Mr.Children) TD (2001/8/22)))

$D_2 =$
TABLE (TR (TD (順位) TD (タイトル)
TD (アーティスト名) TD (発売日))
TR (TD (1) TD (secret base ～ 君が ···)
TD (ZONE) TD (2001/8/08)))

$mch(D_1, D_2) =$
TABLE (TR (TD (順位) TD (タイトル)
TD (アーティスト名) TD (発売日))
TR (TD (1) X_1 TD (X_2 re X_3) X_4
TD (2001/8/X_5)))

図 7: Excite:ミュージックの HTML 文書とその MCH

5 まとめ

本稿では HTML 文書の一定の共通構造に含まれる情報を抽出するために, 複数の HTML 文書に共通するパターンを生成し, その共通パターンを利用して情報抽出を行う手法を提案した. 本手法では HTML 文書や XML 文書などの半構造化文書は生垣として扱うことができることから複数の HTML 文書をすべて生垣で表現し, 生垣の極小共通汎化を求めてそれを HTML 文書の共通パターンとした. この極小共通汎化は HTML 文書を表現した基礎生垣に対して, 個々の HTML 文書ごとに異なる部分が変数に置き換えられており, 情報抽出の対象となる HTML 文書とマッチングを行うことで極小共通汎化の中の変数とマッチした部分が抽出される. 一般に, 生垣の極小共通汎化を求めるアルゴリズムは存在しないが, 生垣のクラスを限定することで極小共通汎化を求めることは可能である. [1] において単純で正則な生垣のクラスの極小共通汎化を共通パターンとして情報抽出する手法を提案したが, 本稿では生垣のクラスを正則なクラスに拡張し, このクラスにおける極小共通汎化を利用して情報抽出を行う手法を提案した. 正則な生垣のクラスの極小共通汎化の一つとして極大共通生垣を定義した. この極大共通生垣は正則な生垣の極小共通汎化のうち最も高いスコアを持つ生垣である. また, 本手法を用いて実際の Web サイトから情報を抽出する実験を行った.

参考文献

[1] 福田 賢治, 石野明, 竹田 正幸, 松尾 文碩, "生垣上の反単一化を用いた情報抽出手法の提案", 第 53 回 人工知能学会 知識ベース研究会資料, SIG-KBS-A102-4, pp.47-52, 2001.

[2] A.Yamamoto, K.Ito, A.Ishino, and H.Arimura, "Deductive and Inductive Reasoning on Semi-Structured Documents Modelled with Hedges", *the 11th International Conference on Inductive Logic Programing*, 2001.

[3] 山本 章博, 伊藤 公人, 石野 明, "生垣論理プログラミングによる情報の抽出と変換", *The 15th Annual Conference of Japanese Society for Artificial Intelligence*, 2001.

[4] B.Courcelle, "On Recognizable Sets and Tree Automata", *Resolution of Equations in Algebraic Structures*, Academic Press, 1989.

[5] HTML Tidy, `http://www.w3.org/Raggett/tidy/`.

[6] N.Kushmerick, "Wrapper Induction for Information Extraction", *Intl. Joint Conference on Artificial Intelligence (IJCAI)*, 1997.

[7] B.Thomas, "Anti-Unification Based Learnig of T-Wrapper for information Extraction", *Proceedings of the Workshop on Machine Learning for Information Extraction*, 1999.

[8] B.Thomas, "Leaning T-Wrapper for information Extraction", *Proceedings of the Workshop on Machine Leaning in Human Language Technology*, 2001.

[9] S.Soderland, "Learning Iormation Extraction Rules for Semi-structured and Free Text", *Machine Learning*, pp.1-44.

[10] W.W.Cohen, "WHIRL:a word-based information representation language", *Artificial Intelligence*, Vol.118, pp.163-196, 2000.

[11] Cormen, Thomas H., Charles E.Leiserson, and Ronald L.Rivest, "Introduction to Algorithms", *Cambrige, MA:MIT Press*, 1990.