# HTML表データの構造認識システムとその評価

塚本 修一[†] 増田 英孝[††] 中川 裕志[‡]
[†]東京電機大学大学院工学研究科 [††]東京電機大学工学部 [‡]東京大学情報基盤センター

## 1 はじめに

近年、携帯電話やPDAなどの携帯端末からWebページをブラウズしたいという要求が急増している。しかし、現状では、PCの高解像度大画面(解像度が最低でも640×480以上)を前提として作られているページがほとんどである。携帯端末デバイスの画面解像度は年々高くなっているが、携帯端末の画面サイズは限られているために読める大きさで表示できる文字数には物理的限界がある。また、画面をスクロールさせるための操作量が増加する。さらに、Webページ上の表を表示する際に、ブラウザによって <TABLE> タグの取り扱い方や、対応するタグの種類が異なるため、その表示に問題が発生する。そこで、本研究では高解像度大画面向けに作られた既存のWebページを携帯端末でブラウズする際の表の表示に発生する問題解決のために、まず、表の項目名、項目名に対応するデータ(以下、「項目データ」と呼ぶ)の境界を同定することにより、その構造を認識するアルゴリズムを提案し、評価した。次に、このアルゴリズムを用いて表を携帯端末に適した形に自動変換して表示するシステムを実装した。

## 2 携帯端末における表示の問題点

携帯電話、PDAなどの携帯端末を用いてWebページをブラウズする際には、小画面、低解像度のためにさまざまな問題が発生する。ここでは、具体的な問題点を挙げその解決方法の提案を行う。

### 2.1 携帯端末で表を表示する際の問題点

表は、本来情報を整理し分かりやすくするために作られている。しかし、小画面低解像度の携帯端末で表をブラウズすると、逆に可読性が低下し、読み誤りが生じることがある。また使用するブラウザによって表示が異なる場合がある。図1にPCで表を含むページを表示した例を示す。解像度が高く画面サイズが大きいため、表全体を見渡すことができる。図2にPalmsOS[1]上のAvantGo[2]ブラウザ、図3にXiino[3]ブラウザで図1と同一の表を含むページを表示した例を示す。図2のAvantGoでは、罫線が表示されないために表の行と列の関係を保持することが難しい。次に、図3のXiinoでは罫線が表示されているので行と列の関係を認識できるが、小画面低解像度のために以下の問題が発生する。第1に、各セルの横幅が狭くなるためにセルデータの途中で折り返しが発生し、読み誤りを起こす可能性がある。第2に図4は、図3の画面をスクロールしたものであるが、表の項目名の部分が隠れてしまい、表の各セルが何を示すか見失ってしまう。その結果、スクロールしてページを戻さなくてはならない。表の行と列の数が大きくなればなるほどこれら2つの問題が顕著となる。また、表の <TD>,<TH> タグの colspan,rowspan オプションの値が増加すると、1つのセルデータを1画面内に収めて表示できなくなり、さらに可読性が低下する。図5にその例が顕著に表れたものを示す。

## 3 表の構造認識システム

### 3.1 システムの概要

本システムは、本質的な表[4]のみを対象する。これまでに、表の研究では言語的性質を点数化し表の表すドメインを認識する研究[5, 6, 7]があるが、複雑な表や、セルの複数に属性があるもの、また未知のドメインには対応できない。

[8]の研究で主としてタグの構造を用いてで項目名の認識を行っているが、正解率は60%程度である。これに対して、本研究ではセル間の類似度をベクトル空間法によって計算し、類似度の比を用いて、行と列の項目名と項目データを計算し区別する。表の $i$ 行 $j$ 列のセルを $Cell_{ij}$ として、各セルの $N$ 個の言語的性質 $k = 1, \ldots, N$ に対応して、その性質を持てば1、持たなければ0と値 $w_k$ を定義する。$w_k$ を要素とするベクトルを式(1)のように定義し、表のセルデータをベクトル化し、計算する。

$$\overrightarrow{Cell_{ij}} = (w_1, w_2, \cdots, w_N) \quad (1)$$

以下にベクトルの要素となる言語的性質を列挙する。今回の実験では $N$ は合計で101であり、概要を以下に示す。

- 連続データ (1次元)

  行、列を基準として、"1,2,3,…" 等のある決まった連続性を持ったデータ群を1つのベクトルとして定義する

- 句読点 (2次元)

  項目名は句読点のない簡潔な文字列で表されていることが多い。よって、句点、読点がないことを

(人)

| | 総数 | 30～39歳 | 40～49歳 | 50～59歳 | 60～69歳 | 70歳以上 |
|---|---|---|---|---|---|---|
| 総数 | 8369 | 1480 | 1660 | 1995 | 1701 | 1533 |
| 男性 | 3854 | 682 | 777 | 928 | 832 | 635 |
| 女性 | 4515 | 798 | 883 | 1067 | 869 | 898 |

図 1: PC 画面での表の表示例


図 2: AvantGo での表示例


図 3: Xiino での表示例


図 4: スクロールした時の Xiino での表示例


図 5: rowspan オプションがあるページを表示した例

それぞれ 1 つのベクトルの次元として定義する。

- 文字長 (3 次元)

  項目名は文字長が短いことが多い。文字長が 0(空白)、半角 10 文字以内、半角 11 文字以上をそれぞれベクトルの次元とした。

- 接頭辞 (15 次元)

  “第”,“平成”,“特” など 14 種の接頭辞の各々にベクトルの次元を割り当てる。

- 接尾辞 (44 次元)

  “日”,“課”,“年” など 43 種の接尾辞の各々にベクトルの次元を割り当てる。

- 単位 (17 次元)

  “kg”,“人”,“円” など 17 種の単位の各々にベクトルの次元を割り当てる。

- 特殊文字 (11 次元)

  項目名として、一定の期間を表している “～” や、備考などを示す “( ” ,“ ) ” などが使われることが多いことから、それら 11 種の各々にベクトルの次元を割り当てる。

- 文字種 (5 次元)

  文字種として、“平仮名”,“片仮名”,“漢字”,“数字”,“英語” の 5 次元を割り当てる

- テーブルタグの属性 (3 次元)

  一般的に colspan,rowspan のセル内あるいは、colspan のセルの直下、rowspan のセルの直後の表データは項目名となることが多い。ある表データが、colspan あるいは rowspan の構造中に存在するか、あるいは colspan のセルの直下、rowspan の直後のセルであれば、各々をベクトルの次元に割り当てる。これにより、colspan あるいは rowspan に関係する表データと、そうでない表データとの距離を離すことができる。

### 3.2 認識アルゴリズム

$m$ 行 $n$ 列の表の行間、あるいは列間の類似度を計算するために、まず表の $i$ 行 $j$ 列のセルを $Cell_{ij}$ として表し、同じ列の $Cell_{kj}(k \neq i)$ との類似度の平均 $Sim_{row}(i,j)$ を次式で求める。

$$Sim_{row}(i,j) = \frac{1}{m-1}\sum \frac{\overrightarrow{cell_{ij}} \cdot \overrightarrow{cell_{kj}}}{\left|\overrightarrow{cell_{ij}}\right|\left|\overrightarrow{cell_{kj}}\right|} \qquad (2)$$

ここで、$\sum$ の範囲は、$k = 1,\cdots,n$、但し、$k = i$ は除く。$\overrightarrow{cell_{ij}} \cdot \overrightarrow{cell_{kj}}$ は、$\overrightarrow{cell_{ij}}$ と $\overrightarrow{cell_{kj}}$ の内積を表し、$\left|\overrightarrow{cell_{ij}}\right|$ と $\left|\overrightarrow{cell_{kj}}\right|$ は、それぞれ $\overrightarrow{cell_{ij}}$ と $\overrightarrow{cell_{kj}}$ の絶対値を表す。したがって、$\sum$ の内側の式は、$\overrightarrow{cell_{ij}}$ と $\overrightarrow{cell_{kj}}$ の *cosine* である。図 6 で $Cell(1,1)$ と第 1 列中のセルとの類似度の計算の様子を示した。次に、第 $i$ 行のセル、即ち $Cell_{ij}(j = 1,\cdots n)$ のすべてについて $Sim_{row}(i,j)$ を計算し、その行と他の行との類似度

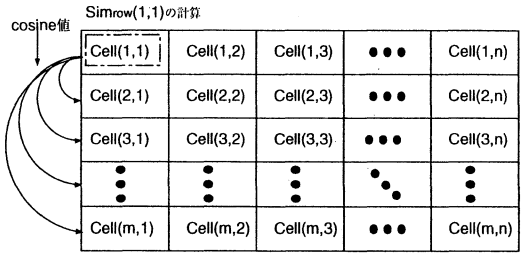


図 6: $Sim_{row}(1,1)$ の計算

の平均 $Sim_{row}(i)$ を次式で求める。

$$Sim_{row}(i) = \frac{1}{n}\sum_{k=1}^{n} Sim_{row(i,k)} \qquad (3)$$

式 (3) で計算した結果を図 7 に示す。$Sim_{row}(i)$ の値は、第 $i$ 行が、他の行と類似していれば大きく、類似していなければ小さくなる。項目名を表す行と項目データを表す行とは類似度が低い。一方、項目データを表す行同士は類似度が高い。また、項目名を表す行は Web ページでは上にくることが一般的である。$i = 1$ が第 1 行である。例えば、2 行目と 3 行目の間が項目名と項目データの境界なら図 8 のようになる。

よって、$Sim_{row}(i)$ と $i$ 行以下の $Sim_{row}(i+1), \cdots, Sim_{row}(m)$ の平均の比 $R(i)$ を、式 (4) のように定義すると、

$$R(i) = \frac{Sim_{row}(i)}{\frac{1}{m-i}\sum_{k=i+1}^{m} Sim_{row}(k)} \qquad (4)$$

1. $i$ 行が項目名、$i+1$ 行以下が項目データなら $R(i)$ は小さい

2. $i$ 行以下が全て項目データなら $R(i)$ は大きい

よって、項目名と項目データの行の境界 $T$ は次のアルゴリズムで求まる。但し、$\theta$ は、境界かどうかを判定する閾値である。

```
T = 0;
for(i=1;i<=m;i++){
    if(R(i)< θ){ T = i; }
    else{ break; }
}
if(T==0){ 縦方向に境界なし }
else{T 行までが項目名の行 }
```

以上は項目名の行と項目データの行の境界を求めるアルゴリズムだが、以上の導出において、縦横を交換すれば、$Sim_{col}(j)$ を計算でき、そして項目名の列と項目データの列の境界を認識できる。以上のアルゴリズムによって、切り出された結果から表の型に当てはめる [4]。

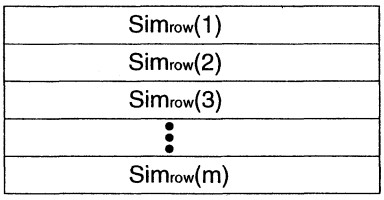


図 7: 式 (3) の計算結果

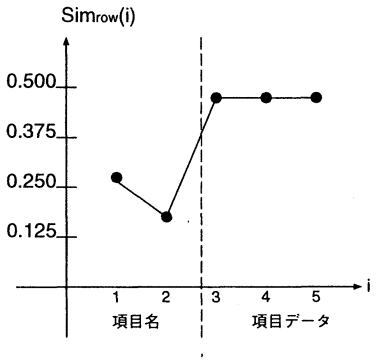


図 8: 項目名と項目データの境界における $Sim_{row}$ の変化の様子

### 3.3 認識アルゴリズムの評価実験

さて、3.2 で述べたアルゴリズムで $R(i)$ の大きさの判定に用いる閾値 $\theta$ を最適化しなければならない。これは、人手で作った正解によって実験的に決める。そこで、本アルゴリズムの評価には、$\theta$ の最適化を含め 10 fold 交差検定によって評価した。まず、最適な閾値 $\theta$ を求めるための教師データとして、Web 上にある 300 の表を人手によって項目名と項目データの行あるいは列の境界を決めた。この教師データによって 10 fold 交差検定を行った。その結果、行の閾値 $\theta$ は 0.90、列の閾値 $\theta$ は 0.70 となった。

評価の結果を表 1、2 に示す。また、評価を行った表の大きさの平均は 9.2 行 6.3 列であり、それぞれの型の個数とその内訳を表 3 に示す。

この表 3 の結果の内、時間割型となるものは 67 個あり、切れ目の内訳は 1 行目 1 列目が 44 個、2 行目 1 列目が 22 個、2 行目 2 列目が 1 個である。この結果から、システムはおよそ 80% の正解率で表の項目名を認識することができる。残りの 20% の表は項目名の部分にもかかわらず、言語的類似度がすべて高く認

表 1: 行方向の結果

| データの種類 | 正解率 |
|---|---|
| トレーニングデータ | 83.23% |
| テストデータ | 82.11% |

表 2: 列行方向の結果

| データの種類 | 正解率 |
|---|---|
| トレーニングデータ | 79.11% |
| テストデータ | 78.11% |

表 3: 交差検定によるテストデータとして評価をした 300 表の内訳

| | | 切れ目の行 (or 列) | | | | |
|---|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 | 合計 |
| 型 | 縦 | 70 | 202 | 25 | 3 | 300 |
| | 横 | 183 | 115 | 2 | 0 | 300 |

識できない表 (40%)、逆に項目データの部分にもかかわらず、言語的類似度が低い表 (60%) の 2 つに大別できる。

## 4 表示変換

3.3 で認識した項目名と項目データを携帯端末で理解しやすい形に表示した一例を図 9 に示す。変換の方針としては、常に項目名と項目データをペアで表示することにした。システムの認識結果を用いて、はじめに列の項目名の“男性”を表示し、次にそれらに付随する行の項目名と、そのペアの値を表示してあり、図 2、図 3 よりは理解しやすい。図 10 では、スクロールによって、見えなくなってしまった、項目名の“種類”、“内容”、“重量”、“料金”、とそれらペアの値を表示することにより、表の最上部にページ戻すことなく値が何を示すのか理解できる。このように項目名と項目データが認識できてしまえば、その後の表データを携帯端末等に適した形で出力することは、容易である。

## 5 まとめ

本稿では表形式データの変換のために表の項目名と項目データを切り出すシステムについて述べた。提案したアルゴリズムを適用したシステムはおよそ 80%の正解率で項目名と項目データを認識することができる。今後は、現段階では各々のベクトル要素の値は 1 か 0 としているが、実際に強く働いているものを調査し、ベクトル値を最適化し、項目名の認識率を向上させる。


図 9: 図 1 の表をシステムで変換した例

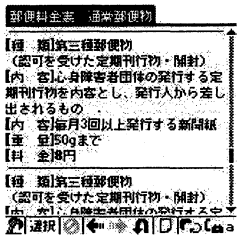


図 10: 図 5 の表をシステムで変換した例


参考文献

[1] パームコンピューティング株式会社, http://www.palm-japan.com/.

[2] AvantGo,Inc:AvantGo4.2, http://avantgo.com/.

[3] 株式会社イリンクス:Xiino2.1/SJ, http://www.ilinx.co.jp/.

[4] 塚本修一, 増田英孝, 中川裕志：HTML の表形式データの変換と携帯端末表示への応用, 第 151 回自然言語処理研究会, pp. 35–42 (2002).

[5] HURST, M. and DUGLAS, S.: Layout and Language: Preliminary Experiments in Assingning Logical Structure to Table Cells, *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pp. 217–220 (1997).

[6] 伊藤史朗, 大谷紀子, 上田隆也, 池田祐治：属性オントロジーの抽出と統合を用いた実空間と情報空間のナビゲーション システム, 人工知能学会, Vol. 14, No. 6, pp. 69–77 (1999).

[7] YOSHIDA, M.: Extracting Attributes and Their Valuse from Web Pages, *ACL-02 Student Research Workshop*, pp. 72–77 (2002).

[8] MASUDA, H., YASUTOMI, D. and NAKAGAWA, H.: How to Transform Tables in HTML for Displaying on Mobile Terminals, *6th NLPRS2001 Workshop of Automatic Paraphrasing:Theories and Applications*, pp. 29–36 (2001).