単語の属性名らしさを利用したWeb上の表の構造認識
Table Recognition Using Attribute Likelihood Computed from Words

北山 翼 * **嶋田 和孝** † **遠藤 勉** †
Tsubasa Kitayama Kazutaka Shimada Tsutomu Endo

1. はじめに

近年，Webの急速な普及により，膨大な数のWebページがWeb上に存在することとなった．これに伴い，利用者のニーズに合うページ検索サービスの要求が高まっている．Webページはテキストだけではなく，表や画像などといった要素からも構成されている．本論文では「表」に着目する．

「表」は「属性名」と「属性値」という構造を持っており，この構造は情報の関連性を表す重要な要素の一つである．しかしながら，多くのWeb検索システムでは表の構造認識を行わずにテキストとみなして処理している．このことから，表の構造を利用した検索は，システムがWeb情報を検索する際の有用な手がかりになりうると考えられる．

HTML文書中で表は<TABLE>タグを用いて表される．しかしながら，<TABLE>タグは必ずしも表として用いられているわけではない．あるドメインでは，HTML文書中で表として用いられる<TABLE>タグが全体の30 %以下であるという報告がある [1]．Chenら [1] はHTML文書から表を抽出するための手法を提案しているが，それらはヒューリスティックを用いており，データセット変更の際にヒューリスティックを選定しなおすという手間がかかる．Wangらは決定木とSVMを用いた機械学習よる表抽出を提案している [4]．この手法では表の抽出は行っているが，表の構造認識は行っていない．

本稿では，単語の属性名らしさに着目した表の構造認識方法を提案する．提案手法は，<TABLE>タグを「本物の表」・「レイアウトとして用いられている<TABLE>タグ」の2つに分類するタスクだけでなく，表の構造認識タスクまでもを同時に行うことのできるという利点がある．

2. 表の構造認識

本稿では,「本物の表」である<TABLE>タグは「属性名」と「属性値」という2つの部分から構成されると定義する．図1は「本物の表」の例である．図1の1列目が「属性名」部分であり，2列目は「属性値」部分である．

また，本稿では，表の構造認識を「表の属性名部分をシステムに正しく認識させること」と定義する．

2.1 単語の重み付け

本手法では，単語の重みを用いて表の構造認識を行う．単語の重みは学習データから計算する．この重みは単語が「属性名」・「属性値」のどちらになりやすいかを表している．重みの大きな単語は「属性名」として表に頻出する単語であり，重みの小さな単語は「属性値」として表に頻出する単語である．本手法では，表のセルごとに

価格(税抜き)	36,000円
発売日	3月23日
当初月産台数	10,000台

図 1: 本物の表の例

「属性名」・「属性値」とタグ付けを行った学習データを用い，以下の手順で単語の重みを計算する．

1. テーブルの各セルごとに文字列を抽出する．
2. 文字列を単語に分割する‡.
3. 式 (1) でそれぞれの単語ごとに重みを計算する．$P_{word}(w)$ は単語 w の属性名らしさを示している．

$$P_{word}(w) = \frac{\text{単語 } w \text{ の属性名としての出現数}}{\text{単語 } w \text{ の総出現数}} \quad (1)$$

2.2 属性名らしさ

本手法では，式 (1) で求めた単語の重みから，次のような手順で，各行および列の属性名らしさを計算する．まず，各セルの重みを求め，次に，各行および列の重みを計算する．最後に，各行および列の属性名らしさを求める．属性名らしさが閾値以上の行（列）が存在すれば，その<TABLE>タグを「本物の表」とし，その行（列）を属性名とする．属性名らしさが閾値以上となる行（列）が存在しない場合，その<TABLE>タグは「レイアウトとして用いられている<TABLE>タグ」であると判断する．アルゴリズムを以下に示す.

1. 表中の各単語の属性名らしさから，各セルの属性名らしさを求める．

$$P_{cell}(x) = \frac{\text{セル } x \text{ 中の } P_{word}\text{の総和}}{\text{セル } x \text{ 中の総単語数}} \quad (2)$$

2. 各セルの属性名らしさから，表中の全セルの属性名らしさの平均を求める．

$$P_{table} = \frac{\text{表中の } P_{cell}\text{の総和}}{\text{表の総セル数}} \quad (3)$$

3. それぞれの行もしくは列が存在しない場合の，表中の全セルの属性名らしさの平均を求める．

$$P_{row}(i) = \frac{\text{表中の行}_i\text{を除いた } P_{cell}\text{の総和}}{\text{表中の行}_i\text{を除いた総セル数}} \quad (4)$$

*九州工業大学大学院 情報工学研究科 情報科学専攻
†九州工業大学 情報工学部 知能情報工学科

‡単語の抽出には日本語形態素解析ツール「茶筌」を用いた．http://chasen.naist.jp/hiki/ChaSen/

$$P_{col}(j) = \frac{\text{表中の列}_j\text{を除いた}\ P_{cell}\text{の総和}}{\text{表中の列}_j\text{を除いた総セル数}} \quad (5)$$

もし，行$_i$ または 列$_j$ が属性名である場合は，P_{row} や P_{col} の分子の値が小さくなるため，P_{row} や P_{col} も小さくなる．すなわち，P_{row} や P_{col} が小さいほど，その行や列は属性名らしいことを表す．

4. P_{row}, P_{col} の中で最小の値 P_{min} を求め，その行または列を属性名候補とする．

$$P_{min} = min(P_{row}, P_{col}) \quad (6)$$

5. P_{table} から P_{min} を引き P_{max} を求める．P_{max} がある閾値以上であれば，属性名候補の行もしくは列のセルに「属性名」というラベル付けを行い，それ以外のセルには「属性値」とラベル付けを行う．P_{max} が閾値未満であれば，全てのセルに「属性値」とラベル付けを行う．

6. <TABLE>タグのセル中に「属性名」とラベル付けの行われたセルが存在すれば，その<TABLE>タグを「本物の表」と判断する．全てのセルに「属性値」というラベルがつけられている場合は，その<TABLE>タグは「レイアウトとして用いられている<TABLE>タグ」であると判断する．

処理の例を図 2 に示す．図中の表の数字はそのセルの P_{cell} を表す．

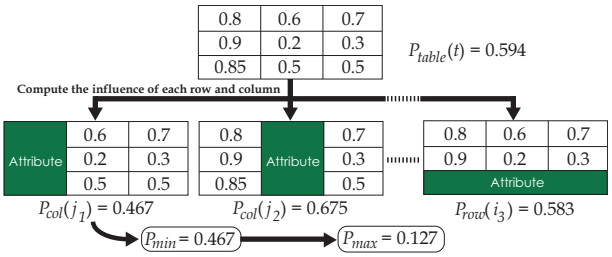

図 2: 処理の流れ

3. 実験

3.1 実験内容

提案手法の実験のために，データセットの作成をした．ファイルダウンロードソフトを用い，複数の企業ページから 10935 ページの HTML 文書を取得した．この HTML 文書中から，2 行× 2 列以上の行および列を持つ<TABLE>タグを抽出し，<A>タグ§，<IMG>タグ¶，COLSPAN・ROWSPAN 属性‖および，<TABLE>タグの入れ子構造を含まない<TABLE>タグを 3229 個選出した．この中からランダムに 1000 個を選出し，データセットとした．データセット中には「本物の表」と「レイアウトとして用いられている<TABLE>タグ」の両方が含まれている．データセットの内訳を表 1 に示す．このデータセットに対し，5 分割交差検定を行い，評価した．

§ハイパーリンクを設定するためのタグ
¶Web ページ内にイメージを配置するためのタグ
‖表中のセルを横・縦方向に結合するための属性

表 1: データセットの内訳

テーブルの種類	テーブル数
本物の表	531
レイアウト	469
計	1000

本手法では，表の構造認識に閾値を用いている．そこで，以下の式 (7) で機械的に求めた閾値を用いた場合と手動で変化させながら求めた最適な閾値を比較した．また，先行研究 [2] と比較し，その有効性を検証した．

$$Th = \frac{\text{学習データ中の単語}\ w\ \text{の属性名としての出現数}}{\text{学習データ中の単語}\ w\ \text{の総出現数}} \quad (7)$$

3.2 実験結果

本稿では，(1)「本物の表」と「レイアウトとして用いられている<TABLE>タグ」の分類，(2) 手動で閾値を変化させた場合と機械的に閾値を求めた場合の表の構造認識精度の変化という 2 つの実験を行った．

「本物の表」と「レイアウトとして用いられている<TABLE>タグ」の分類精度を表 2 に，表の構造認識精度を表 3 に示す．本物の表とレイアウトの分類の最高精度は，閾値が 0.07 のときの 90.2%であり，表の構造認識における最高精度も閾値が 0.07 のときの 85.6%であった．

表 2: 「本物の表」と「レイアウト」の分類精度

閾値	最高精度	機械的手法
	0.07	0.0753
精度	**90.2**	88.9

これら実験の結果では，機械的に求めた閾値よりも人手で求めた閾値のほうが精度がよかった．しかしながら，機械的に求めた閾値は人手で求めた最適な閾値に近い値となったため，人手で閾値を求める際の目安になると考えられる．

また，表の構造認識で最高精度となった際の詳細な実験結果を表 4 に示す．2 行目の「本物の表」の精度とは「本物の表」である<TABLE>タグを対象にして本手法を用いた際に，システムが<TABLE>タグを「本物の表」であると認識し，かつ，属性名行（列）を正しく認識した場合の精度であり，3 行目の「レイアウト」の精度とは「レイアウトとして用いられている<TABLE>タグ」を対象にして本手法を用いた際に，システムが<TABLE>タグを「レイアウト」であると認識した場合の精度である．この結果,「本物の表」の分類精度は「レイアウトとして用いられている<TABLE>タグ」の分類精度よりも低くなった．これは,「本物の表」は表のどの部分が属性名行（列）であるかまでを判別しなくてはならない点で「レ

表 3: 表の構造認識結果

閾値	閾値ごとの精度の変化							機械的手法	増田らの手法 [2]
	0.04	0.05	0.06	0.07	0.08	0.09	0.1	0.0753	
精度	72.3	80.3	84.2	**85.6**	84.2	83.1	82.1	84.7	56.1

イアウトとして用いられている<TABLE>タグ」よりも複雑な処理が必要であるためだと考えられる．

表 4: 詳細な実験結果 (閾値：0.07)

テーブルの種類	精度 (正解数/総数)
本物の表	75.3 (400/531)
レイアウト	97.1 (456/469)
計	85.6 (856/1000)

増田らの手法 [2] は，表のセル中に出現する文字列特徴（ヒューリスティック）に基づき各セルの行（列）方向の類似度を求め，これから隣り合う行（列）同士のの類似度を求めることで表の構造認識を行う手法である．この手法の精度が低くなった理由としては，手法のヒューリスティックがデータセットに依存することが挙げられる．また，この手法は 2 × 2 のような小さな表に対して有効に機能しない．これは，この手法に用いられている類似度の求め方が小さな表に対応していないためである．一方，表 5 で提案手法における表の大きさの影響について示す．表 5 からわかるように，われわれの手法は 2 × 2 の極めて小さな表・それ以上の大きさの表ともに高い精度を得ており，本手法の有効性を確認できた．

表 5: テーブルの大きさごとの精度

テーブル種類	精度 (正解数/総数)
2 × 2 のテーブル	91.9 (216/235)
2 × 2 以上のテーブル	83.7 (640/765)
計	85.6 (856/1000)

表の構造認識に失敗した典型的なケースとして,「テーブル中に同じ単語が複数回登場するテーブルタグ」があげられる．図 3 は「同じ単語が複数回登場するテーブルタグ」の例である．このようなテーブルタグを誤認識する理由は，複数回登場する単語の重みが，各行（列）の属性名らしさに過度に影響しているためだと考えられる．

提案手法の問題点は，学習データを必要とする点である．学習データを人手で作成することにはコストがかかる．このような問題を解決するために，Yoshida ら [5] は EM アルゴリズムを用いている．また，大前ら [3] は表の構造に着目して表の構造認識を行い，学習データを用いずに 77.4%の F 値を得ている．しかしながら，大前らの手法をわれわれのデータセットに適用したところ，約

製品名	A	B
対応電池種類	リチウム電池	単三アルカリ電池
連続稼働時間	15時間	10時間
価格	オープン価格	オープン価格

図 3: 同じ単語が複数回登場するテーブルタグ

30%程度の精度しか得ることができなかった．これは，大前らの手法がいくつかのキーワードで絞り込んだデータを対象としているためだと考えられる．これらの手法を参考にすることで，学習データを必要としない手法を考案することが今後の課題である．

4. おわりに

本稿では，学習データを利用して単語の重みを求めることで，<TABLE>タグを「本物の表」と「レイアウトとして用いられている<TABLE>タグ」に分類するタスクと，表の構造認識を同時に行うことのできる手法を提案した．今後の課題としては，2 つ以上の属性名行（列）を持つような表への対応や学習データを必要としない手法の導入などが挙げられる．

参考文献

[1] H. H. Chen, S. C. Tsai and J. H. Tsai: Mining tables from large scale HTML texts, Proc. of COLING2000, pp. 166-172, 2000.

[2] 増田 英孝, 塚本 修一, 安富 大輔, 中川 裕志, “HTML の表形式データの構造認識と携帯端末表示への応用”, 情報処理学会論文誌, Vol.44 No.198, 2003.

[3] 大前 信弘, 黄瀬 浩一, “Web の表を対象とした属性の自動識別”, 情報処理学会 研究報告, NL-171, pp. 43-48, 2006.

[4] Yalin Wang, Jianying Hu, “A Machine Learning Based Approach for Table Detection on The Web”, The Eleventh International World Wide Web Conference, 2002.

[5] Minoru Yoshida, Kentaro Torisawa, Jun’ichi Tsujii, “Integrating Tables on the World Wide Web”, Transactions of the Japanese Society for Artificial Intelligence, 19(6). pp. 548-560, 2004.