繰り返し構造を利用したWebページからの情報抽出

松下 京群[1*]

[1] 株式会社富士通研究所

Abstract:
ウェブページからの情報抽出においては，HTML タグの属性値やタグの繰り返し構造，タグの種類に基づく特徴などを用いた構造化が提案されている．しかし，属性値などはそのウェブページの要素が作る意味的な構造などと必ずしも対応があるわけではない．そこで本研究では，ウェブページを作成するにあたってその見た目が重要視されていると仮定し，ウェブページ内の各要素の表示座標位置とタグの繰り返し構造を活用した情報の構造化を試みた．結果として，見た目 (各タグに対応する要素の座標情報) を用いて，属性値を用いるよりも高い recall と precision を得た．

1 背景・目的

インターネット上には膨大な量の情報が公開されており，有用なデータ資源としてとらえられている．例えば，Common Crawl[1] が提供するデータを利用した様々な研究が行われている [2].

web ページからの情報抽出では，HTML タグの繰り返し情報や style 情報などを用いた手法が提案されている [4, 5].

しかし，style 情報は見た目を制御するために用いられる仕組みであり，それらは必ずしも同じ種類のデータに同じ値が割り振られているわけではない．そのため，style 情報に従って構造化を行うことで，望ましくない構造化データが得られる可能性がある．

本研究では，web ページが見た目を重要視して作成されていると仮定し，ある web ページにおいて，繰り返し出現する HTML タグの構造と，各タグに対応する要素の表示座標情報を活用した情報の構造化を試みた．ここでの「構造化された情報」とは，同じ属性の情報が同一のカラムに含まれるテーブルを指している．

2 手法

2.0.1 概要

解析対象の木 T(以降，解析木とする) が持つノード n_i (i はノード固有の ID) を根とする森を森 F_i とする．すべての森 F_i に対して，最大共通部分木 LCS_i を生成する．T 内部に含まれれる LCS_i と同形の部分木を抽出することで，共通の構造を持つ部分木の集合を得る．これらは共通の構造をもつため，テーブル形式への変換 (構造化) が可能である (つまり，LCS_i が構造化後のテーブルの 1 レコード対応することになる)．この LCS_i を用いた部分木の抽出の際に座標情報や繰り返し構造を利用し，より望ましい構造に変換することを目指す．

2.1 データの解析

2.1.1 前処理

データの前処理として以下の 4 つを行った．

- テーブル内の結合されたセルを内容を複製し分割
- タグが保有するテキストがなく，alt 属性値を持つ場合，alt 属性値をそのタグが保有するテキストとして設定
- 全てのテキストを text タグでくくる
- br タグでテキストが改行されている場合，br で分割して text タグでくくる (br タグは削除する)

2.1.2 rep 化

人が web ページを見て想定する階層構造やグループ構造と HTML タグの構造が必ずしも一致する必要はない．例えば，以下の図 1(下) ような階層構造を持ったページを作成する場合でも，h3 タグと p タグは木構造上の同じ深さに記述することができる (図 1 上)．

図 1 では上から 2 層目の h3 タグと p タグが (h3, p, p) のパターンを持って繰り返し出現しているが，(h3, p, p) の 1 つ 1 つは木構造として明示的にグルーピングされていない．このとき，LCS_i として，(h3, p, p) を一塊として含む最大共通部分木を得ることができない．

*連絡先：株式会社富士通研究所
神奈川県川崎市中原区上小田中 4-1-1
E-mail: m.kyoumoto@fujitsu.com

図 1: HTML の木構造 (一部) と web ページの例.

(つまり (h3, p, p) を含む最大共通部分木が得られず, (h3, p, p) を 1 レコードに含む構造化ができない.) そのため, これらの構造を繰り返し単位ごとに別のタグの子要素としてまとめる rep 化を行った.

図 1 を rep 化したものを図 2.1.2 に示す.

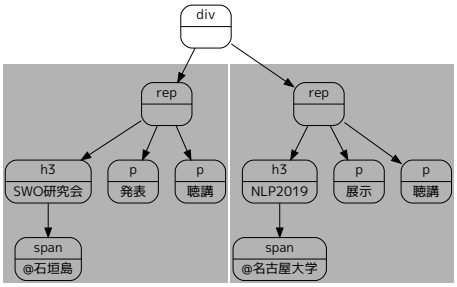

図 2: 図 1 に rep 化を行った木

.

具体的には次のような操作を行った. ある木 T の根の子 c_i を考える. ただし, $0 \leq i < N_c$, N_c は T の根の子の数である. また, $LABEL(c)$ はノード c に与えられたラベル (タグの名前やクラスの名前などから生成) を表す. この時, $2 \leq p \leq \frac{N_c}{2}$ を満たす自然数 p において,

$$LABEL(c_{i+j}) = LABEL(c_{i+p+j}) \quad (0 \leq j \leq p,\ j \in \mathbb{N})$$

を満たすようなノード列が存在する場合, それらを新たに生成した rep タグの子要素とし T からは削除した. その削除した子要素が存在していた位置のいづれかに生成した rep タグを追加した. なお, これらの操作は葉の方から, また小さい p から順次行った.

2.1.3 block 化

ある 2 つの木構造 T_i, T_j を考える. この時 2 つの木の編集距離を $TED(T_i, T_j)$ とする. この時 T_i, T_j が共通の親を持つうえで,

$$TED(T_i, T_j) \leq \log_e \left(\frac{\mathrm{size}(T_i) + \mathrm{size}(T_j)}{2} \right) \qquad (1)$$

を満たす場合, それらは同じ構造を持つと仮定した. ただし, size は T の持つノード数を返す関数である. 本研究では TED の近似値として, 対象の木構造を Euler Tour を用いて木を文字列に変換し, その文字列の編集距離を用いた. また, rep 化と同様に, $(0 \leq j \leq p,\ j \in \mathbb{N})$ の範囲でノード$\mathrm{c}_{i+j}, c_{i+p+j}$ を根とする 2 つの木が式 (1) を満たす場合, それが連続する全ての範囲を 1 つの block タグでくくった.

なお, これらの操作は葉の方から, また小さい p から順次行った.

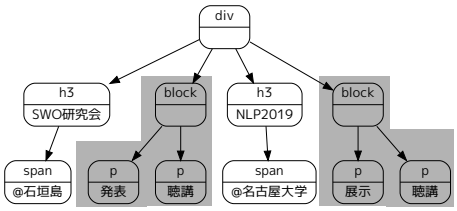

図 3: block 化の例.

2.1.4 パターンの抽出

全てのノードに対して LCS_{F_i} を求め, $3 < \mathrm{size}(LCS_{F_i}) < 20$ を満たすものをパターン T_p として抽出した.

2.1.5 構造化データの抽出

パターン T_p に従って解析木 T からの情報の構造化を行う. 基本的にはパターン T_p に一致する T の部分木を抽出するが, いくつかの条件下において, 異なる処理を加えることで構造化の改善を目指した.

blcck タグは繰り返し構造を内包し, それらを個別の木として親に渡すか, 共通の木として親に渡すかを, その block タグ以下の要素の座標情報を用いて決定する. 具体的には図 4 のように block タグを複数の木に分解し, 親と組み合わせをとる場合 (図 4 右上) か組み合わせをとらない場合 (図 4 右下) をとる.

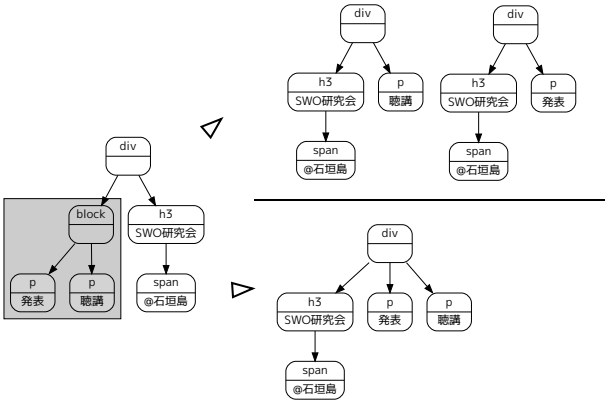

図 4: block タグの展開例.

基本的には組み合わせをとるように処理を行うが，block タグでくくられたそれぞれの要素が横並びである場合，全ての要素を同じ親ノードに結合し，新たな木構造を生成する．

また，複数の block タグが並列に存在する場合，(A1, B1), (A1, B2), ... のようにそれらのすべての組み合わせのサブツリーを生成する (図 2.1.5 右下) か，(A1, B1), (A2, B2), ... のように同じインデックスの組み合わせのみをとるかを座標情報から分類する (図 2.1.5 右上).

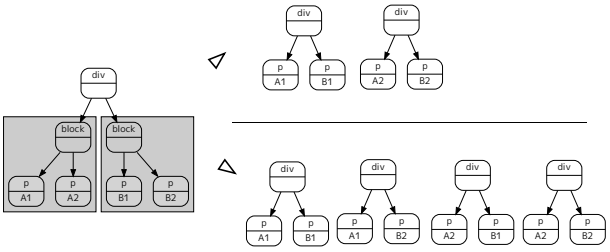

図 5: block タグが同じ層に存在する場合の展開分類

それぞれの block 内部に存在する子ノードの数が同じであり，かつ，対応する位置にある子ノードがそれぞれ横並びである場合，同じインデックス同士で結合する (図 2.1.5 右上).

2.2 座標情報による横並び要素の判断

web ページの表示領域の左上端 $(0, 0)$ を原点とし，各タグに対応する要素の表示領域の左上端を (x, y) とする．あるノードの集合のすべての要素 n について，x が異なり，y が等しい場合，それらの要素 n は横に並んでいると判断した．

3 評価

3.1 データの収集

本研究では評価用データとして，124 のアニメーションに関する web ページを対象にした．また正解データとしてしょぼいカレンダー [3] が提供している API より取得したデータを解析し，(キャラクターの名前，声優の芸名) 及び (役職，スタッフの氏名) の組を作成した．これを正解データ集合 C とする．

また，C に含まれるキャラクターの名前，声優の芸名，役職，スタッフの氏名の集合を C_a とする．また，C に含まれるキャラクターの名前，役職の集合を C_r とする．また，C に含まれる声優の芸名，スタッフの氏名の集合を C_n とする．今回解析対象とした 124 ページには全て，(キャラクターの名前，声優の芸名) または (役職，スタッフの氏名) の少なくともどちらかの組み合わせで，正解データに含まれる組が 2 組以上含まれるように選択した．(これは今回繰り返し構造を用いて抽出することを前提としているためである．) 評価に用いた正解データの組数は 1,284 であった．ただし，あるタグが保有する文字列がキャラクターの名前などと完全一致する場合に，キャラクターの名前などが web ページに含まれると判断した．

また，各タグの座標に関してはウェブブラウザ上に表示された要素の座標情報をウェブブラウザより取得し，HTML タグの属性値として付与した．座標情報はブラウザにおける表示領域の左上端 (0,0) を原点とした際の相対位置として，各タグの表示上の左上端の座標 (x, y) を取得し，使用した．なお，座標情報付与の際，ウェブブラウザのウィンドウサイズは横幅を 1920px に固定し，縦幅は対象ページのスクロール幅に合わせた状態で行った．また，動的な要素 (プルダウンメニューなど) に関しては，アクセス時に表示されていない場合，すべての座標情報を 0.0 として扱った．

3.2 評価値

解析対象とした 124 ページの各 HTML 文書 D_i に含まれる正解データを C_{D_i} とし，各ファイルに対して抽出されるべき正解データとした．D_i から本手法によって構造化された j 番目のテーブルを $M^{D_i,j}$，$M^{D_i,j}$ の k 行目を $m_k^{D_i,j}$ とする．この時，

$$\{(role, name) | role, name \in m_k^{D_i,j} \ \wedge \ role \neq name\}$$

を満たす組が C_{D_i} に存在し，かつ，$\{e | e \in m_k^{D_i,j} \setminus \{role, name\}\}$ を満たす e のいづれも，C_a に存在しない場合はその予測 $(role, name)$ を正解とした．また，D_i ごとの正解と判断された予測の集合を E_{TP_i} とした．

一方で，$\{(r', c')|r', c' \in m_k^{D_i,j}\}$ において，

$$\{(r', c')|(r' \in C_r \ \wedge \ c' \in C_n \ ) \wedge \ (r', c') \notin C_{D_i}\}$$

を満たす (r', c') を不正解とした．また，D_i ごとの不正解と判断された予測の集合を E_{FP_i} とした．さらに，すべての D_i に対して，

$$E_{TP} = \bigcup_i E_{TP_i}$$

$$E_{FP} = \bigcup_i E_{FP_i}$$

を全体の評価に用いた．

以下の表 1 にすべての D_i に対して recall などのスコアを計算し，合計した結果を示す．

各 D_i ごとの precision の平均値はどの手法においても，

表 1: 評価結果

	precision	recall
*	0.884	0.462
attrib	0.887	0.402
attrib&pos	0.881	0.408
attrib&pos&rep	0.887	0.463
attrib&rep	0.892	0.454
pos	**0.896**	0.630
pos&rep	0.881	**0.672**
rep	0.877	0.500

attrib は各タグの class の情報を含めてタグを区別したこと，pos は座標情報を用いた処理を行ったこと，rep は rep 化を行ったことを表す．

0.88 程度の数値になった．一方で各 D_i ごとの recall の平均値は座標情報を用いたものが比較的高い結果となった．また，E_{TP}, E_{FP} などを用いて recall などのスコアを計算した結果を以下の表 2 に示す．

表 2: 評価結果

	$\|E_{TP_i}\|$	precision	recall
*	779	0.696	0.607
attrib	830	0.685	0.646
attrib&pos	842	0.695	0.656
attrib&pos&rep	914	0.700	0.712
attrib&rep	899	0.690	0.700
pos	883	**0.731**	0.688
pos&rep	**950**	0.722	**0.740**
rep	842	0.696	0.656

attrib は各タグの class の情報を含めてタグを区別したこと，pos は座標情報を用いた処理を行ったこと，rep は rep 化を行ったことを表す．また $|E_{TP_i}|$ は抽出された正解データ数を示す．

すべての D_i から得られた結果 (E_{TP}, E_{FP}) で評価すると，座標情報のみを用いた場合に最も高い precision となった．また，加えて rep 化を行うことで，precision は 0.01 ポイントほど下がったが，recall が 0.05 ポイントほど上昇した．また，$M^{D_i,j}$ ごとに正解データが存在した列番号の分散を計算したところ，すべて 0.0 であった．

4 考察

今回の評価では，rep 化を行い座標情報を用いた処理を加えることで，属性値を用いたものより多くの正解データを，attrib を用いた場合と同程度以上の精度で得ることができた (表 1)．このことから，必ずしも属性値が有効に働くわけではない，もしくは属性値では構造化できない場合があると考えられる．また，座標情報を用いた場合の recall が大きいことから，属性値 (タグのクラス) がうまく機能しない場合でも，一部の構造については座標情報を用いることで構造化を行うことが可能であると考えられる．

また，各テーブル $M^{D_i,j}$ 毎の正解データが存在する列番号の分散が 0.0 であることから，正解データとして抽出された組はすべて構造化されたテーブルのレベルでは同じ列に存在していたと考えられる．

4.1 精度を下げる原因

精度を下げる原因として，本来は同じ block タグに繰り返し構造としてくくられるべき構造が，異なる block タグでくくられてしまい，必要のない組み合わせが多数生成されてしまうことがあげられる．これは，今回の手法では扱えない，繰り返し構造の間に,「先頭のページへ」などの別の要素が含まれている場合やテーブルのセルに異なる構造を持っている (例えばテーブルの同じ列にリンクが張られていたりいなかったりする) 場合などによくみられた．

4.2 評価方法について

今回の評価方法を採用したのは評価用データセットの用意が比較的容易であったことが大きな理由である．そのため，適切な評価であるか，また，他のデータセットでどのような評価が得られるかについては議論の余地があると考える．例えば，今回は無視したが，異なる構造化がきちんと行われたテーブルが存在する可能性は十分にある．一方で，実際には不要な組み合わせを持った (誤った構造化が行われた) データ多数存在する可能性もある．

4.3 他のデータへの適応例

本手法はアニメーション以外であっても web ページ内に繰り返し出現する情報を構造化することを前提としている．そのため適応例としては企業の沿革情報やプレスリリース，表彰情報など単一ページに繰り返し出現する情報の収集に役立つと考える．今回，適切な評価を行っていないが，沿革が記載された 24 の web ページについて，本手法 (pos&rep) を適応し，構造化を試みた．結果として，複数のページで年と内容を同一の列に含んだ適切だと思われるテーブルが得られた．一方で，不要な組み合わせ (年月と内容がちぐはぐなど) を生成してしまったケースも少なからず見られた．

4.4 まとめ

今回は HTML タグに対応する要素の表示上の座標位置と HTML タグの繰り返し構造を用いて，構造化された状態での情報の抽出を試みた．結果として，今回の評価では属性値を用いた場合よりも座標位置を用いた場合のほうが，高い precision と recall を得ることができた．また，今後，precision や recall を高めるうえで，rep 化や block 化の範囲の決定方法の改善やより有効な座標情報の活用が必要であると考える．

5 参考文献

参考文献

[1] Common Crawl
`http://commoncrawl.org/the-data/`
(2019 年 2 月 28 日)

[2] Common Crawl を用いた研究例
`http://commoncrawl.org/the-data/examples/`
(2019 年 2 月 28 日)

[3] しょぼいカレンダー
`http://cal.syoboi.jp/`
(2019 年 2 月 28 日)

[4] 南野 朋之，齋藤 豪，奥村 学 : 繰返し構造に基づいた Web ページの構造化，情報処理学会論文誌，vol. 45，No.9，pp.2157–2167 (2004)

[5] 沙鵬，松本章代，小西達裕，高木朗，小山照夫，三宅芳雄，伊東幸宏 : 繰り返し構造の検出に基づく Web ページの見出しの階層構造の解析，情報処理学会研究報告，Vol.2010-DD-75 No.6，pp.1–8 (2010)