Webページのテキストセグメント階層構造の抽出

松本 吉司　高橋 哲朗　乾 健太郎　松本 裕治
奈良先端科学技術大学院大学情報科学研究科
{yoshi-m, tetsu-ta, inui, matsu}@is.naist.jp

1 はじめに

World Wide Web(WWW) は膨大な情報源であり, 日常生活の様々な問題解決に使用されている. しかし, 現在 Web から情報を取得する主な手段である Google 等の検索エンジンではページ単位での検索が主流であるため, 特に情報量の多いページでは目的とする情報の正確かつ迅速な入手が困難となっている. このような状況の中で, 近年計算機が Web 上の特定の情報を収集してユーザに提示することを可能にする RSS (RDF Site Summary) 等の規定が定められている. しかし, 現在 WWW 上に RSS のようなメタデータが付与されていない HTML 文書がまだ大量に存在し, これらの文書にメタデータを人手で付与するためには膨大なコストが必要となる. このようなメタデータの付与されていない HTML 文書から階層構造を構築することができれば, ディレクトリ型検索による効率的な情報検索や, 対話的な情報検索が可能となる [5].

そこで本研究では効率的な情報検索のために HTML 文書から階層構造を抽出することを目的とする. 具体的には図 1 のような HTML 文書から図 2 に示すようなテキストセグメントの階層構造を抽出する. 階層構造の各ノードは基本的に HTML のブロックレベル要素で囲まれたテキストセグメントである. 本研究では図 2 に示すように「見出し」に注目し, 内容を含む「本文」とは区別している.「見出し」ノードは階層構造において 当該見出しノードをルートとする部分木の内容を簡潔に表したリーフ以外のノードであり, 情報検索時にインデックスとして用いることができる. また「本文」ノードはテキストセグメントの階層構造において見出しノード以外の全てのノードを指し, 子ノードを持つノードと持たないノードに分けた. 子ノードを持たない場合には情報検索の検索結果となりうる内容を含むノードとなり, また子ノードを持つ場合は「見出し」と子ノードとの関係に必要な情報付加を行うノードとなる.

2 構造化の問題点

DOM 構造

階層構造を抽出する一つの手がかりとして, 図 1 の左側に示す DOM(Document Object Model) 構造が考えられる. しかしながら, この DOM 構造はブラウザ上で表示させることを目的に作成されているため, 意味的な構造を表現するものではないことが多くの研究で指摘されている [4, 2, 1]. 例えば, 図 1 において, 「出生届」が「戸籍関係の届出」の一つの要素となっていることは DOM の構造からでは明白ではない.

また Web サイトの中には DOM 構造を意識せずに構造化を行っているページがある. 例えば図 3 のような

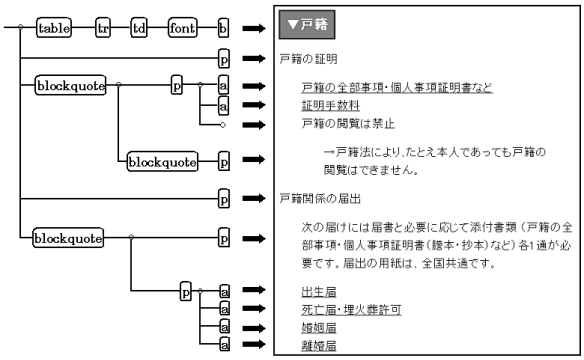


図 1: web ページの例と対応する DOM 構造 (左)

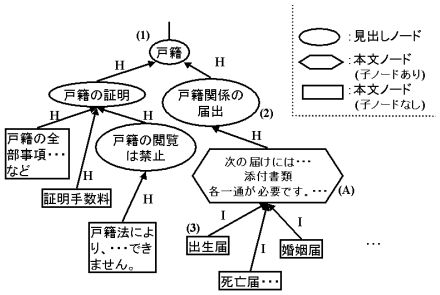


図 2: 図 1 に対するテキストセグメント階層構造

ページは「■」や「・」などのテキストで書かれた箇条書きを示す記号によって構造化していると考えられる. このページでは 2 行目の「■当サイトからのリンク」から末尾の「雑誌等商用メディア…」までのセグメントは DOM のパスが全て同一であり, DOM のパス情報のみを用いた場合これらは並列と認識されてしまうため, 適切に抽出することはできない.

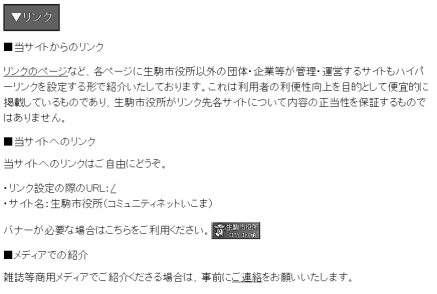
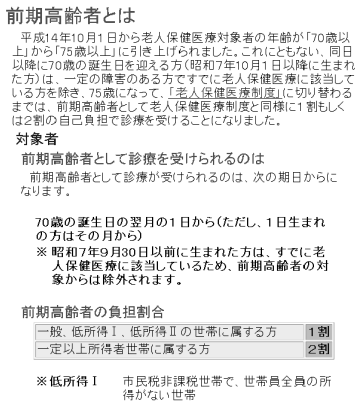


図 3: 構造化に言語的情報が必要な例

繰り返し構造

HTML の階層構造化の関連研究として, DOM 構造中の繰り返し構造 [2] や HTML 文書中の繰り返し構造

[4] を捉えて構造化を行う研究がある．ポータルサイト，トップページ，サイトマップなどでは繰り返し構造が明示的に書かれることが多いが，我々の予備調査では具体的なコンテンツが書かれたページでは繰り返しの構造が明示的でないものが多い．例えば，図 4 のようなページは，ページ中に繰り返し構造がほとんどなく，繰り返し構造を手がかりとした構造化は困難であると考えられる．このように繰り返し構造のみを用いるだけでは今回我々が目的としている階層構造の抽出は難しいため，異なるアプローチが必要となる．


前期高齢者とは

平成14年10月1日から老人保健医療対象者の年齢が「70歳以上」から「75歳以上」に引き上げられました。これにともない、同日以降に70歳の誕生日を迎える方（昭和7年10月1日以降に生まれた方）は、一定の障害のある方ですでに老人保健医療に該当している方を除き、75歳になって、「老人保健医療制度」に切り替わるまでは、前期高齢者として老人保健医療制度と同様に1割もしくは2割の自己負担で診療を受けることになりました。

対象者

前期高齢者として診療を受けられるのは

前期高齢者として診療が受けられるのは、次の期日からになります。

70歳の誕生日の翌月の1日から（ただし、1日生まれの方はその月から）

※ 昭和7年9月30日以前に生まれた方は、すでに老人保健医療に該当しているため、前期高齢者の対象からは除外されます。

前期高齢者の負担割合

一般、低所得Ⅰ、低所得Ⅱの世帯に属する方	1割
一定以上所得者世帯に属する方	2割

※低所得Ⅰ　市民税非課税世帯で、世帯員全員の所得がない世帯

図 4: 繰り返し構造が明示的でないページの例

見出しと本文の判別

我々はテキストセグメントを見出しと本文に分けることを考えているが，見出しと本文の判別は自明ではない．例えば図 1 において，「戸籍の全部事項…など」と「証明手数料」のノードは本文ノードであるが，「戸籍の閲覧は禁止」のノードは「→戸籍法により，…できません．」という本文ノードに対する見出しノードとなっている．この 3 つの要素は DOM のパスが等しいことや，インデントレベルが同じであることなどから，各ノードを単独で見ると類似した要素であるのにも関わらず，「戸籍の閲覧は禁止」のノードのみ見出しに判別するのは困難である．すなわち，単独のテキストノードに対して，見出しか本文かを分類するのは容易ではない．テキストセグメント階層構造における見出しはまとめるべき部分に対するインデックスとなっていることに注目し，当該セグメント同士の関係を見て，各セグメントの見出しと本文を判別するアプローチが適していると考えられる．

3 提案手法

テキストセグメント階層構造では，親ノードが必ず上方のノードとなることや，親子関係が交差することはないという性質があり，日本語の係り受けの構造と類似している．見出しの親子関係については親ノードができるだけ近い見出しに係ることや，モデル自身に非交差条件を考慮していること等から，スコープの小さい係り関係から優先的にボトムアップに解析する工藤ら [3] の提案手法を元に，以下の 2 種類のアルゴリズムにより解析を行った．

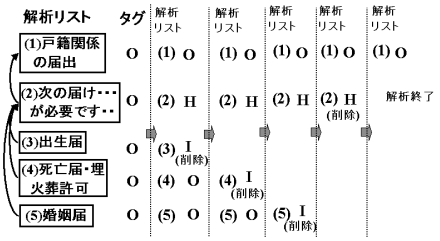


図 5: 3 値分類による構築アルゴリズムの解析例

3.1 3 値分類による構築アルゴリズム

3 値分類による構築アルゴリズムでは，当該テキストセグメントと直前のセグメントとの関係を教師あり機械学習で決定的に判定することにより，ページ全体の階層構造をボトムアップに構築する．テキストセグメントの関係は，親ノードが見出しである親子関係，親ノードが本文である親子関係，係り関係なしの 3 つに分類される．図 2 において，親ノードが見出しである親子関係を”H”，親ノードが本文である親子関係を”I”で示している．具体的な解析の流れを以下に示す．

1. 解析リスト中の全てのテキストノードに対し，係り関係が未定義の意味の”O”タグを付与する
2. ページ先頭のテキストノードを除き，入力ノードリストにおいて直前のノードとの係り関係を推定する．当該ノードは推定された係り関係タグで更新する．ここで，係り関係は直前ノードが見出し親子関係，本文親子関係，係り関係なしの 3 値に分類し，それぞれ，係り関係タグを”H”，”I”，”O”に設定する．
3. ”H”または”I”が付与されたノードについて，直後のノードが”O”タグならば解析リストから削除し，直前のノードを親に設定する．ページ末のノードに係り関係タグが付与されている場合は無条件で解析リストから削除し，直前のノードを親に設定する．
4. 解析リスト中のノードが全て”O”タグの場合は解析終了，それ以外の場合は 2. に戻る

3.2 2 段階構築アルゴリズム

2 段階構築アルゴリズムは 3 値分類による構築アルゴリズムで扱っていた問題を，「係り先の推定」と「係り関係の分類」の 2 段階で行うものである．2 段階に分けた主な理由は，3 値分類による構築アルゴリズムの手法を係り先の推定と係り関係の分類の二つに分けて評価するためである．2 段階構築アルゴリズムの第一段階として，まず当該ノードの係り先を推定する．この段階の流れは 3 値分類による構築アルゴリズムと同様である．係り先の推定では，当該ノードと直前ノードとの関係について，「係り関係あり」と「係り関係なし」の 2 値に分類する．すなわち，3 値分類による構築アルゴリズムの 2. においては，3 値に分類していたものが，ここでは「係り関係あり」と「係り関係なし」の 2 値に分類する．係り関係タグはそれぞれ”D”，”O”に設定する．

2 段階目では，係り先が推定されて構築された階層構造の各辺に対して，親ノードが見出しの場合 (H) と本文の場合 (I) の分類を行う．

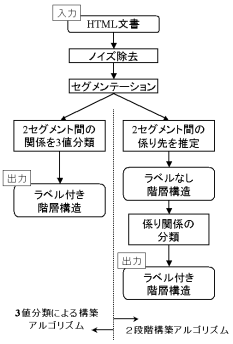


図 6: 階層構造構築実験の流れ

3.3 判定の際に用いる情報

3 値分類による構築アルゴリズムでの 2. や, 2 段階構築アルゴリズム中での係り関係推定を行う際に用いる情報としては以下の情報を用いた.

- DOM のパス
  2 節で指摘したように, DOM 構造のみにより構造化しようとすると問題が発生するが, DOM のパス情報は係り関係がない並列要素を発見する際には大きな手がかりとなる. このため, 2 つのテキストセグメントを比較する際に DOM のパスの相違の情報を用いた.
- インデント情報
  図 1 に現れているように, インデントを用いて構造化が行われているページでは, より左に詰められたセグメントが見出しとなりやすい. このようなページを適切に構造化するため, 各テキストセグメントのインデント情報を用いた.
- 言語的情報
  図 3 のように, テキストで書かれた箇条書きを示す記号を用いて構造化されたページにも対応するため, 2 つのテキストセグメントの文頭記号の相違や, 文頭記号の有無の情報を用いた. また, 見出しは簡潔なインデックスとなるセグメントであることから, 各テキストセグメントの長さ, 文末の句読点の有無の情報を用いた. その他には, 見出しの文末は名詞や助詞になりやすいという予備調査から, 文末の品詞情報も用いた.

4 実験

本節ではテキストセグメントの階層構造構築実験について述べる. 図 6 に階層構造構築実験の流れを示す. 本論文では対象として横浜市と生駒市の HTML, PHP, JSP, ASP 等のテキストファイルを用いた. 横浜市のデータは横浜市ドメインのページ 8,083 ファイルからランダムに選んだ 32 ファイル, 生駒市のデータは「生駒市 市民便利帳」のページ 328 ファイルから選んだ 91 ファイルを用いた. これらのデータに対して, 階層構造構築実験を行った流れを図 6 に示す.

4.1 セグメントの抽出

階層構造構築にあたり, Web ページからテキストセグメントの抽出を行う. セグメントの抽出はノイズ除去とセグメンテーションの 2 つの処理から成る. 本研究ではノイズの除去の際には各ページの共通するメニューバーや「トップに戻る」等の記述を半自動で除去を行った. セグメンテーションの際には, 基本単位は HTML タグのブロックレベル要素とし, 装飾などに用いられるインラインレベル要素[1]は削除した. 我々の目的は階層構造構築を全自動で行うことであるが, 今回の実験では 3 節で示したアルゴリズムの評価に焦点を当て, セグメント抽出誤りは人手で修正を行った. 最終的に抽出されたテキストセグメント数は, 生駒市で 848 セグメント, 横浜市では 493 セグメントとなった.

4.2 階層構造の構築

階層構造構築は, 3 節で示した 2 つのアルゴリズムにより行う. 3 値分類による構築アルゴリズムでは 2 セグメント間の関係を 3 値分類する段階で, また 2 段階構築アルゴリズムでは 2 セグメント間の係り先の推定と係り関係の分類で, Support Vector Machines (SVM) により分類を行う. Kernel には線形カーネルを用い, 素性には以下のものを用いた.

- 前, 当該, 次セグメントの各セグメントの素性:
  下方参照を表す語[2], 文末の句読点, 文末の品詞, 右寄せ, 中央寄せ, 文頭記号の有無
- 前セグメントと当該セグメント, 当該セグメントと次セグメントの 2 セグメントの素性:
  インデント (字下げ数) の差, DOM におけるパスの一致, 文頭記号の一致

4.3 実験結果, 考察

3 値分類による階層構築実験の生駒市と横浜市のデータに対する結果を, それぞれ表 1, 表 2 に示す. ここで, "indent"は 2 セグメント間の関係を判定する際, 当該セグメントの字下げが直前セグメントより多ければ係り関係あるというルールで判定した結果であり, "samepath"は当該セグメントの DOM のパスが直前セグメントと異なれば係り関係ありとしたルールの結果であり, 提案手法との比較に用いる. SVM-3, SVM-2 は SVM を用いて分類した結果で, それぞれ 3 節での 3 値分類によるアルゴリズムと, 2 段階構築アルゴリズムの「係り先の推定」に対応する結果である. SVM ([自治体名]) は [自治体名] のデータで学習したモデルを当該自治体ににテストした結果で, SVM (cv) は当該自治体のデータで 5 分割交叉検定を行った結果を表す. 各表の数値は係り関係正解率である[3].

まず, 他自治体で学習したモデルを適用した結果が DOM のパス情報のみを用いたルール (samepath) よりも正解率が悪くなった理由について考察する. 我々は, 見出しや本文を区別する言語的特性は対象データによらないと考えて, 見出しの特徴を表す文長や句読点の情報を分類素性に加えて考えてきたが, 見出しの言語的特性はサイト作成者によって異なることが分かった. このことは表 1 において, SVM2 (cv) が SVM2 (横浜) より

[1] 本実験では a, b, i, s, u, tt, cite, small, big, sub, sup, span, strong, strike, font を削除
[2] 今回は「次の」, 「下の」, 「下記」, 「以下の」の 4 語の有無
[3] 全体の係り関係数は生駒市: 757 関係, 横浜市: 461 関係

	正解率
indent	62.4%
samepath	57.3%
SVM-3 (横浜)	56.7%
SVM-2 (横浜)	62.9%
SVM-3 (cv)	86.1%
SVM-2 (cv)	83.1%

表 1: 係り先正解率 (生駒市)

	正解率
indent	33.8%
samepath	55.7%
SVM-3 (生駒)	44.3%
SVM-2 (生駒)	44.0%
SVM-3 (cv)	51.0%
SVM-2 (cv)	53.8%

表 2: 係り先正解率 (横浜市)

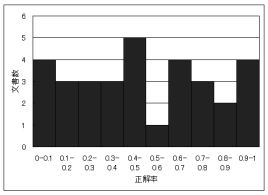


図 7: 横浜市の正解率別の文書数

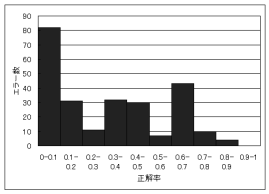


図 8: 横浜市の係り関係誤り数

も大幅によい結果となっていることからも読み取ることができる.

また, SVM で生駒市のモデルを横浜市に適用した実験の結果を正解率ごとに文書数および係り関係誤り数について集計すると図 7 および図 8 に示す結果が得られた. これらの結果から, 正解率が 10%以下の特定のページで多くのエラーが発生し, 全体の正解率を悪化させていることがわかる. エラーの原因の一つとしては, 学習に用いた生駒市のページはインデント情報を用いてページの構造化を行っていたページが多かったが, 横浜市ではインデント情報を用いて構造化したページが比較的少数だったことが挙げられる. 表 1 と表 2 において, インデント情報のみを用いた結果が生駒市の方が横浜市よりも大幅に良いことが, このことを表している.

生駒市と横浜市の両方の実験において, DOM のパス情報の取得を誤ったページでは, 係り関係の推定を誤ることが多かった. これは親が見出しの場合, セグメント抽出の際にインラインレベル要素タグを削除したために DOM のパスが同一になったことが影響している. DOM のパスが同一ならば, 係り関係がなく並列要素になる事例が多いためである.

また, 言語的情報の影響で係り関係の推定を誤った場合も多かった. 3.3 節で述べたように, 言語的情報が階層構造抽出の一つの手がかりとなるという予測から, 言語的情報を SVM の素性に用いた. しかし, DOM のパスの相違のみでは正しく推定できた係り関係を, 見出しが親である親子関係では, 誤って係り関係なしと推定された関係がある. その原因は, 文末が句読点であったり, 文長が長い等であった. この問題に対しては, 学習データを増やし, 文末が句読点であったり, 文長が長い見出しの事例を増やすことで解決できると考えられる.

その他には, 並列のノードが比較的多いページで係り先の推定誤りが多かった. 並列ノードが多いノードにおいて, 途中「係り関係なし」と誤判定されると, 解析に失敗したノード以降の兄弟ノードは全て解析に失敗することになり, 係り先の誤り数が多くなったと考えられる. これは我々の手法では解析リスト中で隣接するノード同士しか係り関係を判定しないためであると考えられる.

次に, 本文が親である係り関係の判定結果を表 3 に生駒市の結果, 表 4 に横浜市の結果を示す. それぞれ互いのデータで学習したモデルを各アルゴリズムについて適用している. 2 段階構築アルゴリズムの結果から, 係り先が既知であれば係りタイプの判定は比較的容易であることがわかる.

生駒市	recall	precision	F 値
3 値分類	61.1% (33/54)	36.3% (33/91)	0.455
2 段階	100.0% (54/54)	93.1% (54/58)	0.964

表 3: 本文が親である係り関係判定結果

横浜市	recall	precision	F 値
3 値分類	12.1% (4/33)	44.4% (4/9)	0.190
2 段階	54.5% (18/33)	100.0% (18/18)	0.706

表 4: 本文が親である係り関係判定結果

5 終わりに

本論文では Web ページを構造化するために, テキスト間の意味的な親子関係を捉えたテキストセグメントの階層構造を抽出する手法を提案した. その際まず HTML のブロック要素に基づくテキストセグメントにセグメンテーションを行った後, 当該テキストセグメントと直前のテキストセグメントとの関係を教師付き機械学習により決定的に判定することによりページ全体の階層構造をボトムアップに構築する方法を提案した. 実験の結果, 並列要素が多いページで誤りが多く発生するなどの課題を残すものの, コンテンツが書かれた繰り返し構造が明示的でないページに対しても, 本手法で構造化が可能であることが分かった.

参 考 文 献

[1] Xiao-Dong Gu, Jinlin Chen, Wei-Ying Ma, and Guo-Liang Chen. Visual based content understanding towards web adaptation. In *AH '02: Proceedings of the Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, pp. 164–173. Springer-Verlag, 2002.

[2] Yudong Yang and HongJiang Zhang. Html page analysis based on visual cues. In *ICDAR 2001*, pp. 859–864, 2001.

[3] 工藤拓, 松本裕治. チャンキングの段階適用による係り受け解析. 情報処理学会研究報告 (自然言語処理研究会), 2001-NL-142, pp. 97–104, 2001.

[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, pp. 2157–2167, 2004.

[5] 伊藤亮介, 駒谷和範, 河原達也. 機器操作マニュアルの知識と構造を利用した音声対話ヘルプシステム. 情報処理学会論文誌, Vol.43, No.7, pp. 2147–2154, 2002.