XML 文章のタグの出現位置に基づく テキストのつながりに関するタグの分類

徳田　隆志†　　田島　敬史†

† 京都大学大学院情報学研究科社会情報学専攻 〒 606-8501 京都市左京区吉田本町
E-mail: †tokuda@dl.kuis.kyoto-u.ac.jp, ††tajima@i.kyoto-u.ac.jp

あらまし　文章データを格納した XML 文書には，文章本体以外にも，その文章データに関するメタデータを記述したヘッダ部などの，文章以外のデータが混在していることが多い．このようなデータ部分では，通常，各データ項目毎にタグで囲まれて記述されているため，それらのタグの前後ではテキストは全くつながりを持たないが，一方，文章部分においては，文章を文単位に分けるために使用されるタグや，文の一部を強調するタグなどのように文の途中を分断して出現するタグなど様々なタグがあり，タグの前後のテキストのつながりの度合いも様々である．本研究では，各タグ種毎に，そのタグは前後のテキストをどのような強さで分割するものとして使われているかの判定を行う手法を開発する．

キーワード　XML，タグの分類

1. ま え が き

XML は現在様々なデータの保存形式として確実に普及しつつある．Web においては古くから RSS や XHTML など多くの規格が XML 型式を採用しており，また，最近では Microsoft の Office 製品群や ISO の標準規格である OpenDocumentFormat (ODF) などでも XML 型式をベースとしたデータ形式が採用されている．このような流れから，今後も，XML 形式で保存されるデータがさらに増加していくことが予想される．

XML 型式では，データはタグによるマークアップを含むテキストの形で記述される．そして，XML 型式で記述されたデータは，対となる開始タグと終了タグによって囲まれた範囲に対応する要素と，開始タグ中に記述される属性の二種類のノードを内部ノードとし，テキストを葉とする木構造として解釈されるのが一般的である．この時，タグによって表現される要素は，このデータのコンテンツ全体を意味的なまとまり毎に階層的に分割したものと考えられる．

さらに，各要素の意味，役割を明示するために，各タグにはタグ名が付けられる．HTML 形式が，Web ページの記述という特定のアプリケーションのための形式であることから，タグ名があらかじめ決められた集合に固定されているのに対し，XML は様々なアプリケーションのデータを統一的な形式で記述するためのものであることから，タグ名は固定されておらず，各アプリケーション毎に使用するタグ名の集合が決定される．通常，タグ名には，そのデータを作成あるいは利用するユーザにとってわかりやすいように，その役割を端的に表すような語が使用される．そのため，タグの名前を見れば，その要素の部分が何を表すデータであるのかが，ある程度推測可能な場合が多い．

このように，XML 型式では，様々なアプリケーションのデータをテキスト型式と意味のあるタグ名によって，人間にとって，あるいはそのアプリケーション以外のソフトウェアにとっても可読性の高い型式で記述する．この点は，従来の各アプリケーション専用のバイナリデータ型式が，その特定のデータ形式に対応したなんらかのソフトウェアがないとほとんど利用できないことと比べた場合の，XML 形式の大きな特徴の一つである．この特徴により，処理系の開発がバイナリ形式に比べて容易である，あるいは，あるアプリケーションのデータの，そのアプリケーション以外での再利用が容易であるなどの利点が得られる．

しかし，データが XML 型式を使用して記述されていても，そのデータを使おうとするユーザにとって，どのタグがどのような用途で使われているかがタグ名を見ただけでは推測できない場合も多く存在する．例えば，Microsoft Office Word では

- Paragraph → p タグ
- Text Run → r タグ
- Text → t タグ

など，データ量を減らすために極力文字数を減らしたタグ名が使用されているため，これらのタグ名を見ただけではその要素が何を表しているかは即座にはわからない．

あるいは，人間が見ればその意味がある程度，推測できるようなタグ名が使われている場合であっても，出現するタグ名の種類が非常に多いような場合，全てのタグ名を調べてその意味を人手で推測するのは煩雑である．同様に，XML 形式に基づく様々なアプリケーション用のデータを一括処理するような応用を考える場合も，各アプリケーションに出てくる全てのタグ名を調べてその意味を人手で推測するのは煩雑である．さらに，そもそも対応するデータ形式を特定せず，任意の XML 形式のデータを扱いたいような処理の場合は，入力となるデータ中に現れる全てのタグ名を事前に知ることはできない．

前述のように，各アプリケーション毎の専用のソフトウェアを用いなくても，ある程度データの利用が可能であるという点

が XML 形式の利点の一つであった．しかし，このようなタグ名を見ただけではその部分のデータの意味がわからないようなタグを含む XML データを使用して何かを行おうと考えた場合，結局，タグの意味がわからないために思うような処理ができないという問題がしばしば生じる．あるいは，XML 形式に基づく任意のアプリケーションデータを一括処理するような応用を考えた場合にも同様の問題が生じる．

例えば，任意の XML 形式のデータについて，二つのデータの差分を求めるような処理を考える．この場合，そのデータを順序木と考えるか非順序木と考えるかによって結果が大きく変わり得るが，XML データの中には順序木を表しているものもあれば，非順序木を表しているものもあり，さらには，一つのデータの中に兄弟間の順序に意味がある場所と意味がない場所が混在しているような場合もある．そのため，結局，タグの意味を理解しないと，もっとも望ましいような差分の計算はできないことになる．同様に，任意の XML 形式のデータに対して検索処理を行う場合，タグの意味がわかれば，その情報を様々なランキング処理や最適化処理に利用可能だが，任意の XML データのタグの意味をあらかじめ知ることはできないため，そのようなことはできないことになる．

このような問題を解決するためには，XML 形式のデータが与えられた時に，そこに現れる各タグの意味をある程度自動的に判定する技術が必要となる．しかし，そのような任意のタグの「意味」の判定は困難である．そこで，本研究では，そのような研究へと向けた第一歩として二つのことを行う．

まず一つ目としては，与えられた XML データ中に現れるタグのうち内部にテキストを持つタグについて，タグの意味までは判定しないが，そのタグのデータ全体を区切る際の役割に基づいて以下の 2 種類のタグへと分類することを考える:

- 独立したデータを囲んでいるタグ
- 文を途中で切ることがあるタグ

タグをこの二種類に分類することができれば,「文書」を含むような XML データを処理する際に，文を途中で切ることがあるタグを取り除いて，テキストノードを文単位以上のまとまったものにできるため，係り受け解析などの文を単位として行うような処理が行えたり，検索処理において複数検索語が同一の文内に現れている場合と異なる文内に現れている場合を区別できるなど，様々な有用な処理が可能となる．

このような分類を行う手法については 2. 章で詳しく説明するが，基本的な考え方としては，タグの後方のテキストの最初の品詞と局所的な形態素列より文境界を推定する方法を用いて判定を行う．このような手法によって分類を行った実験結果を 3. 章で示す．

二つ目として，タグの中で表やリストに当たる構造を構成しているものを発見することを考える．表やリストに当たる構造は多くのデータに共通して現れるものであり，これらを発見できれば XML データからの知識抽出など，様々な処理が可能になることが期待できる．また，前述のようにデータの具体的意味についてはアプリケーション毎に様々なものが使われるため，表やリスト中の各データ項目の意味については，これを自動的に判定するのは困難だが，表やリストという構造自体はほぼアプリケーションに依存しない共通の概念なため，汎用的な手法である程度発見が可能だと考えられる．表やリストの構造を見つけ出す手法については，4. 章で説明する．

2. 判 定 方 法

この章では，先ほど説明したように，内部にテキストをもつタグを独立したデータを囲んでいるタグと，文を途中で切ることがあるタグの 2 種類のタグに分類する手法について説明する．本研究では，この分類を二段階で行う．まず最初に，タグの後方のテキストの最初の品詞が助詞である確率を用いておおまかな分類を行う．しかし，このおおまかな判定では，独立したデータを囲んでいるタグの一部が，文を途中で切ることがあるタグと判定されてしまう．そこで，第二段階として，先ほどの判定で文を途中で切ることがあるタグに分類されたタグに対して，局所的な形態素列より節境界を発見する自然言語処理の手法を用いて判定を行い，独立したデータを囲んでいるタグでないかの判定を行う．以下，2.1 節で品詞の助詞を用いた判定手法の説明を行い，2.2 節で文を途中で切ることがあるタグに分類されたタグから独立したデータを囲んでいるタグを発見する判定方法について説明する．

判定方法の詳細の説明の前に，ここで考えるタグの 2 つの種類の定義について説明しておく．タグのこの 2 種類への分類は意味的なものであるため，この 2 種類を完全に形式的に定義することは困難だが，直感的にはおおよそ以下のように定義される．

- **データを囲んでいるタグ**:

内部のテキストが文として完結しており，タグの前後でテキストの文としてのつながりが低いタグ

- **文を途中で切ることがあるタグ**:

内部のテキストが文として完結しておらず，タグの前後でテキストの文としてのつながりが高いタグ

また，以下で説明する判定方法の設計に当たっては，まず，各タグの意味がわかっており，かつ，データを簡単に大量に集めることができる HTML を用いて，ここで考える 2 種類のタグが，それぞれテキストに関してどのような特性を持っているかを解析し，この結果を元に，判定方法や判定の中で使われる閾値の決定などを行った．今回の解析に使用した HTML は，Web 上の日本語の HTML 約 100 万個 (991,041 個) を 2008 年の 12 月から 2009 年の 1 月にかけて収集したものである．収集するにあたって，同じドメインからは 5 個までという上限を設けている．また，統計的手法を用いた判定方法であるため，統計情報が十分に意味を持つと考えられるような場合についてのみ判定を行うことにし，具体的には 1 万回以上出現しているようなタグについてのみ判定を行うこととした．

2.1 品詞の助詞を用いた判定

第一段階では，タグの後ろに助詞が出現する確率を用いて判定を行うが，最初に品詞の助詞を用いて判定する理由について説明する．今回の分類は，前後のテキストのつながりに基づいた分類である．よって，テキストの内容を用いた判定方法を用

いるのが，もっとも直接的であり適切な判定方法であろうと考えられる．また，文のつながりという点から見た場合，もっとも特徴的な品詞を考えると，助詞は文の構成上，決して文の初めに出現しない品詞であり，文のつながりに関する判定を行う上でもっとも信頼性が高いであろうと考えられる．

次に具体的な判定方法について説明する．上述の理由から，ここでは判定に，タグの直後のテキストの最初の品詞が助詞である確率を用いることとする．タグ x の直後のテキストの先頭に助詞が現れる確率 $P(x)$ は以下で与えられる．

$$P(x) = \frac{(\text{直後のテキストが助詞で始まる終了タグ } x \text{ の個数})}{(\text{終了タグ } x \text{ の出現数})}$$

直後のテキスト助詞で始まる場合とは下記の例のようなものである．

この部分が太字に なります
助詞

前述の収集した HTML データ中の各タグについて，その直後のテキストの最初の品詞が助詞である確率をまとめたものを表 1 に示す．なお，品詞の特定には MeCab(注1)を用いた．また，これらの HTML タグについて，人手によりここで考える 2 種類への分類を行うと以下のようになる．

- データを囲んでいるタグ:
  address, blockquote, body, caption, center, dd, div, dl, dt, fieldset, form, frame, frameset, h1, h2, h3, h4, h5, h6, head, legend, li, iframe, label, marquee, noframes, noscript, ol, option, p, pre, select, table, tbody, td, th, title, tr, script, ul
- 文を途中で切ることがあるタグ:
  a, abbr, b, big, cite, del, em, font, i, ins, kbd, rb, rp, rt, ruby, s, small, span, strong, sup, tt, u

上の分類と，表 1 に示した結果から，一般的にデータを囲んでいるタグは直後に助詞が現れる確率が低く，文を途中で切ることがあるタグは確率が高くなっていることがわかる．そこで，閾値を定め，その値より確立が低いものはデータを囲んでいるタグと判定し，そうでなければ，文を途中で切ることがあるタグと判定することにした．表 1 のデータから閾値は，0.01 とした．この結果，仮にこの HTML データに対して自動判定を行ったとした場合，62 種類のタグ中，58 種類について正しい判定を行うことができることになる．また，判定に失敗したものは，文を途中で切ることがあるタグであるのに，データを囲んでいるタグと判定された kbd, small の 2 つのタグと，データを囲んでいるタグであるのに文を途中で切ることがあるタグに判定されてた pre, blockquote の 2 つのタグの合計 4 つのタグである．

(注1)：MeCab: Yet Another Part-of-Speech and Morphological Analyzer, http://mecab.sourceforge.net/

表 1　HTML タグにおける $P(x)$ の確率の分布

x(タグの名前)	$P(x)$	x(タグの名前)	$P(x)$
rp	0.2050139	h2	0.003350282
ruby	0.1662697	kbd	0.00330891
strong	0.1146626	h5	0.003147408
rt	0.1120169	marquee	0.00309194
ins	0.08919446	h1	0.003037915
u	0.07962097	small	0.002921325
em	0.07712836	dt	0.002856133
sup	0.07203435	script	0.002448539
rb	0.06873372	noscript	0.002218781
big	0.04696006	div	0.001956423
i	0.03502707	ul	0.001889869
b	0.03355956	tbody	0.001840523
del	0.03288563	tr	0.001811366
pre	0.03162345	table	0.001766095
span	0.02725839	td	0.001714468
s	0.02398823	iframe	0.001593875
font	0.02038257	h6	0.001542905
cite	0.01627859	dd	0.001528046
tt	0.01558956	legend	0.00144818
abbr	0.01363849	dl	0.001393185
a	0.01297297	option	0.001109908
blockquote	0.01154759	form	0.001105631
p	0.006584421	frame	0.00064576
select	0.005351236	th	0.000586388
h3	0.004515915	frameset	0.000578369
ol	0.004398913	fieldset	0.000519579
head	0.004306301	label	0.000263365
center	0.004089998	caption	0.000227554
h4	0.004073665	address	0.000171801
li	0.003942301	body	0.0000357
title	0.003585954	noframes	0

2.2　局所的な形態素列による文境界推定を用いた判定

次に，文を途中で切ることがあるタグと判定されたものに対して，局所的な形態素列より文境界を推定する方法を用いた判定を行う．これにより，pre, blockquote 等の第一段階で文を切ることがあるタグとして判定されたものを，データを囲んでいるタグとして再判定できる．

この判定は，丸山らが開発した CBAP [1] [2] を用いて行う．CBAP は，判定する文章を ChaSen(注2)により形態素解析した結果を入力として，日本語の文章に含まれる節境界の位置を網羅的に検出し，その種類を特定するプログラムで，NHK のニュース原稿，日本経済新聞，バイリンガル旅行会話コーパス，旅行会話基本表現集の 5 つのコーパスで，再現率，適合率ともに 97%以上という高い節境界の判定を行うことができるものである．また，節境界とは，文の一部を構成する要素のうち，主語と独自の時制を持つ提携同士の組合せを持つ語の集まりである節と節の境界である．

判定方法は以下のようになる．データを囲んでいるタグは定義上，内部が文として終わっているはずである．そこで，CBAP を適用して，タグ名 x を持つ終了タグの前に文末が存在する確率 $Q(x)$ を求め，第一段階の手法と同様，ある閾値を定めて，

(注2)：形態素解析器茶筌, http://chasen-legacy.sourceforge.jp/

表 2　HTML タグにおける $Q(x)$ の確率の分布

x(タグの名前)	$Q(x)$
blockquote	0.16593104
pre	0.16572525
tt	0.11795533
font	0.09430823
del	0.08147347
s	0.08001636
u	0.0755045
ins	0.06717598
i	0.06583471
big	0.06563126
strong	0.05790773

x(タグの名前)	$Q(x)$
span	0.05696204
b	0.05202395
em	0.048034
cite	0.03524857
a	0.017193194
sup	0.013817565
rb	0.010560584
ruby	0.002534487
rp	0.00076226
abbr	0.000342242
rt	0.00007557

$Q(x)$ がこの閾値より高いか低いかをもとに判定を行う．

先ほどの HTML タグの判定で文を途中で切ることがあるタグと判定された各タグの $Q(x)$ の値を表 2 に示す．表 2 より，$Q(x)$ の値が 0.16 以上であるようなものを，データを囲んでいるタグとして判定し直すこととする．この結果，HTML のタグの判定は，62 種類中 kbd, small の 2 つのタグをのぞく 60 種類の判定の判定を正しく行うことができ，判定精度は 96.8%となる．

3. 実 験 結 果

この章では，先ほど紹介した手法を実際の XML データに対して適用した実験結果について説明する．今回の実験では，INEX から提供されている Wikipedia の XML コーパス [3] と Microsoft Office Word の二種類のデータを用いた．

3.1 Wikipedia の XML コーパスでの判定結果

Wikipedia の XML コーパスでの結果について説明する．前述のように，今回の手法は統計的な手法であるため，出現数が少ないタグの分類には適していない．そこで，コーパス中の出現回数が 1000 回以上である，62 種類のタグについてのみ判定を行うこととする．

これらの 62 種類のタグの人手による判定は以下のようになっている．

- データを囲んでいるタグ:
  body, cadre, caption, cell, center, conversionwarning, definitionitem, definitionlist, div, figure, image, indentation1, indentation2, indentation3, item, languagelink, name, normallist, numberlist, p, row, section, table, td, template_1, template_2, template_3, template_4, template_5, template_6, template_7, template_8,template_9, template_anotheruse, template_commons, template_genre, template_name, template_title, term, th, title, tr, value
- 文を途中で切ることがあるタグ:
  b, collectionlink, emph2, emph3, emph5, font, i, math, outsidelink, small, span, sub, sup, template_, template_ipa, template_lang, unknownlink, weblink, wikipedialink

表 3　Wikipedia タグにおける $P(x)$ の分布

x(タグの名前)	$P(x)$
math	0.3744116
wikipedialink	0.2363255
collectionlink	0.2287086
template_ipa	0.1708464
unknownlink	0.1439661
emph3	0.1375116
sub	0.1240135
i	0.09878966
sup	0.07928256
emph2	0.07435556
emph5	0.06777714
span	0.06243411
b	0.05997434
weblink	0.03866171
template_lang	0.03809524
font	0.02878041
center	0.01828012
indentation1	0.01809838
numberlist	0.01529547
outsidelink	0.01259687
small	0.00979902
normallist	0.008575876
cadre	0.008134949
table	0.006498138
template_2	0.006348193
template_commons	0.005878713
indentation2	0.003882125
section	0.003749602
td	0.003243292
template_	0.003133445
item	0.002909048

x(タグの名前)	$P(x)$
tr	0.002548149
p	0.002151276
value	0.002078371
definitionlist	0.001926266
definitionitem	0.001859859
template_5	0.001778004
div	0.001524722
title	0.001269905
template_7	0.001116487
term	0.000942266
conversionwarning	0.000928114
figure	0.000885478
cell	0.000875655
template_1	0.000837843
row	0.000774492
caption	0.000761615
template_anotheruse	0.000701754
indentation3	0.000636537
th	0.000370212
template_6	0.000259
image	0.000177096
template_4	0.000143082
template_3	0.0000949
languagelink	0.0000797
body	0
name	0
template_8	0
template_9	0
template_genre	0
template_name	0
template_title	0

この 62 種類のタグについて 2.1 章の方法を用いて判定を行った結果を，表 3 に示す．この表に示すように，62 種類中，57 種類について正しい判定を行うことができた．判定に失敗したものは，文を途中できることがあるタグであるのに，データを囲んでいるタグと判定された small, template_の 2 つのタグと，データを囲んでいるタグであるのに文を途中で切ることがあるタグに判定された center, indentation1, numberlist の 3 つのタグの合計 5 つのタグである．

続いて，2.2 章の方法をさらに用いて再判定を行った結果を表 4 に示す．この表に示すように，indentation1, numberlist の 2 つのタグがデータを囲んでいるタグとして正しく判定し直されている．

以上の結果，Wikipdia の XML コーパスでのタグの判定は 62 種類中 small, template_, center の 3 種類を除く 59 種類の判定の判定を正しく行うことができた．よって，判定精度は 95.2%となる．

3.2 Microsoft Office Word の判定結果

次に，Microsoft Office Word のデータを用いた実験結果について説明する．使用する Word のデータは，2009 年の 1 月に Web 上から取得してきた 984 個の Word ファイルを Word 2003 XML ドキュメント形式に変換したものを使用した．また，

表 4　Wikipedia タグにおける $Q(x)$ の分布

x(タグの名前)	$Q(x)$
indentation1	0.48236609
numberlist	0.3174692
emph2	0.01638641
emph5	0.01550833
center	0.013645725
font	0.013038641
wikipedialink	0.012896258
weblink	0.008178438
i	0.007523716
sup	0.006433585

x(タグの名前)	$P(x)$
b	0.00609365
outsidelink	0.003077463
span	0.002577018
emph3	0.002208309
sub	0.00190248
unknownlink	0.000984251
collectionlink	0.000376841
math	0.0001207
template_ipa	0
template_lang	0

Wikipedia のデータでの実権の際と同様の理由から，出現回数が 200 回以上のタグ 38 種類に対してのみ判定を行った．

これらの 38 種類のタグの人手による分類結果は以下のようになった．

- データを囲んでいるタグ:
  o:Author, o:Characters, o:CharactersWithSpaces, o:Company, o:Created, o:DocumentProperties, o:LastAuthor, o:LastPrinted, o:LastSaved, o:Lines, o:Pages, o:Paragraphs, o:Revision, o:Title, o:TotalTime, o:Version, o:Words, w:binData, w:body, w:p, w:pict, w:tbl, w:tc, w:tr, w:txbxContent, wx:sect, wx:sub-section
- 文を途中で切ることがあるタグ:
  aml:annotation, aml:content, w:delText w:fldData, w:hlink, w:instrText, w:r, w:rt, w:ruby, w:rubyBase, w:t

この 62 種類のタグに対して，まず，2.1 章の方法を用いて判定を行った結果を表 5 に示す．この表に示すように，38 種類中 33 種類のタグについて正しい判定を行うことができた．判定に失敗したものは，文を途中で切ることがあるタグであるのに，データを囲んでいるタグと判定された w:instrText, w:fldData, w:rt の 3 つのタグと，データを囲んでいるタグであるのに文を途中で切ることがあるタグと判定された w:binData, w:pict の 2 つのタグの合計 5 つのタグである．

続いて，2.2 章の方法を用いて再判定を行った結果を表 6 に示す．この表に示すように，aml:annotation, aml:content の 2 つのタグがデータを囲んでいるタグとして判定し直された．これらについては，正しく判定されていたものが誤って再判定されてしまう結果となった．

この結果，Word でのタグの判定は 38 種類中 w:instrText, w:fldData, w:rt, w:binData, w:pict, aml:annotation, aml:content の 7 種類を除く 31 種類の判定の判定を正しく行うことができ，判定精度は 81.6%となった．

4. 表やリスト構造の判定

この章では，HTML データ，Wikipedia の XML コーパス，Microsoft Office Word の三種類のデータから，表やリストを表すようなタグを発見する方法について説明する．まず，判定方法を 4.1 節で説明し，4.2 節で，その方法を用いて判定を行った実験結果について説明する．

表 5　Word のタグにおける $P(x)$ の分布

x(タグの名前)	$P(x)$
w:ruby	0.1094148
w:rubyBase	0.1094148
w:hlink	0.06467662
w:t	0.04696903
w:binData	0.04686192
w:r	0.04439256
aml:content	0.03322395
w:delText	0.03185361
w:pict	0.02083333
aml:annotation	0.01423591
w:tbl	0.007131537
w:instrText	0.005904288
w:p	0.005157261
wx:sub-section	0.004694836
wx:sect	0.003891051
w:txbxContent	0.002832861
w:tr	0.001882353
w:tc	0.001056662
o:Author	0

x(タグの名前)	$P(x)$
o:Characters	0
o:CharactersWithSpaces	0
o:Company	0
o:Created	0
o:DocumentProperties	0
o:LastAuthor	0
o:LastPrinted	0
o:LastSaved	0
o:Lines	0
o:Pages	0
o:Paragraphs	0
o:Revision	0
o:Title	0
o:TotalTime	0
o:Version	0
o:Words	0
w:body	0
w:fldData	0
w:rt	0

表 6　Word のタグにおける $Q(x)$ の分布

x(タグの名前)	$Q(x)$
aml:annotation	0.17943926
aml:content	0.17943926
w:delText	0.1118265
w:t	0.0755757
w:r	0.07529581

x(タグの名前)	$Q(x)$
w:pict	0.06359338
w:hlink	0
w:binData	0
w:ruby	0
w:rubyBase	0

4.1 判定方法

まず，次の条件を満たすタグを探す．

- 子孫ノードに少なくとも一つテキストノードがある
- 兄弟に自身を含めて同じ名前のタグが現れる個数の平均が 1.5 以上
- 兄弟の中でそのタグが占める割合の平均が 0.6 以上
- 本研究の判定方法でデータを囲むタグと判定される

以上の条件を満たすタグ T が見つかったら，そのような T 各々について以下の条件を満たす S を探す．

- T の親が S である確立が 0.7 以上である，すなわち，T の出現のうち 70%以上について S が親ノードとなっている．
- S の全出現の全子ノードのうち T ノードが 70%以上を占める．

これらの条件を満たすタグの組 T, S があった場合，これらをまずリストや表の構成要素となる親子関係であると判断する．次に，これらの親子関係のうちの子供側，すなわち T にあたる側になったタグを全子ノードのうちの 70%以上占めるようなタグと，最初に T と判断されたもので，T, S の親子関係となっているあるタグを親ノードとする確率が 0.7 以上であるようなタグをさらに発見し，これらの親子関係をリストや表の構成要素となる親子関係として追加する．その結果，3 種類のタグの間に，親，子，孫の三階層の関係が構成された場合は，これを表を表す構造であると判定し，二階層の親子関係にしかならない

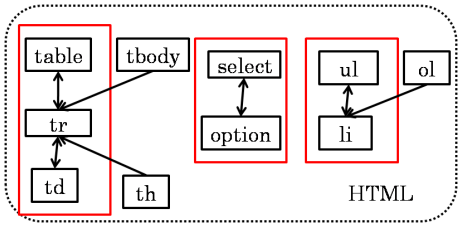

図 1　HTML の表やリスト

ものについては，これをリストを表す構造であると判定することにする．具体的な例については，次節で示す．

4.2　実験結果

先ほど説明した手法で HTML データ，Wikipedia の XML コーパス，Microsoft Office Word に対して判定を行った結果を以下で説明する．

4.2.1　HTML データでの実験結果

HTML に対して，T タグを探すと表 7 より，`div`, `dl`, `li`, `option`, `p`, `table`, `td`, `th`, `tr` が T となる．これらのタグに対して，4.1 章で説明した条件を持つ親タグを持つのは，表 8，表 9 より，(`li`, `ul`), (`option`, `select`), (`td`, `tr`), (`tr`, `table`) のペアである．これらに関連しているものをまとめると，図 1 のようになる．矢印の方向があるものは，上述の「70%以上」の条件を満たしている組である．この図より，`table`, `tr`, `td`, `tbody`, `th` が表を構成していて，`select` と `option` や `ul` と `li` と `ol` によりリストが表されていると判定することができる．

4.2.2　Wikipedia での実験結果

Wikipedia の XML コーパスに対して，T の条件を満たすタグを探すと表 10 より，`cell`, `definitionitem`, `item`, `row`, `td`, `th`, `tr` が該当する．これらのタグに対して，4.1 章で説明した条件を持つ親子関係となるのは，表 11，表 12 より，(`cell`, `row`), (`definitionitem`, `definitionlist`), (`item`, `normallist`), (`row`, `table`), (`td`, `tr`) のペアである．これらに関連しているものをまとめると，図 2 のようになる．この結果，`table`, `row`, `cell`, `tr`, `td`, `th` が表を構成し，`definitionlist`, `definitionitem` や `normallist`, `item`, `numberlist` がリストを構成していると判定することができる．

4.2.3　Microsoft Office Word での実験結果

Microsoft Office Word に対して，T の条件に該当するタグを探すと表 13 より，`w:tc`, `w:tr` が該当する．これらのタグに対して，4.1 章で説明した条件を満たす親子関係となるのは，表 14，表 15 より (`w:tc`, `w:tr`), (`w:tr`, `w:tbl`) のペアである．これらに関連しているものをまとめると，図 3 のようになる．この結果より，`w:tbl`, `w:tr`, `w:tc` により表を構成していると判定することができる．

5.　関連研究

本研究では，タグがどのような使われ方をしているかで分類を行ったが、タグごとではなく XML ごとに分類を行う研究として [4] [5] などといった研究がある。

また、XML のタグの用いられ方に関する研究としては，XML 文書中の各ノードの「キー」に関する研究 [6] や，そのようなキーを自動的に発見する手法に関する研究 [7] 等がある．

そして、表やリストなどを発見する手法を示したが、これ

表 7　HTML のタグの兄弟に関する情報

タグ名	兄弟の個数の平均	兄弟の中でそのタグの占める割合の割合
option	16.93931746	0.989423858
li	6.326068153	0.983977164
th	3.01139027	0.720754425
tr	2.863343179	0.981996332
p	2.711103473	0.649792832
dd	2.657544512	0.574915454
div	2.436539006	0.717465779
dl	2.243487202	0.665054939
td	2.096162382	0.986582381
dt	1.924868013	0.437535828
script	1.918362983	0.434528626
pre	1.589995476	0.570946637
table	1.586451612	0.748432284
h5	1.578040038	0.440990706
center	1.566924061	0.477193022
h4	1.427610608	0.392162961
h3	1.339975845	0.386801864
h2	1.339408735	0.402674855
h6	1.325971778	0.527955063
noscript	1.291255113	0.247099494
ul	1.249147517	0.642144549
fieldset	1.242794247	0.812787423
label	1.229376913	0.743343972
iframe	1.185910921	0.56272371
blockquote	1.158082465	0.395402501
small	1.151662245	0.745325648
select	1.1330599	0.506220076
ol	1.111095139	0.355100388
h1	1.093721946	0.467434133
form	1.062559256	0.620129632
marquee	1.059432023	0.498292544
address	1.017260699	0.397979099
frameset	1.011776754	0.495367772
kbd	1.009865542	0.462546688
frame	1.006009176	0.978185825
title	1.003116518	0.125921753
body	1.002233721	0.486439632
tbody	1.001736521	0.973337945
head	1.001017232	0.48019121
legend	1.00063004	0.376905485
caption	1.000371823	0.160870325

表 8　HTML のタグ T の現れる確率が最大の親に関する情報

タグ T の名前	最大の親タグ	最大の親タグの現れる確率
div	div	0.845898136
dl	div	0.851317081
li	ul	0.952805903
option	select	0.98474295
p	div	0.783519887
table	td	0.475340335
td	tr	0.994186722
th	tr	0.997971214
tr	table	0.816790434

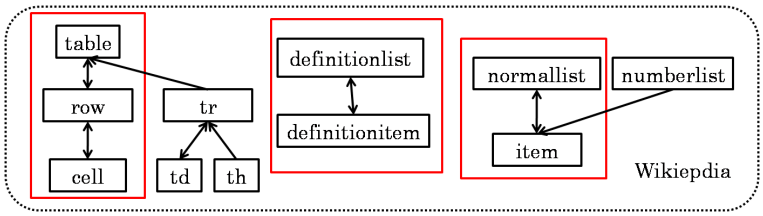

図 2　Wikipedia の表やリスト

表 9　HTML のタグ T を子ノードの全体で 0.7 以上持つもの

タグ T の名前	タグ T を子ノードのうち 0.7 以上持つタグ	子供のうち タグ T を持つ割合
li	ol	0.942058062
option	select	0.989043633
tr	table	0.883138628
tr	tbody	0.986914281
td	tr	0.941207548
li	ul	0.964739637

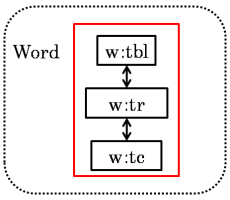

図 3　Word の表やリスト

は、表やリストといった構造となるものをマッピングしているとも考えられるため、スキーマの統合 [8] [9] [10] や構造的な類似度 [11] といったものに通じる部分がある。

また、これからの研究目的としてペアとなっているものの構造なども取り出そうと考えている。ペアの構造を発見するという分野では HTML から属性とその値を見つけるという研究 [12] [13] が存在しているが、これは構造を見つけ出すのでなく、構造をもとにしてこれら属性とその値を見つけ出すという点で違ってくる。

6. ま と め

本研究では，タグの後ろのテキストの最初の品詞が助詞である確率や局所的な形態素列より文境界を推定する方法を用いてタグを 2 種類に分類することを行った．また，表やリストといった構造を発見することも行った．

これからは，分類をもっと細かくして分類ができるようにするとともに，表やリスト以外の構造も見つけられるようにしていきたいと考えている．

謝辞　本研究は科研費 (特定領域研究「情報爆発」，18049041) の助成を受けたものである．

文　献

[1] 丸山, 柏岡, 熊野, 田中：“節境界自動検出 ルールの作成と評価”, 言語処理学会第 9 回年次大会 発表論文集, pp. 517–520 (2003).
[2] 丸山, 柏岡, 熊野, 田中：“日本語節境界検出プログラム cbap の開発と評価”, 自然言語処理, **11**, 3, pp. 39–68 (2004).
[3] L. Denoyer and P. Gallinari: “The Wikipedia XML Corpus”, SIGIR Forum (2006).
[4] M. Theobald, R. Schenkel and G. Weikum: “Exploiting structure, annotation, and ontological knowledge for automatic classification of xml data”, 6th International Workshop on the Web and Databases (WebDB-03), pp. 1–6

表 10　Wikipedia のタグの兄弟に関する情報

タグ名	兄弟の個数の平均	兄弟の中でそのタグの占める割合の割合
row	19.38652799	0.985106271
languagelink	6.63363773	0.469511976
tr	5.277614424	0.842437716
cell	4.094266881	0.999916451
item	4.010371489	0.967925418
indentation2	3.529175672	0.26534096
section	3.329864242	0.371778
td	3.277953815	0.914982039
indentation1	3.071763666	0.291646027
indentation3	2.739130435	0.222168563
definitionlist	1.737936878	0.411488345
th	1.729168688	0.638687922
definitionitem	1.533955733	1
p	1.386149291	0.188813553
template_	1.302826514	0.157561297
normallist	1.268770238	0.407268566
figure	1.111111111	0.289199228
numberlist	1.106880111	0.307549215
small	1.106853751	0.608536338
div	1.097850937	0.262435355
cadre	1.078100358	0.916020048
table	1.074859962	0.42272767
template_anotheruse	1.00422833	0.109647411
template_commons	1.003415088	0.10759678
template_1	1.00248217	0.337566089
template_8	1.000886132	0.115355071
caption	1.000707214	0.453382734
template_2	1.000598887	0.33296273
template_7	1.000368732	0.121944419
template_6	1.000256739	0.137436719
template_3	1.000094908	0.290602874
title	1	0.365608527
value	1	0.499993644
template_4	1	0.205559351
template_genre	1	0.086645265
template_name	1	0.085948132
template_title	1	0.412973916
template_9	1	0.104143564
template_5	1	0.143388788
image	1	0.5
term	1	0.5
body	1	0.333345779
conversionwarning	1	0.333333333
name	1	0.333333333

表 11　Wikipedia のタグ T の現れる確率が最大の親に関する情報

タグ T の名前	最大の親タグ	最大の親タグの現れる確率
cell	row	0.999813719
definitionitem	definitionlist	1
item	normallist	0.926410555
row	table	0.999048577
td	tr	0.853186348
th	tr	0.925025244
tr	table	0.836947724

表 12　Wikipeida のタグ T を子ノードの全体で 0.7 以上持つもの

タグ T の名前	タグ T を子ノードのうち 0.7 以上持つタグ	子供のうち タグ T を持つ割合
definitionitem	definitionlist	1
item	normallist	0.956998814
item	numberlist	0.979859004
cell	row	1
row	table	0.729005719
td	tr	0.848473076

表 13　Word のタグの兄弟に関する情報

タグ名	兄弟の個数の平均	兄弟の中でそのタグの占める割合の割合
w:tr	6.735340729	0.60742487
wx:sub-section	6.264705882	0.63531754
w:tbl	3.859327217	0.136436588
w:tc	3.841176471	0.703508458
w:p	2.758759507	0.589089651
wx:sect	2.357798165	0.148771523
aml:annotation	1.464057848	0.401803926
w:instrText	1	0.548891651
aml:content	1	1
o:Title	1	0.064408662
o:Author	1	0.064875515
o:LastAuthor	1	0.064875515
o:LastPrinted	1	0.06381378
w:rt	1	0.333333333
o:Characters	1	0.064949907
o:CharactersWithSpaces	1	0.064949907
o:Created	1	0.064949907
o:DocumentProperties	1	0.124920996
o:LastSaved	1	0.064949907
o:Lines	1	0.064949907
o:Pages	1	0.064949907
o:Paragraphs	1	0.064949907
o:Revision	1	0.064949907
o:TotalTime	1	0.064949907
o:Version	1	0.064949907
o:Words	1	0.064949907
w:body	1	0.124920996
o:Company	1	0.063651449
w:txbxContent	1	1
w:fldData	1	1

表 14　word のタグ T の現れる確率が最大の親に関する情報

タグ T の名前	最大の親タグ	最大の親タグの現れる確率
w:tc	w:tr	1
w:tr	w:tbl	1
wx:sub-section	w:body	0.523004695

表 15　Word のタグ T を子ノードの全体で 0.7 以上持つもの

タグ T	タグ T を子ノードのうち 0.7 以上持つタグ	子供のうち タグ T を持つ割合
w:tr	w:tbl	0.770940093
w:tc	w:tr	0.730474081

(2003).

[5] M. J. Zaki and C. C. Aggarwal: “Xrules: an effective structural classifier for xml data”, KDD ’03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, ACM, pp. 316–325 (2003).

[6] P. Buneman, S. Davidson, W. Fan, C. Hara and W.-C. Tan: “Keys for xml”, Proceedings of the 10th international conference on World Wide Web, pp. 201–210 (2001).

[7] G. Grahne and J. Zhu: “Discovering approximate keys in xml data”, CIKM ’02: Proceedings of the eleventh international conference on Information and knowledge management, New York, NY, USA, ACM, pp. 453–460 (2002).

[8] C. Batini, M. Lenzerini and S. B. Navathe: “A comparative analysis of methodologies for database schema integration”, ACM Comput. Surv., **18**, 4, pp. 323–364 (1986).

[9] R. J. Miller, Y. E. Ioannidis and R. Ramakrishnan: “The use of information capacity in schema integration and translation”, VLDB ’93: Proceedings of the 19th International Conference on Very Large Data Bases, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc., pp. 120–133 (1993).

[10] L. Chiticariu, M. A. Hernández, P. G. Kolaitis and L. Popa: “Semi-automatic schema integration in clio”, VLDB ’07: Proceedings of the 33rd international conference on Very large data bases, VLDB Endowment, pp. 1326–1329 (2007).

[11] A. Nierman and H. V. Jagadish: “Evaluating structural similarity in xml documents”, pp. 61–66 (2002).

[12] 吉永, 鳥澤：“Web からの具体物の属性・属性値情報の自動獲得”, 言語処理. 学会第 13 回年次大会発表論文集 (2007).

[13] M. Yoshida, K. Torisawa and J. Tsujii: “Extracting attributes and their values from web pages”, Web Document Analysis: Challenges and Opportunities (2003).