DBSJ Journal, Vol.8, No.1
June 2009

XMLタグの文書構造上の役割に関する自動分類

Classification of XML Tags According to Their Roles in Document Structure

徳田 隆志♥　田島 敬史♦

Takashi TOKUDA　Keishi TAJIMA

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．また，表やリストといった多くの文章に共通して使われる構造の発見手法も開発する．

In XML documents, in addition to text data corresponding to their main bodies, various non-sentence data, such as header sections including some meta data, often exist. In such non-sentence data sections, each data item is usually represented by a text enclosed by a pair of tags. Therefore, within such sections, two text data appearing before and after one tag are not related to each other. On the other hand, the main bodies of documents include various tags, such as tags delimiting sentences, tags forming bigger structure like paragraphs, and tags appearing in the middle of one sentence in order to annotate only parts of the sentence. Therefore, text data appearing before and after one tag in such sections have various degree of continuity. This paper proposes a method of automatically classifying XML tags according to the degree of continuity of text on their both sides. Some tags in XML documents also have special roles to represent some document structure. Among them, those representing tables and lists are commonly used in many documents. In this paper, we also propose a method to find tags representing such document structure.

♥ 学生会員　京都大学大学院情報学研究科社会情報学専攻 修士課程 tokuda@dl.kuis.kyoto-u.ac.jp

♦ 正会員　京都大学大学院情報学研究科社会情報学専攻 准教授　tajima@i.kyoto-u.ac.jp

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，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タグをpタグ，Text Runタグをrタグ，Textタグを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. 文の連続性に関する判定方法

この章では，先に説明したように，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．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．この手法は第1段階と第2段階を入れ替えても行うことができるが，入れ替えると第2段階で判定しなおす数が増えるため提示した順番で判定を行っている．

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．

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

直後のテキストが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[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た*pre*, *blockquote* の2つのタグの合計4つのタグである．

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[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のニュース原稿など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

表1 HTMLタグにおける確率*P(x)*の分布
Table 1 Distribution of probability *P(x)* for HTML tags

x(タグの名前)	*P(x)*	*x*(タグの名前)	*P(x)*
rp	0.205014	*h2*	0.00335
ruby	0.16627	*kbd*	0.003309
strong	0.114663	*h5*	0.003147
rt	0.112017	*marquee*	0.003092
ins	0.089194	*h1*	0.003038
u	0.079621	*small*	0.002921
em	0.077128	*dt*	0.002856
sup	0.072034	*script*	0.002449
rb	0.068734	*noscript*	0.002219
big	0.04696	*div*	0.001956
i	0.035027	*ul*	0.00189
b	0.03356	*tbody*	0.001841
del	0.032886	*tr*	0.001811
pre	0.031623	*table*	0.001766
span	0.027258	*td*	0.001714
s	0.023988	*iframe*	0.001594
font	0.020383	*h6*	0.001543
cite	0.016279	*dd*	0.001528
tt	0.01559	*legend*	0.001448
abbr	0.013638	*dl*	0.001393
a	0.012973	*option*	0.00111
blockquote	0.011548	*form*	0.001106
p	0.006584	*frame*	0.000646
select	0.005351	*th*	0.000586
h3	0.004516	*frameset*	0.000578
ol	0.004399	*fieldset*	0.00052
head	0.004306	*label*	0.000263
center	0.00409	*caption*	0.000228
h4	0.004074	*address*	0.000172
li	0.003942	*body*	3.57E-05
title	0.003586	*noframes*	0

[1] MeCab: Yet Another Part-of-Speech and Morphological Analyzer, http://mecab.sourceforge.net/

[2]形態素解析器茶, http://chasen-legacy.sourceforge.jp/

表2 HTMLタグにおける確率$Q(x)$の分布
Table 2 Distribution of probability $Q(x)$ for HTML tags

x(タグの名前)	$Q(x)$
blockquote	0.165931
pre	0.165725
tt	0.117955
font	0.094308
del	0.081473
s	0.080016
u	0.075505
ins	0.067176
i	0.065835
big	0.065631
strong	0.057908

x(タグの名前)	$Q(x)$
span	0.056962
b	0.052024
em	0.048034
cite	0.035249
a	0.017193
sup	0.013818
rb	0.010561
ruby	0.002534
rp	0.000762
abbr	0.000342
rt	7.56E-05

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，$Q(x)$ 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

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

表3 Wikipediaタグにおける確率$P(x)$ の分布
Table 3 Distribution of probability $P(x)$ for Wikipedia tags

x(タグの名前)	$P(x)$
Math	0.374412
Wikipedialink	0.236326
collectionlink	0.228709
template_ipa	0.170846
unknownlink	0.143966
emph3	0.137512
sub	0.124014
i	0.09879
sup	0.079283
emph2	0.074356
emph5	0.067777
span	0.062434
b	0.059974
weblink	0.038662
template_lang	0.038095
font	0.02878
center	0.01828
indentation1	0.018098
numberlist	0.015295
outsidelink	0.012597
small	0.009799
normallist	0.008576
cadre	0.008135
table	0.006498
template_2	0.006348
template_commons	0.005879
indentation2	0.003882
section	0.00375
td	0.003243
template_	0.003133
item	0.002909

x(タグの名前)	$P(x)$
tr	0.002548
p	0.002151
value	0.002078
definitionlist	0.001926
definitionitem	0.00186
template_5	0.001778
div	0.001525
title	0.00127
template_7	0.001116
term	0.000942
conversionwarning	0.000928
figure	0.000885
cell	0.000876
template_1	0.000838
row	0.000774
caption	0.000762
template_anotheruse	0.000702
indentation3	0.000637
th	0.00037
template_6	0.000259
image	0.000177
template_4	0.000143
template_3	9.49E-05
languagelink	7.97E-05
body	0
name	0
template_8	0
template_9	0
template_genre	0
template_name	0
template_title	0

表4 Wikipediaタグにおける確率$Q(x)$ の分布
Table 4 Distribution of probability $Q(x)$ for Wikipedia tags

x(タグの名前)	Q(x)	x(タグの名前)	Q(x)
indentation1	0.482366	b	0.006094
numberlist	0.317469	outsidelink	0.003077
emph2	0.016386	span	0.002577
emph5	0.015508	emph3	0.002208
center	0.013646	sub	0.001902
font	0.013039	unknownlink	0.000984
wikipedialink	0.012896	collectionlink	0.000377
weblink	0.008178	math	0.000121
i	0.007524	template_ipa	0
sup	0.006434	template_lang	0

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，Wikipediaのデータでの実験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．結果は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．結果は*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．

4.1 判定方法

まず，次の条件を満たすタグT 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，全子ノードの70%以上がタグT と判定されたあるタグとなっているタグと，最初にT と判断され，親となるS が見つからなかったもので，T, S の親子関係となった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．

まず，T タグを探すとそれぞれのデータから以下のようなタグが発見される．

HTML：*div*, *dl*, *li*, *option*, *p*, *table*, *td*, *th*, *tr*
Wikipedia：*cell, definitionitem, item, row,td, th, tr*
Word：*w:tc*, *w:tr*

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，
HTML：(li,ul), (option, select), (td,tr), (tr, table)
Wikipedia：(cell, row), (definitionitem, definitionlist), (item,normallist), (row, table), (td, tr)

Word : (w:tc, w:tr), (w:tr, w:tbl)
のペアである.

そして，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.

今回の実験で図1に書かれている表やリストの発見を行うことができた.

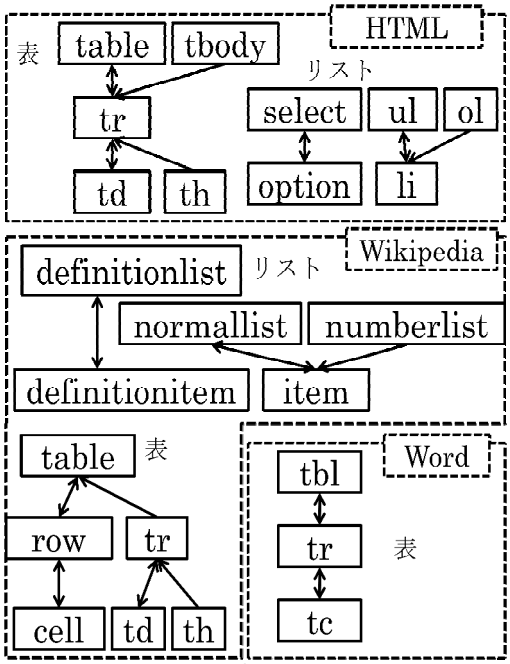

図 1 表やリスト
Fig.1 Table and list

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.

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[8][9]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異なっている.

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.
また，本研究に対して様々な有益なコメントをいただいた情報通信研究機構(NICT)の鳥澤健太郎先生に謝意を表します.

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

徳田 隆志 Takashi TOKUDA
京都大学大学院情報学研究科社会情報学専攻修士課程在学中. 2008 京都大学工学部情報学科卒業. 主にXML検索の研究に従事. 日本データベース学会学生会員.

田島 敬史 Keishi TAJIMA
京都大学大学院情報学研究科社会情報学専攻准教授. 博士(理学，東京大学). 日本データベース学会正会員