Webからの具体物の属性・属性値情報の自動獲得

吉永 直樹 †‡
†日本学術振興会
n-yoshi AT jaist.ac.jp

鳥澤 健太郎 ‡
‡北陸先端科学技術大学院大学
torisawa AT jaist.ac.jp

1 はじめに

本稿では，具体物（例: Ch. d'Yquem）とその上位語（例: ワイン）を入力として，Webから具体物の**属性-属性値の情報**（以下本稿では**属性関係**と呼ぶ，例: 産地-ボルドー）を獲得する方法を提案する．ここで属性とは，人が知りたい具体物の側面（例えばワインであれば,「造られた土地」や,「原料の葡萄の品種」）のことであり，文書中では具体的な**属性語**（例: 「生産地」,「葡萄品種」）によって参照される．

本研究では，入力された具体物とその上位語に対して，オンデマンドで属性関係の知識源となるWebページを収集し，属性関係の記述形式を各ページごとに独立に認識して属性関係を抽出する．より具体的には，1) 具体物名と上位語の属性語を利用して具体物の属性関係を含む可能性の高いWebページを収集し, 2) 収集された各ページにおいて具体物の属性関係を記述した範囲を抽出した後，3) 上位語の属性語を利用して属性関係の記述パターンを獲得し，最終的に 4) 得られたパターンを利用して具体物の属性関係を獲得する．

実験では，5個の上位語についてそれぞれ10個の具体物の属性関係の獲得を試みた．上位語の属性語は，我々が以前提案した文字強調に基づくパターン [7] を用いて，0.7TBの日本語Webリポジトリから獲得したものを用いた．実際に，具体物と上位語の属性語セットに基づくクエリによりYahoo!検索APIを用いて収集された一具体物辺り10ページから属性関係を獲得し，得られた属性関係を評価した．その結果，全具体物-上位語ペアの74%に対し，全ての属性語が適切で，かつその値に適切な属性値を含む属性関係が得られるページが存在した．

2 関連研究

本節では，既存の属性関係の獲得手法について述べる．

高橋らは語彙統語パターンを用いて文から具体物と属性関係の対応を獲得する手法を提案している [8]．彼らは信頼性の高い属性関係を得るために，具体物と属性の相互情報量，及び属性と属性値の相互情報量を用いて得られた属性関係のスコア付けを行っている．しかしながら，属性値の型（名詞句，文など）は属性に依存するため，属性語と独立した語彙統語パターンでは，値の型が複雑な属性の属性値を獲得することは本質的に難しい．また，具体物は文中で指示語で参照されることも多く，広範な属性関係を得るには具体物の照応関係を解く必要がある．

一方，特にWebを知識源として用いる際には，表形式や箇条書きなどのレイアウトが，複数の属性関係をまとめて記述する手段として用いられるため注目されている [1, 5]．Chenらは，単一の表形式を入力とし，表形式中のセル間の類似度を用いて表形式中の属性関係を認識する手法を提案している [1]．一方Yoshidaらは，表形式の集合を入力とし，各表形式中の属性関係をEM法に基づく教師なし学習により認識する手法を提案している [5]．Yoshidaはさらに，表から得られた属性関係を手がかりとしてHMMを用いて箇条書きから属性関係を抽出する手法も提案している [6]．しかしながら，表中には具体物名が必ずしも含まれるとは限らないため，具体物の属性関係を獲得するには，ページ中に含まれる具体物と表中の属性関係を関連づける必要がある．

また，HTMLタグやその階層構造，文字強調などを手がかりとしたWebからの一般的な情報抽出手法として，ラッパー学習がKushmerickらにより提案されている [3, 2]．彼らはCGI等により一貫性のある形式で出力されたページ群を情報抽出の知識源とし，その一部のページに人手で情報抽出の対象となる文字列（例えば属性関係）をタグ付けし，情報抽出のためのパターン（ラッパー）を学習する．教師付きデータの作成コストの観点から考えると，ラッパー学習に基づくアプローチでは，本研究で想定するような不特定多数によって様々な形式で書かれた具体物の属性関係を獲得することは難しい．

本研究では，Webを属性関係の知識源として，Chenらや Yoshidaらと同様に、表形式や箇条書きなどのレイアウトから属性関係の獲得を試みる．この際，具体物名の上位語の属性語を具体物名と合わせて検索エンジンのクエリとすることで属性関係を記述するレイアウトが含まれている可能性の高いWebページを収集する．さらにページ中に含まれる具体物名との（ページのDOMツリー中における）位置関係により，ページ中で具体物名に関する属性情報を含む可能性が高いHTMLタグで囲まれた範囲（**属性情報記述ブロック**）を抽出する．得られた属性情報記述ブロック中での属性関係の記述形式は，ブロック中に含まれる上位語の属性語を手がかりとして認識する．

3 提案手法

本節では，提案手法について詳しく述べる．図1は，提案手法の処理の流れを示したものである．与えられた具体物名とその上位語に対し，我々は以下の手順で属性関係を獲得する．

ステップ 1: 具体物の属性情報記述ページの収集 具体物名とその上位語，及び上位語の属性語を利用して，全文検索エンジンにより具体物の属性関係が記述されている可能性の高いWebページを収集する．

ステップ 2: 具体物の属性情報記述ブロックの抽出 ステップ1で獲得された各ページをDOM解析し，

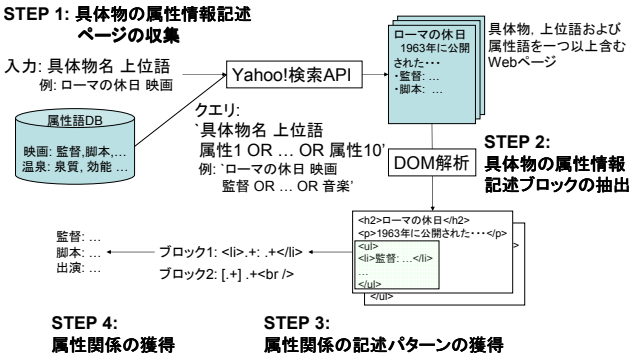

図 1: Web からの属性関係の抽出: 処理の流れ

1. “X(の)Y”, ただし, $Y \in$ { カタログ, ガイド, リスト, 紹介, 感想, 名鑑, 番付, 案内, 一覧, ランキング, データ, 目録, 図鑑, 詳細, 概要 }
2. X について
3. X

図 2: 上位語 X の属性語獲得の知識源の Web ページを集めるために用いたクエリ

ページ中で，入力の具体物の上位語の属性語を含み，具体物の属性関係の書かれている可能性の高い範囲（**属性情報記述ブロック**）を抽出する．

ステップ 3: 属性関係の記述パターンの獲得 ステップ 2 で収集された各ブロックについて，具体物の上位語の属性語の出現する文脈を手がかりとして属性関係の記述パターンを獲得する．

ステップ 4: 属性関係の獲得 ステップ 3 で獲得された属性関係の記述パターンを利用して，各ブロックから属性関係を獲得する．

以下の節で，まず上位語の属性語の獲得について述べた後，各ステップについて順に詳しく述べる．

3.1 上位語の属性語データベース獲得

本研究では，我々が以前提案した上位語の属性語獲得手法 [7] を用いて，様々な上位語に対する属性語データベースを構築する．

[7] では，入力の上位語に対し，まず全文検索エンジンを用いて属性語獲得の知識源の Web ページを収集する．次に，HTML タグあるいは文字修飾で強調された単語を属性語候補として獲得する．その後，サイト頻度と呼ばれる統計量を用いて，獲得された属性語候補をフィルタリングする．属性語候補のサイト頻度は，大ざっぱに言えば，その属性語候補が獲得できたページを，ページの Web 製作者ごとにまとめて得られた集合（Web サイト）の異なり数（Web 制作者の数）である．手法の詳細は，文献 [7] を参照されたい．

本研究では，属性語獲得の知識源として，単純に図 2 の文字列をページタイトル（title タグに囲まれた文字列中）に含むページ最大 10,000 ページを，0.7TB の日本語 Web リポジトリから収集した．次に，得られたページ中で表 1 の HTML タグおよび文字修飾により強調されていた単語を属性語候補として獲得した．属性語候補のサイト頻度は，Web ページをその URL のホスト名ごとにまとめたもの（例えば，http://host/dir/file.html のホスト名は host）を一つの Web サイトとみなして計算し，各上位語についてサイト頻度の上位 10 語を属性語として獲得した．

表 1: 上位語の属性語獲得に用いた HTML タグと文字修飾

HTML タグ:	dt, li, td, th, h1, h2, h3, h4, h5, h6, em, strong, tt, i, span, b, big, small, font, a
接頭修飾:	●, ○, ■, □, ・, ◆, ◇, ★, ☆, ◎, ＊
括弧類:	〔-〕, 【-】, 《-》, 〈-〉, <->, [-], ≪-≫, ⌊-⌋, (-)
接尾修飾:	…, ：, ／, ＼, ＝, ｜, ┃, ー

```
title, body, p, hr, h1, h2, h3, h4, h5, h6, ul,
ol, li, pre, dl, dd, dt, div, table, caption,
tr, td, th, blockquote, fieldset, address,
noscript
```

図 4: DOM 解析の際考慮した block タグ

3.2 ステップ 1: 具体物の属性情報記述ページの収集

入力の具体物に関する属性関係を高精度で効率良く獲得するために，本研究では具体物の属性関係が書かれている可能性の高いページを全文検索エンジンにより収集する．具体的には，入力の具体物名，その上位語，および節 3.1 の手法で得られた上位語の属性語 10 語のうち少なくとも一つが含まれるページを Yahoo!検索 API[1] を用いて 10 件獲得する[2]．これらのページから，次のステップで具体物の属性関係の書かれている可能性の高い箇所を同定する．

3.3 ステップ 2: 具体物の属性情報記述ブロックの抽出

前節で獲得された Web ページは，具体物名，上位語及び属性語が一つ以上含まれるが，属性語が入力の具体物に対する属性語として出現している保証が無いため，具体物と無関係な属性情報を含み得る．そこで本研究では，以下のような仮説を立てた．

仮説: Web ページ中で具体物の属性関係が記述される範囲は，上位語の属性語を含む Web ページの DOM ツリーの葉ノードで，具体物名を含む葉ノードよりも右にあり，かつ最も近いノードである．

ここで DOM ツリーの葉ノードは，Web ページ中で HTML タグに囲まれた文字列である．通常 DOM ツリーには，段落や表形式・箇条書きなどの個々のレイアウトに対応する block タグと，文中での文字修飾等に用いられる inline タグの両方をノードとする木構造であるが，本研究では，属性語を強調するパターンに inline タグが多く用いられることから，図 4 に挙げた block タグのみを考慮して，擬似的な DOM ツリーを構築した（図 3 右参照）．この仮説を元にして，各 Web ページの DOM ツリーを探索して得られる，具体物名に関する記述が含まれている可能性が高い HTML タグ囲まれた文字列（**属性情報記述ブロック**）を抽出する．

図 5 は具体物名と Web ページが与えられたときに，Web ページの DOM ツリーにおける具体物名の各出現に対し，属性情報記述ブロックを数え上げるアルゴリズムである．例えば図 3 の Web ページの場合，具体物名

[1] Yahoo!検索 API: `http://developer.yahoo.co.jp/`

[2] クエリとしては，“具体物名 上位語 属性 1 OR 属性 2 OR … OR 属性 10” を用いた．

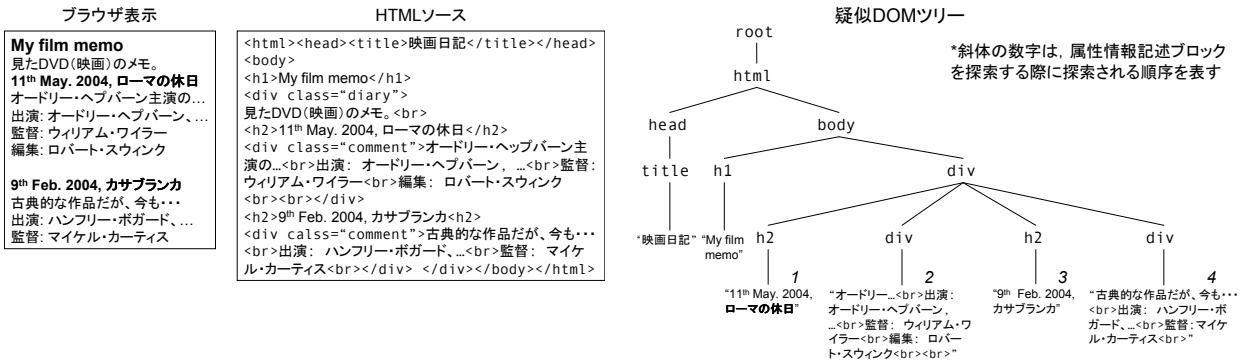

図 3: 属性関係を含む Web ページ（左）とその HTML ソース（中央），及び疑似 DOM ツリー（右）

```
INPUT: Web ページ p
       具体物名 i
       具体物の上位語の属性語集合 A
       関数 dom_tree:  p → dom
OUTPUT:属性情報記述ブロック SB

procedure process_block(p, i, A)
begin
  dom = dom_tree(p)
  IB = {}
  SB = {}
  foreach n in dom
    if n include an intance i then
      IB = IB + {n}
    end if
  end foreach
  sort IB by its position in ascending order
  foreach n in IB
    SB = SB+ find_spec_block(dom, n, A)
  end foreach
  return SB
end

procedure find_spec_block(dom, n, A)
  SB = {}
  foreach n' in dom
            by left-to-right order from n
    if n include an attribute a ∈ A then
      SB = SB + {n}
    end if
  end foreach
  return SB
end
```

図 5: Web ページからの属性情報記述ブロックの抽出アルゴリズム

は最も左の h2 の子ノード中に出現しており，このノードから DOM ツリーの葉ノードを右に辿って上位語の属性語を含んでいる二つの div 要素の子ノード（図 3 右中 2, 4）がこの順で属性情報記述ブロックとして抽出される[3].

3.4 ステップ 3: 属性関係の記述パターンの獲得

このステップでは，ステップ 2 でページから獲得された各属性情報記述ブロックの候補について，その出現のパターンから属性関係の記述パターンを獲得する．

まず，節 3.1 で用いたのと同じ属性語抽出のパターンを用い，各ブロックから属性語を獲得し，上位語の属性語を獲得できたパターンがあれば，獲得に使われたパターンをそのブロックにおける属性語の強調パターンとして採用する．

次に本研究では，属性情報記述ブロック中での属性関係の記述形式に関して以下のような仮説を置く．

仮説: 同一ブロック中では，全ての属性関係は統一的な文字強調パターンによって記述され，かつ，属性値は対応する属性語の直後に出現する．

例えば図 3 の Web ページの例では，属性語が ".+:" というパターンで出現しており，このパターンの直後に属性値が続いている．属性関係の記述パターンは，属性語の強調パターンに，そのパターンを含まない文字の連続として定義できる．

表からの属性関係の抽出　以上で説明した属性・属性値の記述パターンの獲得手法は，その記述形式が上記の仮定を満たす場合（箇条書き）にのみ適応可能である．属性情報が表形式を用いて記述される場合には，具体物の属性値が必ずしも属性語の直後に現れるとは限らないため，特別な処理が必要となる．表形式からの属性関係の獲得については，次節で簡単に説明する．

3.5 ステップ 4: 属性関係の獲得

このステップでは，各属性情報記述ブロックについて，ステップ 3 で獲得された属性関係の記述パターンを用いて，属性関係の抽出を行う．この際，各ブロックの先頭から属性関係の獲得パターンを適用し，同じ属性語が獲得された時点で抽出を終了する．これは，同じ属性語に関する複数の記述が発見された場合には，それは異なる具体物に対する属性情報の記述である可能性が高いためである．例えば，図 3 右からは，属性情報記述ブロック 2 および 4 が抽出されるが，ブロック 4 から獲得される属性語 "出演" はそれ以前に処理されるブロック 2 で先に獲得されるため，この時点で属性関係の抽出を終了し，ブロック 2 から抽出された "出演", "監督", "脚本" に対する属性値のみが出力される．

なお，DOM ツリーで属性情報記述ブロック要素が表形式のセル（td または th で囲まれた範囲）に対応し，かつ属性値の抽出に失敗した場合は，親ノードに対応する表形式から属性関係の抽出を試みる．表形式からの属性抽出では，属性を記述する行または列を，各行・列に含まれる上位語の属性語の数を手がかりに認識する．表が具体物名を含まない場合は，単純にその行または列に

[3] ここでは葉ノードのみを属性情報記述ブロックとして考慮したが，表形式では属性語が具体物名に先行する場合があるため，具体物の先祖ノードに table のセルがある場合には具体物名を含む葉ノードよりも左側のノードも考慮しなければならない場合もある．スペースの関係で詳細は割愛する．

（右または下に）隣接する行または列から属性値を獲得する．表中に具体物名を含むセルが存在する場合には，表形式中の各属性語を含むセルと具体物名を含むセルとの位置関係から属性値を獲得する．それ以上の詳細についてはスペースの関係で割愛する．

なお，本システムを用いて 10 件程度の Web ページ程度から具体物の属性関係情報を獲得する場合，経験的に Web ページをダウンロードする時間を含めても 10 秒程度で出力を返せることも確認しており，リアルタイム Q&A システムとして運用することも可能である．

4 評価実験

提案手法の有効性を検証するために，実際に与えられた具体物とその上位語に対して，属性関係の獲得を試みた．テストに用いた具体物-上位語のペアは，隅田ら [4] の手法により Web 文書から得られた具体物-上位語関係から，恣意的に選んだ上位語 “宿”, “株式会社”, “本”, “歌”, “酒” について，Web 中の箇条書きを情報源として獲得した具体物名からランダムに選んだ 10 個を用いた．

このようにして得られたテストデータに含まれる具体物-上位語関係に対して提案手法を適用し，属性関係を獲得した．具体的には，各具体物-上位語ペアに対し，節 3.2 の手順でクエリを作成し，Yahoo!検索 API で上位 10 件の Web ページから属性関係を獲得した．ダウンロード失敗等の原因により，実際に処理されたページは，1 具体物-上位語ペア辺り平均 8.46 ページ，計 423 ページであった．

まず，得られたページが属性関係獲得の知識源としてどれだけ有用かを調べた．その結果，全体の約 47.5% (201/423) のページでにレイアウト中に入力の具体物の属性関係が含まれていることを確認した．これにより，上位語の属性語を含むクエリと検索エンジンのランキングによって，知識源として有用なページが多く得られることが分かる．

次に，得られたページから提案手法により属性関係の獲得を試みた．表 2 は，具体物の属性関係を実際に含んでいた 201 ページに対する実験結果である．システムが属性関係を獲得したページのうち，獲得した全ての属性語が適切で，かつその全ての属性値に適切な値が含まれたページ（図 2 中 AllCorr + PartCorr）は，59.5%であった[4]．属性語が誤って獲得されたページ（表 2 中で AttrIncorr）のほとんどは，表の認識の失敗に起因するものであり，具体物の属性関係の記述範囲は正しく認識できていることから，この点を改善することで，90%を超える精度を達成できると期待される．

最後に，いずれかのページで適切な属性関係が得られた具体物の数を調べた．50 の具体物に関して，属性関係をレイアウトに含むページが一つでも得られた具体物は 49 (98%) であり，そのうち，提案手法によって，正しい属性関係が得られたページ（表 2 中で AllCorr または PartCorr）が少なくとも一つあった具体物は 37(74%) であった．

5 まとめと今後の課題

本稿では，対象物とその上位語を入力として，Web から具体物の属性関係を獲得する方法を提案した．まず，上位語の属性語を手がかりとして，属性関係獲得の知識源のページを Web から収集し，各ページで具体物の属性関係が書かれている範囲を同定する．その後，属性語を手がかりに属性関係の記述パターンを同定し，属性関係を獲得する．

実験では，5 個の上位語について各 10 個の対象物の属性関係の獲得を試みた結果，提案手法で属性関係が獲得できた全ページのうち，獲得した全ての属性語が適切で，かつその全ての属性値に適切な値が含まれたページの割合は約 59.5%であった．また，約 93.9%については，入力の具体物の属性を記述した範囲を同定できていることが分かった．

今後の課題としては，アプリケーションへの応用 [9] を通してまず広範な上位語に関する評価を行うこと，また表形式での属性関係の認識の精度向上，及び、より大量の属性語を用いてより多くのレイアウトから属性関係の獲得を可能にすることが挙げられる．また，同一の具体物に関して複数のページから得られた属性関係の情報を比較・検討することで，属性関係の言い換えを獲得したり，より信頼性・完備性が高い属性関係を獲得したい．

表 2: 属性関係の獲得実験結果: 属性関係が一切獲得できない (NA), 異なる具体物の属性関係を獲得 (InsIncorr), 不適切な属性語が獲得 (AttrIncorr), 適切な属性関係のみ獲得 (AllCorr), 一部の属性関係で属性値が冗長 (PartCorr)

上位語	total	AllCorr	PartCorr	AttrIncorr	InsIncorr	NA
宿	41	13	4	10	0	14
株式会社	25	3	3	3	6	10
本	52	16	3	16	1	16
歌	34	6	2	8	0	18
酒	49	16	9	4	1	19
平均ページ数	40.2	10.8	4.2	8.2	1.6	15.4
		26.9%	10.4%	20.4%	3.9%	38.3%

参考文献

[1] H.-H. Chen, S.-C. Tsai, and J.-H. Tsai. Mining tables from large scale html texts. In *Proc. COLING*, 2000.

[2] S. Flesca, G. Manco, E. Masciari, E. Rende, and A. Tagarelli. Web wrapper induction: a brief survey. *AI Communications*, 17(2):57–61, 2004.

[3] N. Kushmerick, D. S. Weld, and R. B. Doorenbos. Wrapper induction for information extraction. In *Proc. of the 15th IJCAI*, pages 729–737, 1997.

[4] A. Sumida, K. Shinzato, and K. Torisawa. Concept-instance relation extraction from simple noun sequences using a search engine on a web repository. In *Proc. of the Web Content Mining with Human Language Technologies workshop on the fifth ISWC*, 2006.

[5] M. Yoshida, K. Torisawa, and J. Tsujii. Extracting attributes and their values from web pages. In *Web Document Analysis: Challenges and Opportunities*. World Scientific, 2003.

[6] M. Yoshida, K. Torisawa, and J. Tsujii. Integrating tables on the World Wide Web. *Transactions of the Japanese Society for Artificial Intelligence*, 19(6), 2004.

[7] 吉永 直樹, 鳥澤 健太郎. Web からの属性情報記述ページの発見. **人工知能学会論文誌**, 21(6):493–501, 2006.

[8] 高橋 哲朗, 乾 健太郎, 松本 裕治. 言語パタンと統計的共起尺度による属性関係抽出. In **言語処理学会第** *11* **回年次大会論文集**, pages 432–435, 2005.

[9] 中村 和正, 吉永 直樹, 鳥澤 健太郎. Web から動的に獲得した参考情報を利用する文章作成支援システム. In **言語処理学会第** *13* **回年次大会論文集**, 2007.

[4] 具体物の属性関係を含まないページから誤って異なる具体物の属性関係を獲得したページが 2 ページあり，これを考慮に入れた．