メタデータ付与のための住所録自動生成

村山 紀文　　南野 朋之　　奥村 学
東京工業大学大学院総合理工学研究科　　東京工業大学精密工学研究所
{murayama,nanno}@lr.pi.titech.ac.jp　　oku@pi.titech.ac.jp

1　はじめに

現在 Web 上には多くの情報が存在している．それらの情報をユーザが利用し易くするための方法の 1 つとして, 何らかの軸に基づく情報の組織化がある．たとえば, トピックごとに情報を分類する, 情報を時間軸上に整列する, 含まれている位置情報を元に情報を整理するなどである．

位置情報はこのように, Web 上の情報の組織化において有用な情報と考えられる．位置情報を利用することで，ユーザは自分の求める地域に限定した情報を検索できたり，見ているページ中に書かれている情報と場所的に近い情報が書かれたページへ移動できたりするようになる．また，特に近年では電子地図・GPS 技術・携帯端末・カーナビゲーションシステムなどの発達に伴って，位置情報の有用性は大きく高まってきている．

そのため, Web 上の文書に対して, メタデータとして位置情報を付与する研究も活発化してきている [1]. Web 上の文書中に位置情報が明記されている場合, 情報抽出技術の 1 つである固有名 (住所情報) 抽出技術を用いることで, 文書中の位置情報を抽出すれば, 位置情報をメタデータとして付与することができる．

では, 陽に文書中に位置情報が明記されていない場合, 位置情報をメタデータとして付与することはまったくできないだろうか．たとえば, お店の名前, 病院の名前等の固有名は, それ自体は, 位置情報を明記していないが, 位置情報と関連した固有名と考えることができる．これらの固有名に対し, 位置情報との対応を表すデータベースを作成し, 固有名の出現を元に, 文書に対し位置情報をメタデータとして付与することができれば，より多くの文書に対し位置情報のメタデータを付与することができ, 有用性は大きく増すと考えられる．

そこで, 本研究では, このようなメタデータ付与のために必要な，固有名と対応する位置情報のデータベースを, Web 上の文書から自動的に作成する手法を示す．本研究では，並列して書かれやすい固有名・電話番号・住所の三つ組を Web ページの構造性を利用することで自動的に抽出し，目的となるデータベースを作成する．このデータベースは Semantic Web[2] の考えに基づくなら, 文書中に含まれる固有名に対するメタデータベースということになる．

2　関連研究

本研究で目標とするような住所録型のデータベースは, Yahoo!電話帳[1]やインターネットタウンページ[2]など, 既にいくつか作成されており，Web 上で公開されているものも多い．また，ぐるなび[3]やグルメぴあ[4]のようなポータルサイトで公開されているデータも, 飲食店などに対象が絞られてはいるが，我々の目標とするデータベースに近いものと言える．しかし，これらの電話帳やポータルサイトのデータは人手で構築されたものであり，作成に多大なコストを要する．

本研究では Web から自動的に情報抽出することで, 低コストでのデータベース生成を目指す．同時に，網羅性の高さや新規の情報を素早く獲得出来る点などで, 既存のデータベースよりも有用なデータベースになることが期待される．

本研究の手法に類似した手法としては，Wrapper の自動生成手法 [3][4][5] がある．Wrapper とは，ある特定のフォーマットの Web ページから半構造的なデータを抽出するためのプログラムのことであり，これらの研究ではフォーマットごとに対応した Wrapper を自動生成することで，大量の Web ページから情報収集しようとしている．

この Wrapper の自動生成による情報抽出は， Web ページの構造を自動的に認識・利用して，半構造的な情報を抽出しているという点において，本研究の方針に非常に近い．しかし，Wrapper 生成は複数ページ間に共通する構造に注目しているのに対して，本研究の手法は 1 つの Web ページ内の構造に注目する．これによって，同じ構造を持つページが複数ないようなページからも情報抽出を行うことが出来る．

3　抽出手法

本研究は Web ページから固有名・住所・電話番号の三つ組を自動抽出することを目的としている．このような三つ組が書かれているページは大きく次の二つに分類することが出来る．一つの三つ組だけが書かれているページと, 複数の三つ組が併記されているページである．前者には店舗の HP やポータルサイトの中の一店舗の紹介ページが該当し, 後者にはあるカテゴリに属す

[1] http://phonebook.yahoo.co.jp/
[2] http://itp.ne.jp/servlet/jp.ne.itp.sear.SCMSVTop
[3] http://www.gnavi.co.jp
[4] http://g.pia.co.jp/

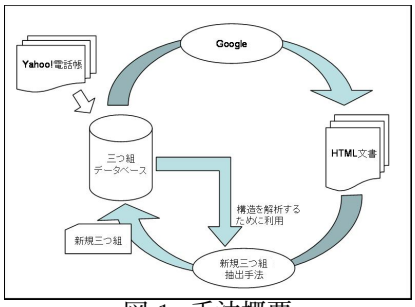

図 1: 手法概要

る店舗の情報を集めたページや, 企業の支店一覧などが該当する.

本研究では特に後者のような複数の三つ組が併記されているページのうち, 三つ組が構造的に規則正しく書かれているようなページを “一覧” 型と呼び，これらのみから三つ組を抽出することにする.

本研究では，あらかじめ三つ組が登録されたデータベースを用意しておき, 上で述べた “一覧” 型ページにおいて, 既にデータベースに登録されている三つ組がどのように記述されているかを解析し，その構造を利用することで新たな三つ組を獲得し, 三つ組のデータベースを拡張していくことを考える.

図 1 に本研究の手法のフローチャートを示す．まず, 既存のデータベース中の三つ組を元に検索を行い, 抽出対象となる HTML 文書を収集する．そこで得られた文書のうち “一覧” 型の文書を対象に, 既存の三つ組を利用して構造解析を行う．その解析結果を用いて, 新規の三つ組を抽出する．本手法では，抽出された三つ組を既知の三つ組に加え，手法を再帰的に適用することで, 新たな抽出対象ページを獲得していくことが出来，また既に獲得済みのページからも新たな構造を解析することが出来るようになる．この繰り返しによって，三つ組のデータベースを拡大していく.

以下で，手法の各処理について，より詳しく説明をおこなっていく.

3.1 抽出対象ページ検索

既知の三つ組から「“電話番号” 市区町村名」というクエリを生成し，Google Web APIs[6] を利用し，検索を行った．これにより，既知の三つ組を少なくとも 1 つ含んでいるページを獲得する.

3.2 DOM Tree の構築

HTML 文書の構造は DOM(Document Object Model) 構造 [7] で表現することが出来る．DOM 構造は DOM Tree という木構造で表現することができる．図 2 に HTML 文書とそれに対応する DOM Tree の例を示す.

本研究では，獲得した HTML 文書から DOM Tree を構築，DOM Tree における各ノードを root（⟨body⟩

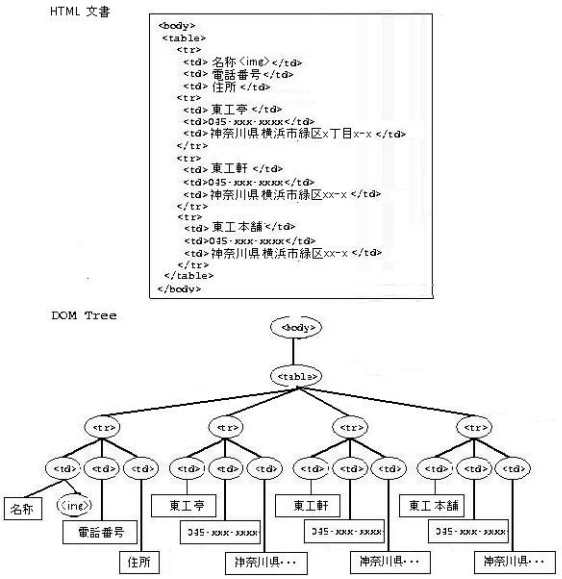

図 2: HTML ファイルと DOM Tree

タグ）からのパスによって表現し，解析・抽出に用いる.

各ノードを一意に示すために本研究では以下で定義されるパス表現を用いる.

/ タグ名 [値 1 ：値 2] /

ここで，値 1 は「同じ階層で何番目の子か」を示す値であり，値 2 は「同じ階層の同じタグの中またはテキストの中で何番目か」を示す値である．例えば図 2 の DOM Tree 表現において「東工亭」が記述された部分を表すパスは

/body[0,0]/table[0,0]/tr[1,1]/td[0,0]/text[0,0]

となる.

3.3 電話番号のマッチング

DOM Tree の中から，フォーマットが一定で簡単にマッチングを行いやすい電話番号の特定を行う.

本研究では，この電話番号を手がかりとして処理を進めていく.

3.4 固有名・住所のマッチング

発見された電話番号のうち，既知の三つ組に含まれるものに関して，対応する固有名と住所が DOM Tree のどの位置に出現しているかを求める.

固有名や住所に関しては表記の揺れが考えられるため，電話番号の前後に存在する文字列と，DB 中の固有名・住所との類似度を求め，その中で類似度が最大かつ 0 以上の文字列に固有名・住所が含まれていると判定する.

以下で，固有名・住所の類似度計算に用いたアルゴリズムについて簡単に説明を行う.

3.4.1 固有名類似度計算

固有名の類似度計算では，文字の出現順序は重要であるが，同時に「株式会社 XYZ」と「XYZ 株式会社」のように共通する文字列が交差するようなケースにも対応出来なければならない．

また，「株式会社 ABCD」という固有名に対して，「株式会社」という文字列はさほど重要ではない．逆に「ABCD」という文字列は重要で，もしマッチング対象に「ABCD」が含まれていた場合には，「株式会社」が含まれていた場合よりも高く評価するべきである．

以上のような点に対応するために，固有名の類似度計算には交差を許す DP マッチング [8] を用い，その計算の重みとして，三つ組データベース中に含まれる全ての既知固有名における文字の 2-gram の出現頻度に基づいた重みを使用した．

3.4.2 住所類似度計算

住所の類似度計算においては，「丁目」「番地」などの数字列の並びが重要であることから，類似度計算の対象となる両文字列から，「数字列」と「数字以外の文字」の配列をそれぞれ作成し，その数字列配列同士または文字配列同士で DP マッチングを行い，双方の類似度の平均を最終的な類似度とする．

3.5 既知三つ組の構造特定

これまでの処理により，DOM Tree 上において，既知の電話番号の記述されたノードとそれに対応する固有名・住所の記述されたノードを特定することが出来る．

これにより，電話番号ノードから固有名ノードへの相対パスと，電話番号ノードから住所ノードへの相対パスをそれぞれ計算することが出来る．

図 2 の例ではそれぞれ，以下のように表現することが出来る．

固有名への相対パス　：../../td[*,-1]/*[0,*]
住所への相対パス　：../td[*,-1]/*[0,*]]

3.6 新規の固有名・住所の特定

特定された相対パスを未知の電話番号に対して適用する．図 2 の例では，未知の東工本舗の電話番号から

```
電話番号　/body[0,0]/table[0,0]/tr[2,2]/td[1,1]/tel[0,0]]
相対パス　../td[*,-1]/*[0,*]
　　↓
固有名　　/body[0,0]/table[0,0]/tr[2,2]/td[*,0]/text[0,0]
```

のようにして，正しく対応する固有名「東工本舗」へのパスを得ることが出来る．住所に関しても，同様に正しくパスを得ることが出来る．

固有名・住所それぞれの相対パスによって示されたノードが存在し，かつそのノードがテキストノードであった場合に，そのノードに存在する文字列を新規固有名または新規住所として特定し，電話番号と合わせて新規三つ組候補として抽出する．

表 1: 実験結果

	一周目	二周目
データベース内三つ組数	46,326	91,846
全対象 Web ページ数	29,148	105,342
（うち，一周目残り）		27,400
電話番号が十分	4,284	15,985
相対パス対	3,887	14,102
新規三つ組獲得	1,748	5,890
（うち，一周目残り）		855
抽出された三つ組数	45,520	153,596

3.7 後処理

新規三つ組候補の住所に対して，地名データベースを参照することで住所としての正しさを判定し，かつ余分な部分を削除する．住所として存在しうると判定されたものを最終的に新規三つ組として抽出し，三つ組データベースに追加していく．

4 実験結果

初期の三つ組のデータベースとして，Yahoo!電話帳の中から神奈川県内の「グルメ・ドリンク」カテゴリの店舗のデータ 46,326 件を使用して抽出実験を行った．

実験の結果を表 1 に示す．ここで，一周目・二周目とは検索・三つ組抽出・抽出された三つ組を DB に追加という一連の作業を一周としたときの，周を表している．（うち，一周目残り）とは一周目で三つ組が一つも発見されず，二周目で再度手法を適用したページ数と，それによって新たに三つ組を獲得することが出来たページ数である．

「電話番号が十分」の項目は，本研究の抽出手法を適用するための最低条件である，既知電話番号を 2 つ以上かつ未知電話番号を 1 つ以上持つ HTML 文書の数である．

「相対パス対」の項目は，固有名・住所のマッチングによって得られた類似度が共に 0 以上の既知三つ組が存在している HTML 文書の数であり，「新規三つ組獲得」は，得られた相対パス対を未知電話番号に適用した結果，新しい三つ組を抽出できた HTML 文書数を示している．

各周で抽出された三つ組のうち，重複した組[5]を取り除いた数を「抽出された三つ組数」で示している．

4.1 考察

本手法を適用することで，一周目・二周目ともに新しい三つ組を大量に抽出することが出来た．全三つ組数は，一周目終了時点で初期三つ組の約 2 倍，二周目終了時点では約 5 倍もの数になった．

[5]本研究では固有名・電話番号・住所の組み合わせが違うものは別の三つ組として数えている

表 2: 抽出漏れ原因

原因	ページ数
構造が存在しない	53
構造が２つ以上	22
固有名・住所不一致	13
固有名・住所不足	12

本手法の重要な特徴の１つは，手法を再帰的に適用することで三つ組数，抽出対象ページ共に増加させていくことが出来る点にある．この新しい抽出対象ページの発見に関しても，２周目で新たに 67,848 ものページを収集することが出来ており, 十分に効果的に機能していると言える．

また, 一周目で構造が発見できなかったものに対して二周目で構造が発見できたページは 855 ページあった．この結果は, 本研究の再帰的な手法が効果的に機能した結果であると考えられる．

4.2 精度評価

三つ組を抽出出来たページの中から，ランダムに 100 ページを抜き出し，それらに対して抽出した三つ組の精度評価を人手で行った．

評価対象の 100 ページのうち，抽出された三つ組が全て正しかったページは 91 ページであった．不適切な三つ組が抽出されていたページを分析したところ，記号など明らかに不適切な文字が固有名に含まれていたページが３ページ，抽出されるべき住所の一部が不足していたものが４ページ，また三つ組構造を間違えて認識していたページが２ページ存在した．

4.3 抽出漏れ評価

既知の電話番号が複数発見され, かつ新規電話番号が存在するページのうち, 三つ組が抽出されなかったページからランダムに 100 ページを抜き出し，それらに対して人手で抽出漏れの評価を行った．

評価対象の 100 ページにおける，抽出されなかった原因と，そのページ数を表 2 に示す．

もっとも多かった原因は，三つ組を抽出出来る構造が存在しないと判断されたケースであり，本研究では抽出対象外とするページである．

次に多かった原因は，１ページ内に構造が２つ以上存在し，それぞれの構造を特定するのに十分な既知三つ組を発見できなかったものである．これらのページは，抽出手法を数回適用し，既知三つ組数を増やしていくことで構造を特定できる可能性がある．

また，ページ中に既知電話番号と対応する固有名と住所は存在するが，データベースに登録されている既知固有名・三つ組と一致しないために抽出出来なかったページも多かった．これらのページも，手法を数回適用しデータベースを拡張していくことで，新規三つ組を抽出出来る可能性がある．

残りの原因は，ページ中に既知固有名の一部，住所の一部しか書かれていないケースであった．この場合，支店名のみや，住所のほんの一部しか書かれていないため，固有名または住所の類似度が下がったため抽出に失敗したものである．これらのページからは，類似度のパラメータを調整することで新規三つ組を獲得することは出来るが，このようなページの情報が信頼できるか，また抽出するべきかは難しい問題である．

5 おわりに

本研究では，住所情報を含む固有名へのメタデータ付与に用いることの出来る，固有名・電話番号・住所の三つ組のデータベースを Web 上から抽出することで自動生成する手法を提案した．

既存の住所録と比較した場合，Web 上からデータを抽出する利点として大規模なデータベースを構築出来る，固有名の表記の揺れを獲得出来ることなどが挙げられるが本研究ではこれらの利点を十分に生かし切れる段階にまでは至っていない．しかし，上記の性質を生かしたデータベースを構築するための第一歩として重要な意味を持っている．

今後の課題としては, 抽出された固有名の余剰部分の削除や，抽出された三つ組のカテゴリ分け，固有名の別称・略称・異表記の獲得などが考えられる．

参考文献

[1] 相良毅, 有川正俊, 高橋昭子. Xml を基本としたテキスト空間情報ベース. 情処研報，99/61, pp. 219–224, 1999.

[2] The World Wide Web Consortium. Semantic web. URL: `http://www.w3.org/2001/sw/`, 2001.

[3] Nickolas Kushmerick, Daniel S. Weld, and Robert B. Doorenbos. Wrapper induction for information extraction. In *Intl. Joint Conference on Artificial Intelligence (IJCAI)*, pp. 729–737, 1997.

[4] Valter Crescenzi, Giansalvatore Mecca, and Paolo Merialdo. Roadrunner: Towards automatic data extraction from large web sites. In *Proceedings of 27th International Conference on Very Large Data Bases*, pp. 109–118, 2001.

[5] 山田泰寛, 池田大輔, 廣川佐千男. 半構造化文書に対する木構造と文字列を組合せたラッパーの自動生成法. 情報処理学会研究報告, 2003-NL-157, pp. 115–122, 2003.

[6] Google. Google web apis - home -. URL: `http://www.google.com/apis/`, 2004.

[7] The World Wide Web Consortium. Document object model. URL: `http://www.w3.org/DOM/`, 1997-2004.

[8] 丸川雄三, 岩山真, 奥村学, 新森昭宏. ローカルアラインメントによるテキスト間の柔軟な対応付け. 情報処理学会研究報告, 2002-NL-151, 2002.