2G4-2

複数のWeb Wrapperを利用した実用的な情報抽出

Practical Information Extraction Using Multiple Web Wrapper

植松 幸生[*1] 内山 俊郎[*2] 片岡 良治[*3] 松井 藤五郎[*4] 大和田 勇人[*5]
Yukio Uematsu Toshio Uchiyama Ryoji Kataoka Tohgoro Matsui Hayato Ohwada

[*1*2*3]日本電信電話株式会社，NTT サイバーソリューション研究所
NTT Cyber Solutions Laboratories, NTT Corporation

[*1*4*5]東京理科大学 理工学研究科
Faculty of Science and Technology, Tokyo University of Science

In this paper, we propose a novel approach to derive web wrapper from numerous amount of web pages using small dictionary data. In the past research, it has to prepare examples of each sites. In our approach it doesn't need input of each site using dictionary data to approximate correct answers. We verified that our approach is effective to automatically extract ingredients data from cooking recipe pages.

1. はじめに

Web 上のデータの増大により，Web 検索のニーズは近年高まりつつある．Web 検索では一般的に文字列を入力としてその文字列に基づくユーザのニーズを推測し検索結果を提示するが，Web データの増大から文字列のみでユーザが欲しい情報に到達する事が困難になりつつある．そこで Web ページからある検索目的にとって有用な情報を抽出し，関係データベースのように高度な検索条件を利用して検索する事は有用である．ある分野とは例えば Web 上のショッピングサイトや，レシピサイト等を指し，高度な検索とは “値段が 10000 円以下の財布” や “材料にナスを利用しているレシピ” のような検索条件を与える事である．こうした属性の付与を大量のデータに対して手作業で行うことは作業コストの面から困難なため，あらかじめいくつかの事例に対して属性を付与し，半自動的に抽出器を生成する Web Wrapper の研究が盛んに行われている [1][2][4]. Web Wrapper とは HTML(Hyper Text Markup Language) のタグを目印に抽出器を生成し，与えた事例以外のページに適用することで多くのページからの抽出を少ないコストで実現出来るため有効な手法と言える．既存研究では HTML タグの反復に着目して抽出器を生成する方法や，HTML タグの DOM 構造 (Document Object Model) を利用して抽出器を表現する方法等がある．しかしながら，既存手法は下記のような点で大量のデータを対象とする場合手間がかかる．

1. 同一の HTML 構造を持つ Web ページ集合を特定する事が必要
2. 同一の HTML 構造を持つ集合別 (主にサイト) に正解データを作成する事が必要

上記した理由から，実用上は正解を付与するコストに見合うだけのサイトのみしか検索対象とする事が出来ない．よって Web における網羅性を欠いたシステムになり，小規模なサイト集合を検索対象とする事が困難であった．例えば料理レシピを対象とした検索を例にすると，材料名+レシピで検索した結果から発見された料理レシピサイトが持つページ数を降順にソートし対数グラフにすると図 1 のようになる．図の縦軸が

連絡先: 植松幸生，NTT サイバーソリューション研究所，神奈川県横須賀市光の丘 1-1，Tel:046 - 859 - 4923，Fax:046 - 855 - 1730，uematsu.yukio@lab.ntt.co.jp

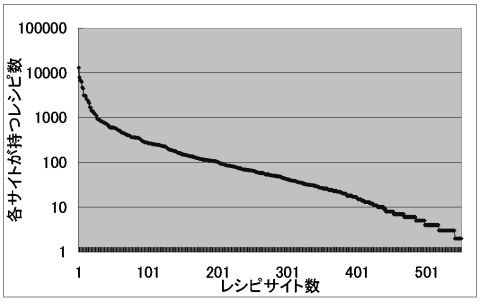


図 1: あるサイトが持つレシピページ数

各サイトが持つレシピ数，横軸がサイト数である．図を見ても分かる通り 1000 ページを有するサイトは上位 50 件にも満たないため，サイトに依存する正解データの付与で大量のレシピサイトを検索対象とする事は非常に手間がかかる．そこで本研究ではサイト毎に完璧な正解データを用意するのではなく，抽出したい文字列の一部が記述された辞書データを与え，その辞書データに基づいて正解データを近似して学習し，サイトに依存しないより実用的な情報抽出を行う方法を提案する．

2. 提案手法

提案手法が解こうとする問題について定義する．情報抽出は大きく分けて 2 つの目的を持つ．1 つは抽出した情報自体を利用する目的，もう 1 つは抽出した情報をそのページもしくはそのページの一部) の属性情報として利用する目的である．前者の抽出した情報自体を利用する場合は辞書拡張等が目的となる [3]．本研究では後者の抽出した情報をページの属性情報として関連づける情報抽出に関して議論する．この場合，前者との違いはあるページに存在するすべての属性情報を抽出する必要がある．例えば料理のレシピ情報が記述されたページから，材料情報を抽出するタスクや，映画の紹介ページからすべての出演者名を抽出するタスクである．本稿では料理レシピから材料情報を抽出するタスクを例に説明する．

提案手法では入力として，ある分野に属する文字列の一部

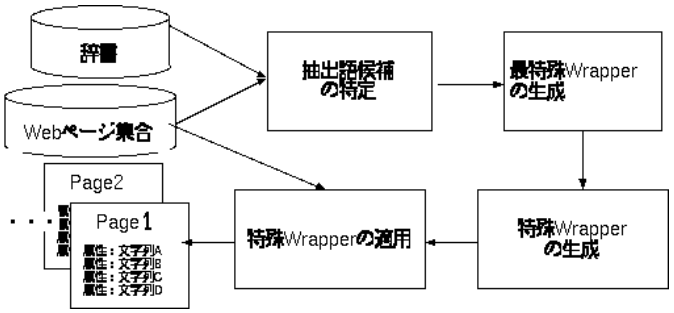


図 2: システム構成図

ヒキ肉
辛味噌
味噌
…

図 3: 辞書データ

が記述された辞書を入力とする．既存研究との違いは既存研究では抽出すべき語がそのページにおける出現位置が与えられるのに対し，本研究では抽出すべき文字列が辞書で与えられるために出現位置が特定不可能な点である．辞書データのみで抽出器を生成する事が実現すると入力がサイトに依存しないため，より多くのデータを対象とした抽出が可能となる．

2.1 システム構成

図 2 にシステム構成を示す．まず，入力として辞書データと Web ページ集合を入力とする．それら入力されたデータから抽出語候補の特定を行う．抽出語候補の特定とは，入力された辞書に記述された文字列が，Web ページ集合のどの部分に出現しているかを特定する事を指す．次にその抽出語候補となった位置を対象として，HTML の構造を利用した最特殊 Wrapper の生成を行う．最特殊 wrapper とは抽出語候補のみを抽出するための最も特殊な Wrapper の事である．その最特殊 Wrapper の集合を入力として，正解データの近似を行い，最終的な Wrapper を導出する．導出された Wrapper を元の Web ページ集合に対して適用し，辞書に含まれない文字列を含めたあるページにおける全ての属性情報の抽出を試みる．ラッパは各ノードまでの PATH と LR のパタンをルールとする [5] を用いる．次章より各機能の詳細を説明する．

2.2 辞書を入力した抽出語候補の特定

抽出語候補の特定方法を説明する．図 3 に入力される辞書例を示す．この辞書の目的はあるページにおいて，レシピに利用される材料情報を抽出することである．辞書には材料情報に取り得る文字列の一部が記述されている．この辞書に記述されている文字列がある文書内に存在した際にその文字列を抽出語候補とする．抽出語候補は記述長の長い文字列から順番に適用する．例えば図 3 の “味噌” という文字列は辛味噌の部分文字列になっているので，“辛味噌”，“味噌” の順番でマッチングを行う．

このような辞書データを利用して図 4 に示す HTML 文書に適用し抽出語候補を特定する．図 5 が図 3 を入力として図 4 に対して抽出語候補を特定した例である．図中の <extract_cand> が抽出語候補である．既存研究の場合は抽出可能なエリアの文字列のみが抽出候補として特定する事が出来るが，本研究の場合は辞書で与えられるため，抽出すべきエリアであるかの判定は不可能である．よって，自然文中の材料情報も抽出語候補になる．

```
<HTML>
<head>
<title>マーボナスの作り方</title>
</head>
<Body>
マーボナスは下記の材料で構成される.
<table>
  <table>
    <tr><td>ナス</td><td>1 本</td></tr>
    <tr><td>ヒキ肉</td><td>100g</td></tr>
    <tr><td>(A)</td></tr>
    <tr><td align=right> ┌</td><td>辛味噌</td><td>10g</td></tr>
    <tr><td align=right> │</td><td>山椒</td><td>適量</td></tr>
    <tr><td align=right> └</td><td>塩・胡椒</td></tr>
  </table>
まず，ヒキ肉を炒め塩・胡椒で味を整える…
</Body></HTML>
```

図 4: 抽出 HTML 例

```
<HTML>
<head>
<title>マーボナスの作り方</title>
</head>
<Body>
マーボナスは下記の材料で構成される..
<table>
  <table>
    <tr><td>ナス</td><td>1 本</td></tr>
    <tr><td><extract_cand></td><td>100g</td></tr>
    <tr><td>(A)</td></tr>
    <tr><td align=right> ┌</td><td><extract_cand></td><td>10g</td></tr>
    <tr><td align=right> │</td><td>山椒</td><td>適量</td></tr>
    <tr><td align=right> └</td><td>塩・胡椒</td></tr>
  </table>
まず，<extract_cand>を炒め塩・胡椒で味を整える…
</Body></HTML>
```

図 5: 抽出語特定例

2.3 Wrapper 導出アルゴリズム

提案手法では前述した最特殊 Wrapper からボトムアップに学習し，目的となるラッパを導出する．この目的となる Wrapper を特殊 Wrapper と呼ぶ．特殊 Wrapper(以下 w_{msh}) とはあるページ集合 P が与えられた時にその P に含まれる属性情報

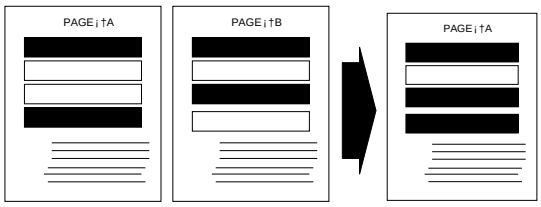


図 6: 正解データの近似例

L のサブセットのみを包含する Wrapper の中で最も一般的な Wrapper である．また，あるページに対して 1 つの Wrapper で抽出する事が不可能な場合は，複数の特殊 Wrapper を生成し抽出する．

2.4 正解データの近似

前述した抽出語候補から生成した最特殊 Wrapper を入力として Wrapper を生成する方法について述べる．前述した抽出語候補の特定は出現位置を特定する入力と比較して 2 つの点で異なる．一つは必ずしも HTML の構造で抽出可能な候補が特定されているわけではない点と，抽出すべき語にも関わらず，抽出語となってない語が存在する点である．正解データの近似では入力されたあいまいなデータから信頼できるデータとそうでないデータを特定する事で完全な正解データが与えられる場合を近似する．近似は下記のような手順で行う．

1. ページ集合 P で生成された最特殊 Wrapper の中で唯一の Wrapper を削除する
2. あるページ P_n で学習する過程で包含されてしまう L_n でない文字列の中で，ページ集合 P 中の最特殊 Wrapper で包含する文字列は L'_n として学習する

生成された最特殊 Wrapper が唯一のものは，自然文中のデータやタイトル等 HTML の構造では抽出が困難な位置に出現している文字列である事が多いのでその抽出語候補は削除する．入力される辞書データはすべての抽出語が特定されているわけではない．図 6 にその例を示す．図中の帯で表記されたものが抽出すべき文字列で，黒い帯が抽出語候補となったものである．そこで同一 HTML 構造を持つその他のページと照らし合わせ，P_n において L_n に出現すべき文字列を P 中の最特殊 Wrapper で包含可能な文字列を含め L'_n とする．図 6 中の例を取ると $PAGE_A$ では 2 つ抽出語候補が特定されているが，同一の HTML 構造をもつ $PAGE_B$ において，包含される抽出語候補を照らしあわせる事で， 右の $PAGE_A$ のように $PAGE_B$ の特殊 Wrapper で包含される L を抽出語候補と同様に扱う．

2.5 特殊 Wrapper の生成方法

図 7 に特殊 Wrapper を生成するアルゴリズムを示す．入力として P, L_{cand} が入力される．P は Web ページ集合で，L_{cand} はそのページ集合に対する抽出語候補の集合である．P に含まれるページ集合 P を入力として，$create\_msh$ である単一の抽出語候補のみを出力する Wrapper(最特殊 Wrapper) を全ての L_{cand} に対して求める．その最特殊 Wrapper の集合を W_{msh_cand} とする．次に $Check\_msh$ で導出された W_{msh_cand} の中で唯一の Wrapper を削除する．$w_i(P_n)$ が L'_n を包含する場合は，その時の Wrapper を $wrapper\_cand$ に保存して $generalize\_wrapper$ にて一般化する．$w_i(P_n) \not\subseteq L'_n$ の場合はその前に保存されていた Wrapper 候補 ($wrapper\_cand$) を特殊 Wrapper として W_{msh} に保存する．この W_{msh} によって

Let P be a set of Web Pages;
Let L_{cand} be a set of label candidates;
Let W_{msh} be a empty set;
Let W_{msh_cand} be a empty set;
$W_{msh\_cand} = create\_msh(P, L)$
$W_{msh\_cand} = check\_msh(W_{msh_cand})$
foreach w_i in W_{msh_cand}
  while($w_i(P_n) \subseteq L'_n$)
    $wrapper\_cand = w_i$
    $w_i = generalize\_wrapper(w_i, P)$
  end while
  $W_{msh} = W_{msh} \cup wrapper\_cand$
  $W_{msh\_cand} = W_{msh\_cand} \cap \bar{W}\_msh$
end for

図 7: 特殊 Wrapper の生成アルゴリズム

包含された事例をすべて削除し，残りの W_{msh_cand} に対して同様のプロセスを行う．これをすべての L に対する最特殊ラッパに対して行った結果 W_{msh} が最終的に求める特殊 Wrapper となる．一般化した結果，特殊 Wrapper が複数の Wrapper で構成される事もある．

3. 検証実験

レシピサイト 2 サイトを対象として提案するアルゴリズムの動作検証実験を行った．利用したサイトはレシピ大手の e-recipe [*1]と Nestle[*2]のデータを用いてどの程度の精度が出るのかを検証実験を行った．入力となる辞書としては，レシピサイトで利用されている材料名を頻度順に並べ，その上位から材料名を辞書として使用し，材料数を 1〜40 まで変化させて実験を行った．実験データとしては，比較的抽出が容易である e-recipe と，HTML が複雑な構造を持つ nestle を対象に実験を行った．

ここで，抽出精度を評価する抽出精度評価の評価軸については，精度 (Precision)，再現率 (Recall) それに精度と再現率の調和平均である F_1 値を用いて評価を行った．今回は精度，再現率を算出する際に，True Negative はない．これは我々の抽出器は抽出すべきでない文字列を判断する方法を持たないからである．よって精度，再現率は下記の式で算出した．

$$Precision = \frac{TP}{TP + FN} \tag{1}$$

$$Recall = \frac{TP}{TP + FP} \tag{2}$$

$$F_1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{3}$$

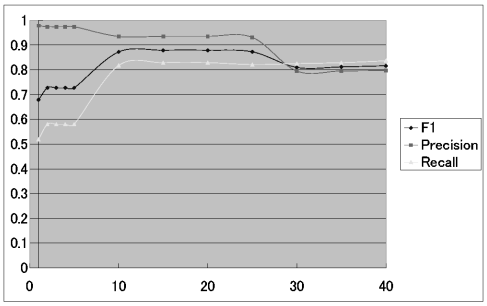
$$TP : TruePositive \tag{4}$$

$$FN : FalseNegative \tag{5}$$

$$FP : FalsePositive \tag{6}$$

ここで，$TruePositive$ は本システムが正解と判定し正解だった数，$FalseNegative$ は本システムが正解と判定したが，不正解だった数，そして $FalsePositive$ は正解だが本システムが抽出できなかった数である．

*1 http://www.e-recipe.co.jp/regulars/
*2 http://www.recipe.nestle.co.jp/recipe/



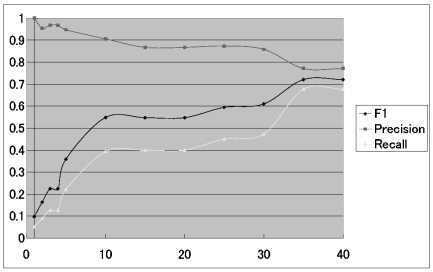


図 8: 入力される辞書 (単語数) の変化と精度の推移 (上図 e-recipe, 下図 nestle)

図 8 に実験結果を示す．横軸に入力した辞書が持つ文字列数と，縦軸には *Recall*, *Precision* それに F_1 値を示した．実際に利用した材料名辞書の頻度上位 10 件を図 9 に示す．

いずれも入力が少ない時は精度は高いものの再現率が著しく低いが，辞書としてある程度の単語数を入れると再現率が向上する．HTML の構造が単純な e-recipe では辞書として 10 単語程度与えるだけで正解を近似してラッパを生成可能である．一方複雑な構造を持つ nestle でも e-recipe 同様に，辞書を増やすほど再現率が向上するが，構造が複雑なためその精度の推移は緩やかであるもの，ある程度辞書を用意する事で同じように抽出出来ることがわかった．

頻度順位	材料名
1	卵
2	砂糖
3	バター
4	ネギ
5	玉ねぎ
6	トマト
7	レモン
8	肉
9	小麦粉
10	にんにく

図 9: 使用した材料名辞書

4. まとめ

本稿では辞書を利用して事例付与を近似することで，サイトに依存せず Wrapper を導出する一手法を提案した．また，その方法を料理レシピの記述されたページから材料情報を抽出するタスクに対して適用し，本手法の動作検証を行った．

今後の展望としては，その他のドメインに適用し，どのようなドメインが有効かを検証する．また，今回は辞書データで特定出来るようなドメインを対象としたが，映画の紹介ページから出演者情報等を抽出する事を目的とする場合は辞書データではなく，固有表現抽出等で人名の出現位置を特定する等の他の技術を組み合わせた抽出語候補の特定を検討していきたい．

参考文献

[1] S. Gupta, G. Kaiser, D. Neistadt, and P. Grimm. Dom-based content extraction of html documents. In *World Wide Web conference*, pp. 207–214, 2003.

[2] Nicholas Kushmerick. Wrapper induction: Efficiency and expressiveness. *Artificial Intelligence*, Vol. 118, No. 1-2, pp. 15–68, 2000.

[3] Marius Pasca and Benjamin Van Durme. What you seek is what you get: Extraction of class attributes from query logs. In *IJCAI*, pp. 2832–2837, 2007.

[4] 山田泰寛, 池田大輔, 坂本比呂志, 有村博紀. Www からの情報抽出–web ラッパーの自動構築. 人工知能学会誌, Vol. 19, No. 3, pp. 302–310, 2004.

[5] 植松幸生, 内山俊郎, 片岡良治, 松井藤五郎, 大和田勇人. 複数の web wrapper による高精度な情報抽出. 情報処理学会データベースシステム研究会, No. 6, pp. 117–123, 2007.