

複数の Web Wrapper による高精度な情報抽出

植松幸生^{†,††} 内山俊郎^{††} 片岡良治^{††}
松井 藤五郎[†] 大和田 勇人[†]

本稿は大量にある Web ページから有用な情報を抽出する Web Wrapper(以下ラッパ)を自動的にかつ高精度に生成する方法を提案する。従来の Web ラッパの研究では抽出器が単一のラッパで構成される事を前提として一般化していた。しかしながら、単一のラッパでは、抽出器の表現力が乏しく、あるページ内における小さなレイアウトの変化に適応できないため、ラッパの生成に失敗する事があった。本研究ではラッパを複数のラッパで構成するために特殊ラッパと呼ぶラッパを定義し、ラッパを特殊ラッパの集合で表現することで、今まで正確に抽出する事が困難だったページに対しても抽出可能である事を確認した。

Information Extraction Using Specic Rule Wrapper Array

YUKIO UEMATSU,^{†,†} TOSHIO UCHIYAMA,^{††} RYOJI KATAOKA,^{††}
TOHGORO MATSUI[†] and HAYATO OHWADA[†]

In this paper, we explain novel web wrapper class to extract efficient informaion from numerous amount of web pages. In the past research, the wrapper expressed by a rule. However expressiveness of a wrapper is not enough to apply pages that have a small difference of layout. We define specific rule wrapper array class to express those site. We experimnted that it can handle more sites than the past research.

1. はじめに

WWW 上にある大量の Web ページから、関係データベースのように属性と属性値のペアを抽出する技術として、Web Wrapper(以下ラッパ)⁴⁾がある。属性とは関係データベースのカラム名に値するもので、属性値とはその属性が付与された文字列をさす。例えば料理レシピの材料や、商品の価格という属性をある文書中の“ナス”や“100 円”という文字列に対して付与する。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、その事例から 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半自動的に抽出器を生成するラッパの研究が盛んに行われている¹⁾³⁾⁴⁾⁵⁾⁶⁾⁷⁾。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

input: $D = \{\dots < P_n, L_n > \dots\}$

output: W such that $W(P_n) = L_n$,
 $\forall < P_n, L_n > \in E$

図 1 我々が解こうとする問題

方法や、HTML タグを DOM 木 (Document Object Model) とみなす事で抽出器を表現する方法等、単純な構造を持った Web ページのみを抽出対象としている。しかしながら、抽出したい文字列が単純な HTML 反復や、DOM 木で指定可能な位置に出現していない場合、既存のラッパクラスでは抽出器を表現する事が出来ず、抽出不可能なページが多く存在するという問題点があった。そこで、本研究ではラッパを複数のラッパ集合で構成する新しいラッパクラス (特殊ラッパ集合クラス) を提案し、今まで単一のラッパで表現することが困難だった Web ページに対しても、抽出器を導出する方法を述べる。

図 1 に我々が解こうとする問題を定義する。本研究も既存研究⁴⁾と同様、既に属性値が付与されたいくつかのページが入力 (事例) となる。 $P_n \in P$ と $L_n \in L$ の組を $< P_n, L_n >$ の集合 E とする。 P_n は

[†] 東京理科大学理工学研究科

Faculty of Science and Technology, Tokyo University of Science

^{††} 日本電信電話株式会社, NTT サイバースリユーション研究所
NTT Cyber Solutions Laboratories, NTT Corporation

```

<HTML><Body>
<title>マーボナスの作り方</title>
マーボナスは下記の材料で構成される..
<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><td></tr>
<tr><td>「</td><td>辛味噌</td><td>10g</td></tr>
<tr><td>|</td><td>山椒</td><td>適量</td></tr>
<tr><td>「</td><td>塩・胡椒</td><td>少々</td></tr>
まず、ヒキ肉を炒め塩・胡椒で味を整える...
</Body></HTML>

```

図 2 料理レシピ例 (P_{ex})

```

<HTML><Body>
<title>マーボ豆腐の作り方</title>
マーボ豆腐は下記の材料で構成される..
<table>
<tr><td>豆腐</td><td>1 本</td></tr>
<tr><td>ヒキ肉</td><td>100g</td></tr>
<tr><td>長ネギ</td><td>1 本</td></tr>
<tr><td>(A)</td><td></tr>
<tr><td>「</td><td>辛味噌</td><td>10g</td></tr>
<tr><td>|</td><td>山椒</td><td>適量</td></tr>
<tr><td>「</td><td>塩・胡椒</td><td>少々</td></tr>
まず、ヒキ肉を炒め塩・胡椒で味を整える...
</Body></HTML>

```

図 3 料理レシピ例 (P_{ex2})

単一のページで、 L_n が P_n 内の属性値の集合である。 $\langle P_n, L_n \rangle$ の事例集合 D から $\forall \langle P_n, L_n \rangle \in E$ に対して、 $W(P_n) = L_n$ となるような関数 W を求める問題である。よって、 D の大きさが、 E よりも小さい場合、 $\langle P_n, L_n \rangle$ をいくつか与えるだけで、すべてのページから抽出可能なため、半自動的にラッパを生成する事が出来る。

ここで我々の方法が効果的な情報抽出の典型的な例として料理レシピの記述されたページから材料属性を付与する文字列を抽出するタスクを上げ、既存研究の問題点と提案手法の効果について述べる。材料属性とは料理レシピで利用される材料の事である。図 2、図 3 に料理レシピが記述された Web ページ例 (P_{ex}, P_{ex2}) を示す。図 2 から、 $W(P_{ex}) = L_{ex} \{ \text{ナス, ヒキ肉, 辛味噌, 山椒, 塩・胡椒} \}$ となる関数 W を求め、その求められた関数を他のページ (例えば図 3) に適用し、 $W(P_{ex2}) = L_{ex2} \{ \text{豆腐, ヒキ肉, 長ネギ, 辛味噌, 山椒, 塩・胡椒} \}$ となる W を自動的に抽出する事が目的である。ここで事例 D として P_{ex} が与えられたとして、 P_{ex}, P_{ex2} を包含するようなラッパ W を求める問題について考える。

図 2、図 3 のページは、HTML のテーブル構造が複雑であり、DOM 木において同一階層であっても、抽

$L: \langle td \rangle \ R: \langle /td \rangle \langle td \rangle$

図 4 LR ラッパ例

出すべき文字列と抽出すべきでない文字列が存在するため、抽出器を単一のラッパでは表現出来ない。例えば図 2 は以下の二つの理由から抽出器を表現する事が困難である。

- (1) “ナス, ヒキ肉” と “辛味噌, 山椒, 塩・胡椒” が HTML の $\langle td \rangle$ タグによって表現される異なる列に記述されている
- (2) “ナス, ヒキ肉” という抽出すべき文字列と、“(A), 「, |, 「” という抽出すべきでない文字列が同列に記述されている

これは既存のラッパには Web ページ内の抽出対象となるサブセット (区間) において、各属性値が目印として指定可能なタグや文字列で囲まれているという仮定があるからである。抽出対象となる区間とは、LR ラッパでは Web ページ全体、HLRT ラッパ[☆]では head と tail によって指定された属性値が存在する区間、Tree ラッパでは HTML の DOM 構造で指定できる範囲である。しかしながら、このような仮定は抽出可能なページを限定してしまう。その例である図 2 に対して、LR ラッパを利用して抽出器を生成すると図 4 に示すラッパを生成してしまい、不必要な情報も抽出してしまう。また、Tree ラッパで DOM の構造を利用して抽出する場合も同様に不必要な情報を抽出してしまったり、抽出漏れを起こしてしまい、 L_{ex}, L_{ex2} をすべて包含し、かつ誤抽出を起こさないようなラッパ関数を生成する事は出来ない。

そこで、本研究では特殊ラッパの集合で抽出器を表現することで、上記問題を解決する。図 5 に本研究が求めたいラッパ例を示す。このラッパを図 3 に適用すると $W(P_{ex2}) = L_{ex2} \{ \text{豆腐, ヒキ肉, 長ネギ, 辛味噌, 山椒, 塩・胡椒} \}$ となり、 P_{ex2} における属性値 L_{ex2} のみをすべて包含するようなラッパ W を表現することが出来る。このようにラッパ W を複数のラッパで表現する事で、抽出したい文字列のみに属性を付与するような抽出器を構成する事を可能にする。また、タグとタグに囲まれた文字列を $\langle TEXT \rangle$ というタグによって表現する事で、図 6 のように、より一般的な抽出器を生成する事が出来る。

2. 提案手法

我々のアイデアは特殊ラッパと呼ぶ過度に一般化し

☆ 詳細は⁴⁾を参照

```

L: <tr><td> R: </td><td>100g
L: <tr><td> R: </td><td>1 本
L: </td><td> R: </td><td>
L: | </td><td> R: </td><td>
L: </td><td> R: </td><td>

```

図 5 本研究が求めたいラップ例

```

L: <tr><td> R: </td><td><TEXT>
L: <TEXT></td><td> R: </td><td>

```

図 6 テキストを一般化した本研究が求めたいラップ例

$$W_{msh} = \{w_1 \cdots w_m\}$$

$$W_{msh}(P_n) = \sum_{i=1}^m w_i(P_n) \text{ where } w_i(P_n) \in L_n$$

$$W_{msh}(P_n) = L_n$$

図 7 特殊ラップ集合クラスの構成

ないラップを利用し、その特殊ラップの集合で抽出器を表現する事で今まで表現出来なかった Web ページの抽出器を表現することを可能にする。図 7 に特殊ラップ集合クラス W_{msh} の構成を示す。特殊ラップ集合 W_{msh} は複数のラップ関数 w_i で構成され、その W_{msh} を構成するラップ w_i を特殊ラップと呼ぶ。特殊ラップ w_i はあるページ P_n が与えられた時にその P_n に含まれる事例 L_n のサブセットのみを包含するラップの中で最も一般的なラップである。また、 $w_i(P_n)$ では、 L_n に含まれる文字列以外が抽出される事は無いように構成する。

2.1 特殊ラップ集合の生成方法

前述した通り、求めたい特殊ラップとは、 $w_i(P_n) \in L_n$ となる w_i で最も一般的なラップの事である。本研究では最も特殊なラップ (最特殊ラップ) からボトムアップに一般化を行い最終的な特殊ラップ集合を求める。図 8 が特殊ラップを生成するアルゴリズムである。特殊ラップを生成するための入力として P, L が入力される。 P は Web ページ集合である L はそのページ集合に対して付与された属性値の集合である。 P に含まれる任意のページ P_n を入力として、 $create_msh$ である単一の文字列のみを出力するラップ (最特殊ラップ) を L_n に含まれる個々の属性値に対して求める。ある単一の文字列のみを抽出するラップが $w_i(P_n) \in L_n$ を満たす場合は、その時のラップを $wrapper_cand$ に保存して一般化を行う。 L_n 以外の文字列が出力された場合はその前に保存されていたラップ候補 ($wrapper_cand$) を特殊ラップとして W_{msh} に保存する。次に図 2 のページと属性値 “ヒキ肉” に対する最特殊ラップから一般化する仮定を図 9 に示す。ここでは PLR ラップ⁷⁾

```

Let  $P$  be a set of Web Pages;
Let  $L$  be a set of labels;
Let  $W_{msh}$  be a empty set;
Let  $W_{msh\_cand}$  be a empty set;
foreach  $P_n$  in  $P$ 
   $W_{msh\_cand} = create\_msh(P_n, L_n)$ 
  foreach  $w_i$  in  $W_{msh\_cand}$ 
     $extracted\_results = apply\_wrapper(w_i(P))$ 
    while( $w_i(P_n) \in L_n$ )
       $wrapper\_cand = w_i$ 
       $generalize\_wrapper(w_i)$ 
       $extracted\_results = apply\_wrapper(w_i(P))$ 
    end while
    push  $wrapper\_cand$ ,  $W_{msh}$ 
  end for
end for

```

図 8 特殊ラップ集合の生成アルゴリズム

で特殊ラップを作成する。PLR ラップとは抽出語候補の左右の文字列の他に、その抽出語候補が HTML の DOM 木のどの位置に存在する (この DOM 木の位置を PATH と呼ぶ) かを保存するラップの事である。まず、ステップ 1 では単一の文字列を抽出する最特殊ラップを生成する。ここでは PATH と、一つ前の要素である “ナス” という文字列と重複しないように、最特殊ラップを構成する。次にステップ 2 では、LR ラップのルール of 短い方から 1 つルールを削除する事で一般化する。そのラップを P_{ex} に適用した際に $w(P_{ex}) \notin L_{ex}$ になるまで一般化を繰り返す。ステップ $n-1$ では、ヒキ肉とナスが抽出されたが、 $\{\text{ヒキ肉}, \text{ナス}\} \in L_{ex}$ なので、一般化を続ける。ステップ n では $\{\text{ヒキ肉}, \text{ナス}, (A)\}$ となり、 $w(P_{ex}) \notin L_{ex}$ なので、ステップ $n-1$ のラップを “ヒキ肉” から導出可能な特殊ラップとする。こうした特殊ラップを各特殊ラップで包含されないすべての属性値に対して作成し、あるページ P_{ex} から特殊ラップ集合を導出する。

3. 特殊ラップ集合クラスの拡張

特殊ラップ集合クラスの拡張例として、事例付与の自動化を行う。提案した特殊ラップ集合クラスは表現力は高いものの、各サイト毎に事例を与える必要がある点に関しては、今までに提案されているラップと同様である。しかしながら、大量のデータに対してラップを適用する場合はこうしたサイト毎に事例を与える方法は属性値を付与するコストが大きい。そこで、特殊ラップは同一サイト内の異なる構造をもったページ



図 13 同一サイトのページ例

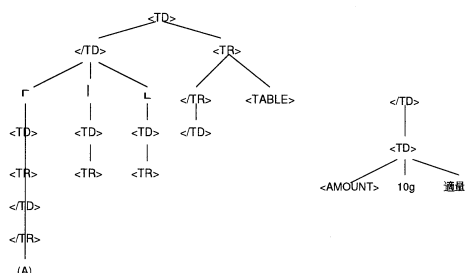


図 14 特殊ラップの構造化例 (左図: L 項, 右図: R 項)

ここでは抽出される語を中心として木を作成する事で、どのラップを選択して抽出すべきかを求める。図 14 に特殊ラップ集合の LR 項に着目し、ラップを木構造に表現した例を示す。図 14 においてルール長が長いものから適用する。

4. 評価実験

前述した提案手法の有効性を示すためにレシピページから材料属性を付与する文字列を抽出するタスクにより下記の 2 つの評価実験を行った。

- (1) 本手法と既存のラップクラス (PLR ラップ) のカバー率の比較。
- (2) 辞書マッチを利用した事例付与方法において、入力される辞書と抽出精度の推移。

まず、実験 1 として本研究の提案するラップクラスが適用可能なサイトが、既存のラップクラス (PLR ラップクラス) とどのような包含関係があるのかをカバー率による評価を行った。カバー率の評価は実験に利用したサイトに対して、各ラップクラスが抽出器を表現する事が可能否かを評価する。また、それらのサイトで表現するのに必要な特殊ラップ数についても調査する。次に実験 2 では特殊ラップ集合クラスの拡張例として辞書マッチを用いた事例付与による制度の推移を検証する実験を行った。入力となる辞書データの量を変化させ、用意する辞書データの量によってどの程度精度、再現率及び F_1 値が変化するかを調査した。

表 1 実験に使用したサイト一覧

サイト ID	サイト URL
site1	www.ajinomoto.co.jp/recipe/condition/menu/
site2	www.e-recipe.co.jp/regulars/
site3	www.recipe.nestle.co.jp/recipe/
site4	www.s-recipe.com/
site5	www.foods.co.jp/cheese/recipe/
site6	www.nisshin.com/life/italian/recipe/
site7	www.kikkoman.co.jp/homecook/
site8	mr.hamacco.net/pasta/
site9	www.boxos.com/cook/
site10	www1.neweb.ne.jp/wa/asiafoods/

4.1 データセット

入力するデータセットは Google ☆に“レシピ”という文字列で検索した結果、料理レシピのページであった URL72 サイトのレシピデータからいくつかのサイトを収集し実験に使用した。使用したサイトを図 1 に示す。これらのサイトから 12 ページずつ収集を行い、2 ページを事例として利用し、残りの 10 ページに対して事例から作成された抽出器を適用し実験を行った。2 ページを事例として利用した理由は、先行研究⁴⁾において、2 ページでほとんどのサイトのラップの導出に成功しているからである。その 12 ページに対して人手で材料属性を付与した。

4.2 実験 1: カバー率評価

前述したデータセットを利用してカバー率評価を行う。正解データの事例を与え、導出されたラップで、各サイトのすべてのページにおいて抽出する事が可能か実験を行った。図 2 がカバー率評価の実験結果である。○と記述されているサイトは抽出器を表現する事が可能サイトで、×は抽出器を表現する事が出来なかったサイトである。左から図 1 のサイト ID、本システムの抽出可否、既存研究の抽出可否、本システムを利用して事例から得られた特殊ラップ数である。実験結果を見ても分かる通り、本研究のラップの方が既存のラップよりもカバー率が高く、表現可能なサイト数が多い。また、特殊ラップ数を見ると、単純に単一のラップのみで表現出来たサイトは 2 つのみで、その他は複数のラップにより表現可能になるサイトであった。複数の特殊ラップによって表現されているにも関わらず、PLR ラップが抽出可能となっているサイトは、PLR ラップがタグの属性 (a タグの href 属性など) を無視して PATH のみを利用しているため、本システムよりも、汎化されている例があるためである。また、本システムで△がついている site3 に関しても、

☆ <http://www.google.co.jp>

表 2 カバー率評価

サイト ID	本システム	PLR ラップ	特殊ラップ数
site1	○	×	2
site2	○	×	6
site3	△	×	-
site4	○	○	2
site5	○	○	1
site6	○	○	2
site7	○	×	5
site8	○	○	7
site9	○	○	1
site10	○	○	2

入力する事例数を増やす事で、抽出可能である事がわかった。

4.3 実験 2:与える辞書と精度の関係

次に特殊ラップ集合クラスの拡張例の実験として、どの程度辞書データを与える事で、どの程度の精度が出るのかを評価した。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選択し、辞書として使用する材料数を 1~25 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、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、再現率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。

$$Precision = \frac{TP}{TP + FN} \quad (1)$$

$$Recall = \frac{TP}{TP + FP} \quad (2)$$

$$F_1 = \frac{2 * Precision * Recall}{Precision + Recall} \quad (3)$$

$$TP : TruePositive \quad (4)$$

$$FN : FalseNegative \quad (5)$$

$$FP : FalsePositive \quad (6)$$

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、間違いだったもの、そして FalsePositive が正解だがシステムが抽出できなかった数である。

図 15 に実験結果を示す。横軸に入力される辞書数と、縦軸には Recall, Precision それに F_1 値を示した。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。これは単純なサイトでは

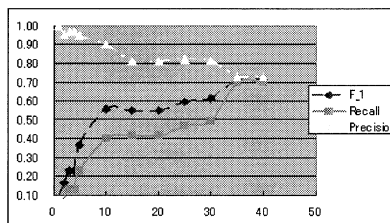
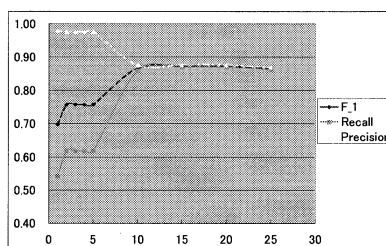


図 15 入力される辞書 (単語数) の変化と精度の推移 (上図 e-recipe, 下図 nestle)

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 16 材料名辞書

あるが、本研究のラップが 10 単語程度でラップを生成する事を示している。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。実際に利用した材料名辞書の頻度上位 10 件は図 16 である。

5. 関連研究

情報抽出し属性値を付与する関連研究について述べる。情報抽出の方法としては自然文から情報を抽出する固有表現抽出や、本研究のように HTML のタグを目印に抽出するラップ⁴⁾ による抽出がある。HTML

の DOM 構造を利用して抽出する研究としては Tree ラッパがある¹⁾⁵⁾³⁾。山田⁷⁾等は Tree ラッパ拡張し、DOM 構造と LR ラッパを組み合わせた PLR ラッパを提案している。上記した研究ではラッパの導出を自動化するために、単純な構造を持った Web ページのみを対象としているため、今回のレシピの材料属性を付与するような単一のページが複雑な構造を持っている Web ページを対象とする事が出来ない。また、楠村等⁸⁾は Web ページのテーブル構造から入力された辞書データをブートストラップを利用して拡張する方法を提案した。この方法は辞書を拡張するには有効だが、あるページから抽出すべき属性値をすべて抽出するには、テーブル構造を持ったページのみしか対象としないため、大きく再現率を犠牲にしてしまう。

Califf²⁾等は自然文から Eliza-Like Pattern を学習し、属性値をメーリングリストに対して付与する RAPIER という手法を提案した。抽出器を生成する際に最も特殊な抽出器を生成し、ボトムアップに一般化を行い、学習する方式である点は本研究と類似する。RAPIER では最も特殊な抽出器の生成時に前後の 2 単語と POS タグ (品詞情報) で抽出しているが、こうした情報のみで属性値を特定する事が可能なドメインには有効であるが、そういったドメインはかなり限定されてしまう。

6. まとめと今後の展望

本研究では今まで抽出が困難であったページ内の小さなレイアウトの違いを表現するために、特殊ラッパ集合クラスと呼ぶ新しいラッパクラスを提案し、その導出方法を説明した。また、辞書マッチを利用して事例付与を行うことで、サイトに依存しない抽出器事例付与を行う事が出来た。それら特殊ラッパ集合を用いた実験を料理レシピから材料情報を抽出するタスクに対して適用し、本手法の有効性を検証した。

今後はラッパを構成する各要素 (タグや、タグ属性等) の中から、一般化すべき要素、一般化すべきでない要素を選択的に学習する方法を利用することで、さらなる精度向上と汎化性能の向上が出来るかを検証していきたい。例えば “(a href=hoge.com)” というタグが存在する時に、(a) と一般化して扱うことでより汎化された抽出器の導出が期待出来る。

参 考 文 献

1) Robert Baumgartner, Sergio Flesca, and Georg Gottlob. Visual web information extraction with lixto. In *The VLDB Journal*, pp. 119–

128, 2001.

2) Mary Elaine Califf and Raymond J. Mooney. Bottom-up relational learning of pattern matching rules for information extraction. *Journal of Machine Learning Research*, Vol.4, pp. 177–210, 2003.

3) S.Gupta, G.Kaiser, D.Neistadt, and P.Grimm. Dombased content extraction of html documents.

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

5) Ling Liu, Calton Pu, and Wei Han. XWRAP: An XML-enabled wrapper construction system for web information sources. In *ICDE*, pp. 611–621, 2000.

6) 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.

7) 山田泰寛, 池田大輔, 廣川佐千男. 交代数を用いた多言語 web テキストからの共通部分特定とラッパーの生成法. 情報処理学会論文誌, Vol.45, No.9, 2004.

8) 楠村幸貴, 土方嘉徳, 西田正吾. テンプレートの交叉と dom 構造の解析による情報抽出手法の提案. データ工学研究会, 2006.