帰納論理プログラミングを用いたWebラッパー自動生成

河野 碧† 西山 裕之† 大和田 勇人†

†東京理科大学大学院理工学研究科

1 はじめに

Web 上に存在する HTML や XML などの半構造化文書の情報を抽出し，構造を再構成する Web ラッパーが注目されている．これは Web ページから特定の箇所を自動的に抽出するためのサイトレイアウトを利用した抽出ルールおよび抽出プログラムを指す．しかし，これらの抽出ルールの導出は一般に困難であり，ページレイアウトごとに異なることなどから，人手によるルール導出はコストが高い [1]．そこで，抽出ルールを機械的に導出するための手法が研究されている [2][3]．

本研究ではユーザが抽出したい項目を入力し，その抽出ルールを帰納論理プログラミングを用いて導出し，そのルールを利用した Web ラッパーの自動生成手法を提案する．ユーザはサンプルページを拡張したブラウザで閲覧し，抽出したいテキストをクリックすることで抽出項目の入力を行う．この入力情報を用いて情報抽出ルールの学習時に用いる学習データの生成，学習，学習結果の解析，ラッパーの生成を行う．

情報抽出ルールの導出には，帰納論理プログラミングによる学習器である **Progol**[4] を利用した．2,3 ページのサンプルページから学習に必要となる情報を抽出し，Progol 学習用入力データを生成した．

本論文では，提案手法を利用した Web ラッパー自動生成システムを実装し，この提案手法の評価を行った．2 つのサイトに対して抽出ルール導出実験を行い，10 項目について 4,000 以上のページから情報抽出する実験を行った．

2 本研究における Web ラッパー

本研究ではユーザがサンプルページから選択した抽出項目に対する抽出ルールを Progol で導出し，Web ラッパーを生成する．ここでは本研究における Web ラッパーの利用形態および Web ラッパー生成の流れについて説明する．Web ラッパー生成の流れを図 1 に示す．

ユーザはサンプルページを拡張ブラウザで閲覧し，抽出したい項目をクリックする．この時，抽出ルール導出に必要となる情報をシステムが自動的に保存する．Web ラッパー生成システムはこの情報を元に Progol への入力データを生成し，学習を行い，抽出ルールを得，これを用いた Web ラッパーを自動で生成する．これは，情報を抽出したいページの URL 群を与えると抽出を行う Web ラッパーである．

Automatic Web Wrapper Generation and Induction using Inductive Logic Programming
Midori Kouno†, Hiroyuki Nishiyama†, Hayato Ohwada†
{†Graduate School of Science and Technology, Tokyo University of Science}

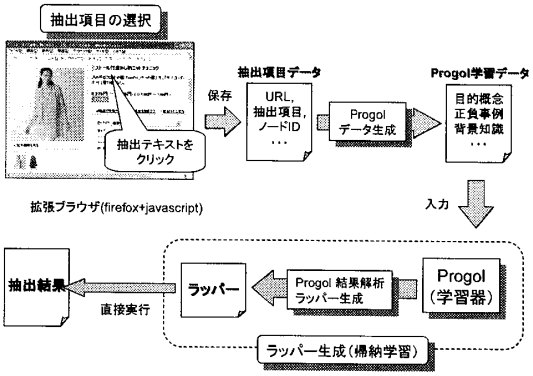

図 1: Web ラッパー生成の流れ

3 Web ラッパー生成

ここでは図 1 に示した Web ラッパー生成システムを実装した際の Progol への入力情報およびサンプルページからの抽出方法，Progol 学習データの生成手法について説明する．また，抽出項目入力のためのユーザインタフェースであるブラウザの拡張について述べる．

3.1 Progol への入力情報

Progol に学習させるために与える必要がある入力データは，モード宣言，正負事例，背景知識などである．モード宣言は問題概念の定義であるため，全抽出ルール導出問題において同一の定義となる．正負事例および背景知識はサンプルページの HTML ソースから抽出した．

モード宣言は，目的概念とそれを説明する概念を宣言するものである．本研究での目的概念は正事例を説明するルールの抽出を行うこと，目的概念を説明する概念には親タグの要素名，属性，属性値のペアの背景知識を用いることを宣言している．実際に抽出ルールを導出した際に用いたモード宣言を表 1 に示す．

表 1: モード宣言

【モード宣言】

```
:- modeh(*,node(+pageID,+nodeID))?
:- modeb(*,parentAttribute(+pageID,
    +nodeID,#attribute,#value,#depth))?
:- modeb(*,parentName(+pageID,+nodeID,
    #parent_name,#depth))?
```

【変数の説明】
- `pageID`:各サンプルページを区別する ID
- `nodeID`:各ノードを区別する ID
- `attribute`:タグに含まれる属性名
- `value`:`attribute` の属性値
- `depth`:テキストノードからの階層

事例は，サンプルページに含まれるテキストを与えた．正事例は抽出項目のテキストであり，負事例は正事例以外の全てのテキストである．学習には複数ページの正事例が必要であり，これをトレーニングサンプルと呼んでいる．ユーザは 2，3 ページ程度のサンプルページに対し，トレーニングサンプルを選択する．

背景知識としては，事例を囲むタグに関する情報を与えた．このタグは HTML ソースを木構造で表現した際に，抽出すべきテキストコンテンツの親ノードにあたる．ここでは事例を囲むタグを便宜上「親タグ」，親タグを囲む親タグを「先祖タグ」と呼ぶ．具体的には，親タグの要素名，属性，属性値のペア，階層の深さの情報を与えた．事例の親タグに関する情報だけではルールが生成されない場合，先祖タグに関して同様の情報を与えて再学習を行っている．

3.2 事例および背景知識の抽出

Progol への入力のうち，正負事例および背景知識はサンプルページの HTML ソースから抽出する必要がある．そのため，各情報がソースのどの位置に含まれるか把握する必要がある．本研究では，事例および背景知識に関する位置情報の把握には Java を用いて DOM(Document Object Model) 木を生成し，利用した．サンプルページの HTML ソースをパースして DOM 木を作成するが，このとき HTML 形式と XHTML 形式の混在や，不正な HTML コードの存在が問題となる．本研究では nekoHTML[5] および Xerces[6] を利用することで，不正な HTML の修正および DOM 木の生成を行い，事例および背景知識の抽出を行った．

3.3 Progol 学習データの生成

Progol が学習に必要とする入力ファイルには，まず，表 1 に示したモード宣言を記述する．次に，Xerces+nekoHTML から得られた DOM 木の XML 型オブジェクトからテキストノードのみの配列を生成する．この配列から正事例を探し出し，全てのサンプルページおよびノードに固有なノード ID をつけ，正事例の登録用雛型にあてはめて Progol 入力ファイルに記述した後，残った配列要素を同様に負事例として記述する．次に，全正負事例について，ノード ID ごとに親タグの要素名，属性，属性値のペア，階層の深さの情報を背景知識として記述する．

3.4 トレーニングサンプル入力インタフェース

トレーニングサンプル入力用ユーザインタフェースを拡張ブラウザを用いて実現した．ブラウザで表示されるテキストはマウスオーバーでハイライトされ，クリックでその URL およびノード ID が保存される．ブラウザには firefox を用い，ハイライトおよび保存には firefox アドオンである Greasemonkey[7] を用いて，javaScript で記述したユーザスクリプトを実行している．ハイライト処理は，ページのロード時に全テキストノードの親タグに SPAN タグを追加し，このタグのロールオーバー時に背景色を変更している．ノード ID の保存には，SPAN タグ追加時，ID セレクタにノード ID を設定，クリック時にこの ID と URL の data スキームを保存している．

3.5 学習結果の解析および Web ラッパー自動生成

本研究では，情報抽出したいページの URL を引数とし，実行すると情報抽出を行う Web ラッパーを生成した．Web ラッパーの雛型を用意し，抽出条件を埋め込むことで生成を行った．Progol の学習によって得られた抽出ルールは述語と変数にパースし，3.2 で述べた DOM 木に対する条件へと変換，雛型に埋め込んでいる．なお，DOM 木の生成に外部 jar([5][6]) を用いているため，生成した Web ラッパーのコンパイルおよび実行時にはそれらが必要となる．

4 評価実験および考察

提案手法を用いて実装したシステムの評価実験および考察について述べる．女性向けアパレル販売 e コマースサイト，「ベルーナ」（`http://www.belluna.net/`）および「IMAGE」（`http://www.st-image.com/`）のサイトを利用して評価実験を行った．10 の抽出項目に対する抽出ルールを求め，Web ラッパーを自動生成した．全ての抽出項目に対して抽出ルールを導出でき，自動生成された Web ラッパーを各サイト 2,000 ページに適用したところ，全てのページ，全ての項目に対して抽出することができた．これにより．本提案手法の有効性を示すことができた．

5 おわりに

本研究ではユーザ入力の抽出項目に対する抽出ルールを帰納論理プログラミングによる学習器である Progol を用いて導出し，そのルールに基づく Web ラッパーの自動生成手法を提案した．また，提案手法による Web ラッパー自動生成システムを実装し，評価実験を行うことでその有効性を示した．今後は他サイトへの適用実験を行う予定である．また，今後の展望としては，トレーニングサンプルの入力からラッパー生成・実行までをブラウザで完結させるシステムを目指す．

参考文献

[1] N. Kushmerick, Wrapper Induction: Efficiency and Expressiveness, *Artificial Intelligence*, Vol.118, pp.15-68, 2000.

[2] I. Muslea, S. Minton, and C.A. Knoblock, STALKER: Learning extraction rules for semistructured, Web-based information sources. *AAAI-98 Workshop on AI and Information Integration*, Technical Report WS-98-01, AAAI Press, Menlo Park, CA, 1998.

[3] C.-H. Chang and S.-C. Lui, IEPAD: Information Extraction Based on Pattern Discovery, *the Tenth International Conference of World Wide Web (WWW2001)*, pp. 4–15, 2001.

[4] S. Muggleton. Inverse entailment and Progol. *New Generation Computing, Special issue on Inductive Logic Programming*, 13(3-4):245-286, 1995.

[5] CyberNeko HTML Parser. `http://sourceforge.net/projects/nekohtml`

[6] Apache Software Foundation Xerces: XML parsers in Java and C++. `http://xml.apache.org/#xerces`

[7] Greasemonkey. `https://addons.mozilla.org/ja/firefox/addon/748`