Web 部分情報抽出システムとその応用
A System for Partial Information Extraction from Web Pages and Its Applications

韓 浩 † 徳田 雄洋 †
Hao HAN Takahiro TOKUDA
† 東京工業大学大学院情報理工学研究科
Graduate School of Information Science and Engineering, Tokyo Institute of Technology
{han, tokuda}@tt.cs.titech.ac.jp

Web ページ上の情報の全体ではなく一部分が必要となる場合がある．これらは単一のページの場合や，複数ページの場合や，同一時刻の場合や，一定期間内の場合などがある．これらの Web ページに対して部分情報抽出法とその応用を提案する．我々の方法は，Web ページの HTML 文書を分析し，ページのレイアウトパターンを獲得して，指定したい部分情報のパスを特定する．レイアウトパターンで Web ページの対応部分情報パスを選び，部分情報を抽出する．

1 はじめに

今日の Web 上の文書は，HTML，XML，マルチメディアデータ，サーバサイドプログラム，クライアントサイドプログラム，メタデータなど多種類の構成要素から構成されている．しかし，Web ページ上の情報の全体ではなく一部分が必要となる場合がある．例えば，Web 上の文字情報を分析し，索引情報を取り出すため，Web ページの文字部分だけを抽出したい場合である．また，ニュースページの特定部分を閲覧する場合もある．

そこで本研究は，ユーザが受け取る Web ページの HTML 文書を分析し，ページのレイアウトパターンを獲得して，指定したい部分情報のパスを特定する．レイアウトパターンで Web ページの対応部分情報パスを選び，その部分情報をテキストフォーマットで抽出する．本手法により，Web ページの対応部分情報パスを選び，部分情報をテキストフォーマットで抽出したり，オブジェクト部分のアドレス情報を抽出することができる．例えば，いくつかの Web サイトからニュースページの指定した特定部分のタイトル，記事と写真を抽出し，１つのページに表示させることができる．

本論文の構成は以下の通りである．第 2 章で Web ページの基本概念を説明し，第 3 章で提案する抽出方法について説明する．第 4 章で応用について述べ，第 5 章で関連研究を説明する．最後に，第 6 章で本研究をまとめ，今後の課題を述べる．

2 Web ページに関する基本概念

2.1 HTML と XML

HyperText Markup Language（HTML）は，現在の Web 上の基本的な記述言語である．しかし，終了タグが無かったり，不規則な入れ子関係になってしまっている HTML 文書もたくさん存在する．特に，プログラム上からそれらの情報を利用する場合は，一般的に，HTML 内のタグや文字列を基に情報を解析する必要があるため，簡単に扱うことができない．Extensible Markup Language（XML）[3] は明示的構造を持つため，XML で記述された文書は内容を解析しやすい．そこで HTML 文書のレイアウト分析のため，JTidy [5] を使い，HTML 文書を XML 文書に変更することにする．

2.2 XML の木構造とノード情報

XML 文書を木構造としてとらえ，各ノードの情報 (親子関係，兄弟関係，葉ノード数) を取得する．ただし，Web ページ上に実際に表示される情報 (文字，画像など) のみを対象とし，直接表示されない情報 (HTML 文書のコメント，javascript など) は対象外

とする．

2.3 パス（XPath）

XML 文書内の特定部分の位置を指定するために XPath [1] を利用する．XPath は，”body:0/form:0/table:1”のような形式で表現する．これは，ノード $< body >$ の 1 番目の子ノードである $< form >$ の 2 番目の子ノードである $< table >$ を表す．

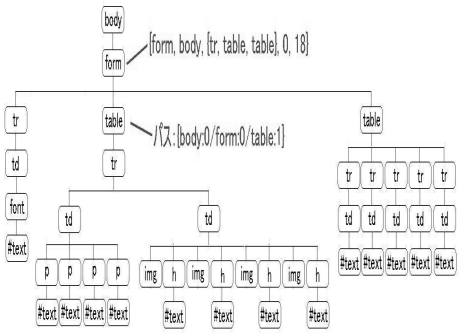

図 1: 木構造とノード情報とパス

3 部分情報の抽出方法

本論文で提案する Web ページからの部分情報抽出の手順は以下のとおりである．

- HTML 文書を分析し，Web ページのレイアウトパターンを獲得する
- 抽出したい部分情報の位置をユーザが指定する
- ユーザが指定した位置に該当するパスを特定する
- 部分情報を抽出する

以下で，詳細を述べる．

3.1 レイアウトパターン

3.1.1 モジュールファイルとレイアウトパターンの概念

同一のニュースサイト内の各ページのレイアウトなど (ニュースタイトルや記事，写真，広告の位置，文字のフォントやサイズ) は類似していることが多い．このことから，2 つの Web ページのレイアウトが類似している場合，その 2 つは同じ種類のページ

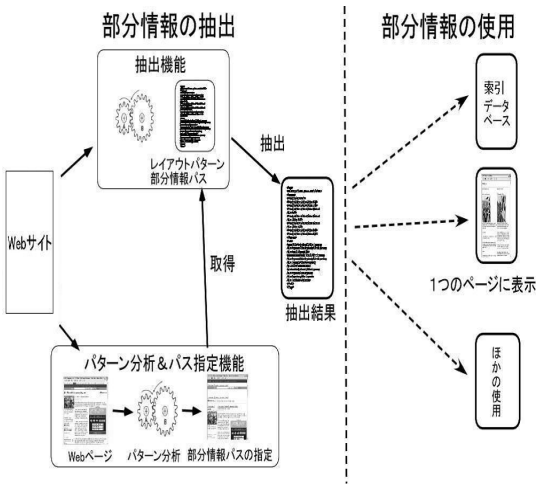

図 2: 部分情報抽出の手順

であると考えられる．同じ種類のページは，同一のモジュールファイル (JSP，ASP，PHP など HTML 埋め込み型のサーバサイドプログラム) から生成されることが多い．ここは，2 つの Web ページのレイアウトが類似しているかどうかを判断するため，レイアウトパターンの定義が必要である．レイアウトパターンとは，1 つのページに対応する全体木において，いくつかの部分木への全体木の分割の仕方，およびこれらの各部分木ルートへ至る全体木ルートからのパスのリストである．例えば，ページ画面を記事部分，画像部分，広告部分，関連リンク部分，ページフッタなどと部分木に分割し各部分木ルートへ至る全体木ルートからのパスのリストがレイアウトパターンとなる．2 つの Web ページのレイアウトパターンが類似している場合，その 2 つのページは 1 つのモジュールファイルから生成されたと考える．

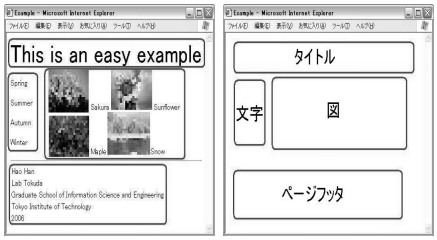

図 3: レイアウトパターン

3.1.2 レイアウトパターンの取得

まず，1 つのページを最大いくつに分割するかを決定する必要がある．我々は，多くの HTML 文書と画面のレイアウトを分析し，その結果，Web ページを木構造で表現した場合の葉ノード数の平方根とすることとした．ページの分割は以下の手順で行う．

MAX = 葉数の平方根;
nodelist = root;
L;
size = 0;
while (size + nodelist に含まれたノード数 < MAX) {
nodelist に含まれたノードは L に移動する;
L から node を削除；
node = L に含まれたノードの中に，葉数が一番大きいノード;
node の子ノードを nodelist に入る;
}

リスト L は，分割した部分木の根ノードを含める．

普通の場合は，HTML 文書の情報は $< body >< /body >$ ノードの間に含まれる．そして，$< body >$ ノードは HTML 文書の木構造の根ノードを指定する．そして，レイアウトパターンは，$< body >$ ノードから，L に含まれる部分木の根ノードまでのパスのリストである．パスのフォーマットは以下のように記述する．

$body : 0/tn1 : o1/tn2 : o2/.../tnN : oN$
tnN は，N 番目のノード（HTML のタグ）名
oN は，N 番目のノードの兄弟の間に順番
tnN-1 は，tnN の親ノード

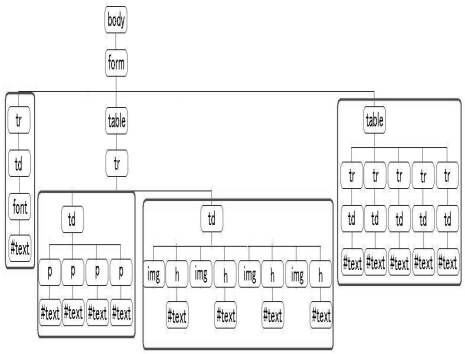

図 4: 木構造上のパターン取得

3.2 部分情報の指定

Web ページから部分情報を抽出するためには，抽出したい部分に関する具体的な情報が必要である．

3.2.1 データタイプ

指定される部分は，種類と構造タイプの指定が必要である．

Web ページから抽出できる情報には，テキストとオブジェクトの 2 種類がある．テキストとは Web ページ中の文字情報であり，例えばニュースページにおける記事 (文章) が挙げられる．オブジェクトとは Web ページ中の画像などの情報であり，例えばニュースページにおける写真が挙げられる．各オブジェクトは，それが存在するアドレスを情報として持つ．

構造は，単一データと連続データの 2 タイプがある．単一データとは，1 つの Web ページの中に 1 つだけ存在するデータである．例えば，ニュースページにおける記事のタイトルがある．一方，連続データとは，1 つの Web ページの中に複数出現するデータである．一般に，それらのデータは連続して出現することが多く，HTML データ (XML データ) を木構造で表現した場合，それらのノードは兄弟関係にあることが多い．連続データの例として，複数段落で構成されるニュース記事が挙げられる．

3.2.2 部分情報パスの特定

1 つのページから抽出したい部分を指定するためには，その位置を表すパスを特定する必要がある．その手順は以下のとおりである．

1. 分割された各部分について，それが抽出したい部分か否かをユーザが判断する
2. 抽出したい部分である場合には，指定する．抽出したい部分と抽出したくない部分が混在している場合は，抽出したい部分だけになるまで再分割し，指定する
3. 指定した部分の位置を表すパスを特定し，保存する

パスのフォーマットは以下で記述する．
body:0:ID/tn1:o1:ID1/tn2:o2:ID2/.../tnN:oN:IDN
tnN は，N 番目のノード（HTML のタグ）名
oN は，N 番目のノードは兄弟の間に順番
IDN は，N 番目のノードの ID
tnN-1 は，tnN の親ノード

3.3 部分情報抽出

Web ページのレイアウトパターンと抽出したい部分のパスを利用し，部分情報の抽出を行う．

3.3.1 レイアウトパターンによる Web ページの対応部分情報パスの選択

１つの Web ページは，１つのレイアウトパターンと対応する．１つの Web ページの部分情報パスリストは，１つのレイアウトパターンと対応する．部分情報のパスは，対応レイアウトパターンをもっている Web ページの部分情報の抽出にしか使えない．Web ページの部分情報の抽出は，対応レイアウトパターンで部分情報パスを選ぶことが必要である．そこで、注目する Web ページに対応するレイアウトパターンを判定する必要がある．

その方法は以下のとおりである．

1. Web ページを XML フォーマットに変更する
2. 木構造とノードの情報を計算する
3. 可能性のあるレイアウトパターンを１つずつ選ぶ
4. レイアウトパターン中の全てのパスが注目する Web ページの木構造の中にあれば，それが対応するレイアウトパターンである

同じ Web サイトの中には，類似した Web ページが数多く存在する．類似 Web ページは，ほとんど１つのモジュールファイルを使って生成されるページであり，レイアウトパターンが類似しているし，部分情報のパスも類似している．ここで，類似パスを以下のように定義する．偏差値 H を指定し，偏差値範囲以内のパスは類似パスと判断する．

body:0/tn1:(o1-H〜o1+H)/tn2:(o2-H〜o2+H)/.../tnN:(oN-H〜oN+H)

tnN は，N 番目のノード（HTML のタグ）名

oN は，N 番目のノードは兄弟の間に順番

tnN-1 は，tnN の親ノード

注目している Web ページが，レイアウトパターンの取得と部分情報パスの指定を既にしている他の Web ページと類似している場合，改めてパターンの取得とパスの指定を行う必要はない．1 つのページのパターンとパスの情報は，他の類似ページに対してもそのまま利用できる．

3.3.2 特定パスによる部分木抽出

対応レイアウトパターンを選択して，部分情報の抽出を行う．

body:0:ID/tn1:(o1-H〜o1+H):ID1/tn2:(o2-H〜o2+H):ID2/.../tnN:(oN-H〜oN+H):IDN

tnN は，N 番目のノード（HTML のタグ）名

oN は，N 番目のノードは兄弟の間に順番

tnN-1 は，tnN の親ノード

IDN は，N 番目のノードの ID 値

偏差値がある一定範囲内にあるパスは，抽出する部分である可能性がある．ID 値を利用し，その中で最も可能性の高いパスを選ぶ．

3.3.3 指定したデータタイプによる部分木情報のテキストフォーマット抽出

部分情報のパスを利用し，部分木を抽出する．抽出した部分木から，以前に指定したデータタイプに従い，部分情報をテキストフォーマットで抽出できる．

テキストタイプの情報は，部分木の葉のノードバリューである．オブジェクトタイプの情報は，部分木のオブジェクト対応のノードのアトリビュートバリューであり．例えば，画像情報は，$< img >< /img >$ ノードの”src”アトリビュートのバリューである．

単一データは，そのまま部分情報パス対応の部分木を取る．連続データは，部分情報パスに対応する部分木について，部分木の根ノードと兄弟関係にあるノードのうち，ノード名と ID 値が同じノードを根とする部分木も部分情報として抽出する．例えば，ニュースの記事部分は複数の段落で構成されており，1 つの段落を指定したら，他の段落も部分情報として抽出する．

4 応用

本章では，Yahoo! News [10] と CNN.com [2] のトップニュースのページを例とし，各ページから部分情報を抽出して 1 つのページで表示する手順を示す．

1. 抽出したい部分のデータタイプを定義する．ニュースタイトルは単一テキストタイプであり，ニュースの写真は単一オブジェクトタイプであり，ニュース記事部分は連続テキストタイプである．

```
<init-param>
  <param-name>SINGLEFIELD</param-name>
  <param-value>newstitle</param-value>
</init-param>
<init-param>
  <param-name>CONTINUALFIELD</param-name>
```

```
    <param-value>newscontents</param-value>
  </init-param>
  <init-param>
    <param-name>OBJECTFIELD</param-name>
    <param-value>picture</param-value>
  </init-param>
```

2. Yahoo! News のニュースページを分析し，部分情報を指定する．

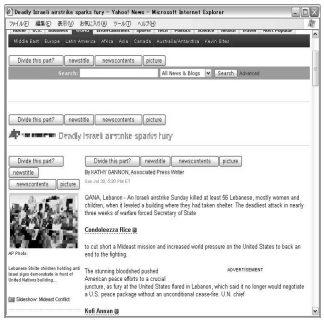

図 5: パターンの取得と部分情報の指定

```
<Page>
<URL>http://news.yahoo.com/s/</URL>
<Pattern>
<P>body:0/div:0</P>
<P>body:0/div:1/div:0/div:0</P>
<P>body:0/div:1/div:0/div:1</P>
<P>body:0/div:1/div:0/div:2/div:0
/div:0</P>
<P>body:0/div:1/div:0/div:2/div:0
/div:1/div:0</P>
<P>body:0/div:1/div:0/div:2/div:0
/div:1/div:1</P>
<P>body:0/div:1/div:0/div:3</P>
<P>body:0/div:1/div:0/div:4</P>
</Pattern>
<Path>
<newstitle>body:0:null/div:1:ynwrap
/div:0:yncont/div:2:ynbody/div:0:ynstory
/div:0:null</newstitle>
<newscontents>body:0:null/div:1:ynwrap
/div:0:yncont/div:2:ynbody/div:0:ynstory
/div:1:ynmain/div:0:storybody
/p:1:null</newscontents>
<picture>body:0:null/div:1:ynwrap
/div:0:yncont/div:2:ynbody
/div:0:ynstory/div:1:ynmain
/div:1:sidebar</picture>
</Path>
</Page>
```

3. Yahoo! News のトップページの URL を入力する．

```
<Info>
<URL>http://news.yahoo.com/</URL>
</Info>
```

4. CNN.com のページについて，ステップ 2 とステップ 3 と同様の作業を行う．

5. 自動的にニュースを抽出し，1 つのページにデフォルトレイアウトで表示する．

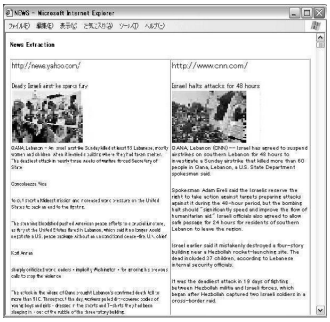

図 6: 2 つの Web ページ部分情報の 1 ページ表示

5 関連研究と比較

Web ページから部分情報を抽出する試みはいくつか存在するが，これらは，大きく分けて，文字列指向手法と木構造指向手法の 2 つがある．

文字列指向手法とは，Web ページを文字列とみなし，文字列のマッチング操作により指定範囲内の部分を抽出する手法である．例えば，I know システム [4] では，2 つのキーワードを指定し，その 2 つのキーワードの間の部分を抽出する．LR-wrapper [8] では，抽出したい文字列の左右のデリミタを定義し，部分情報を抽出する．

木構造指向手法とは，Web ページを木構造とみなし，パスにより指定された部分を抽出する手法である．例えば，XSLT [6] は，指定したパスに対応するノードを特定し，そのノードの情報を出力する．Internet Scrapbook システム [7] では，ユーザが Web ページ中の一部分を指定し，そのパスとタイトルをマッチングして部分抽出を行い，パーソナルページを作成する．PSO [9] では，ユーザがパスを指定することにより，Web ページの一部を抽出する．

本研究の方法は，これらの方法と比較し，以下の特徴がある．

1. レイアウトパターンの取得とページの分割は自動的に行われ，抽出したい部分の指定も簡単である．ユーザは，プログラミングと HTML に関する知識を必要としない．

2. レイアウトパターンを使い，たくさんの Web ページの対応部分情報パスを選び出せる．

3. レイアウトパターンと部分情報パスの「類似」概念を使い，抽出できる Web ページの範囲が大きくなる．

4. データタイプを使い，部分情報をテキストフォーマットで抽出でき，データを再利用することができる．

6 まとめ

本論文では，Web ページからの部分情報抽出手法を提案した．これは，まず Web ページのレイアウトパターンを自動的に取得し，そのパターンを利用して，抽出したい部分を指定した後，その部分を抽出する手法である．しかし，HTML 文書の構造が複雑になると，うまく抽出できない場合がある．例えば，javascript で動的に生成される情報に対応する部分木 (パス) は，javascript 部分であるが，本手法では javascript 部分を抽出の対象としていないため，抽出することができない．また，抽出したい部分情報の表示位置がページによって変化する (表示位置が一定ではない) 場合も，抽出できない．

今後の課題を挙げる．まず，現在のアルゴリズムを改善し，より多くのパターンに対応できるようにする．また，部分情報の指定や抽出を容易にするための GUI についてさらに考える必要がある．

参考文献

[1] James Clark and Steve DeRose. XML Path Language(XPath) Version 1.0. *http://www.w3.org/TR/xpath*, 1999.

[2] CNN.com. *http://www.cnn.com/*.

[3] Extensible Markup Language(XML). *http://www.w3.org/XML/*, 2006.

[4] I know. *http://i-know.jp/*.

[5] JTidy. *http://jtidy.sourceforge.net/*, 2004.

[6] M. Kay. XSL Transformations(XSLT) Version 2.0. *http://www.w3.org/TR/xslt20/*, 2002.

[7] Yoshiyuki Koseki and Atsushi Sugiura. Internet scrapbook: Automating web browsing tasks by demonstration. In *ACM Symposium on User Interface Software and Technology*,pages 9-18, 1998.

[8] N. Kushmerick. Wrapper induction: Efficiency and expressiveness. *Artificial Intelligence*,118(1-2):15-68, 2000.

[9] Tetsuya Suzuki and Takehiro Tokuda. Path set operations for clipping of parts of web pages and information extraction from web pages. In *Proceedings of the 15th International Conference on Software Engineering and Knowledge Engineering*, pages 547-554. Knowledge Systems Institute, 2003.

[10] Yahoo! News. *http://news.yahoo.com/*.