D_036

Webデータの自動抽出とデータ変換

Automatic extraction and conversion of data from the WWW

小沼　寛明 *　　全　眞嬉 *　　徳山　豪 *
Hiroaki Onuma　　Jinhee Chun　Takeshi　Tokuyama

1　はじめに

複数のwebページからデータを自動的に抽出しデータ変換を行いXMLにまとめる．本稿では，リストページとディテールページの構造を持ったwebページ群から情報を自動抽出するLermanらの手法[1]の実装と改善方法を提案し実験を行った．Lermanらの手法ではタグに挟まれている文字列をextractとし，リストページとディテールページの完全一致する文字列のみを抽出し，CSPの制約式を立ててレコード分割に用いる．完全一致した文字列のみ抽出されるため，情報ロスが生じる．

本稿では上記の問題を回避するための改善法を提案する．

- 完全に文字列が一致しなくとも共通のextractとして抽出を行う．
- さらに連続条件を変更し，文字列が連続で現れなくてもextractのレコードへの割り当てが入れ子にならない限り同じレコードに割り当てることを許す．
- 抽出されたデータをXML形式に変換する．

提案する方法の妥当性を調べるため実装し評価を行う．

2　問題の概要

現在，インターネットからは様々な情報を得ることができ，その多くはHTMLで書かれたwebページとして提供されている．HTMLは，webページをどのように見せるかを記述する言語であり，人間が見て判りやすいように工夫されている．webデータに対してラベル付け等の仕掛けが必要であり，現状ではコンピュータがwebページから直接情報として扱うことは困難である．したがって，webデータを情報として利用するために，webページからデータを抽出し，その関連を見出し，表形式のデータ変換を行いデータベースを構築する．本稿では，リスト形式の商品一覧ページをリストページと呼び，各商品の詳細な情報が記載されているページをディテールページと呼ぶ．この二種類のページからは異なる情報を得ることができる．ディテールページからは，リストページには記載されてない細かい情報が，一方リストページからは，あるカテゴリに属する商品のリストや，販売ランキングの順位などディテールページからは得られない情報を得ることができる．

リストページには商品リスト，チェックした商品の履歴，お勧め商品，広告等いろんな情報が提示されている．リストページから商品リスト情報を抽出するには，さまざまな情報から商品リストの特定，また一つの商品を表している範囲の特定をしなければならない．

この問題に対し，考えられるひとつの手法はハイパーリンクを利用する方法である．しかし，webページはヘッダ部やフッタ部，広告などにリストとは関係のない数多くのハイパーリンクがあるためハイパーリンクのみを用いての特定は難しい．本稿では，上記の特定作業にリンクの情報は利用せず，リストページとディテールページに共通する内容(データ)に着目して情報の抽出とデータ変換を行う．

*東北大学大学院情報科学研究科システム情報科学専攻

3　情報抽出と変換のアルゴリズム

リストページとディテールページの枠組みを利用したデータ変換に関して，Lermanら[1]がCSP(Constraint Satisfaction Problem:制約充足問題)を用いた手法を提案している．以下はその手順である．

- テンプレートの探索およびスロットの特定
- CSPの条件式を立ててこれを解く

3.1　テンプレートの探索およびスロットの特定

上記のような企業のサイトの多くはリストページやディテールページを企業が独自に持つデータベースと連携し，webサーバがあるテンプレートを基に自動的に作り情報を提供している．我々の望む各商品の情報はテンプレートではなくそれ以外の部分(スロット)に含まれていると考えられる．スロットを特定するためには，テンプレート探索アルゴリズム[2]を用いる．

3.2　CSPによるレコード分割

リストページとディテールページに共通した内容(データ)を以下extractと呼び，これをスロットから抽出し，レコードというものに割り当てる．同じレコードに割り当てられたextractはある同一の商品に関するデータであることになる．この作業をレコード分割と呼ぶ．なお，[1]では，THMLタグでない(HTMLタグにはさまれた)部分を1つのextractとして抽出しておりまったく同じ文字列でなければ共通のデータとして抽出しない．例えば，「980円」，「980円(税込み)」の2つのデータは，共通のデータではないとみなされる．

以後，x_{ij}を，i番目のextract E_iがレコードr_jに割り当てられるとき$x_{ij}=1$，E_iが レコードr_jに割り当てられないとき$x_{ij}=0$となる割り当て変数とする．CSPの制約条件として以下の3つを採用している．

独立条件　各extractE_iは多くとも1つのレコードr_jに割り当てられる．

$$\sum_j x_{ij} \leq 1$$

連続条件　連続した番号のextractのブロックだけが，同じレコードに割り当てることができる．ある$k<n<i$を満たす数nがあり，$x_{nj}=0$ならば$x_{ij}+x_{kj}\leq 1$.

位置条件 2つのextractが，あるディテールページの同じ位置に現れる(つまり同じデータ)ときは，それらは異なるレコードに割り当てられなければならない．

4 改良

本稿では上記のレコードへの割り当ての手法を改良する．改良点は以下の2つである．

4.1 部分一致の導入

以下のような条件でも共通であるとみなして，extractとして抽出しレコードに割り当てる．

改善1 リストページのあるextractをE_l，ディテールページのあるextractをE_dとする．「E_lがE_dの部分文字列である」あるいは，「E_dがE_lの部分文字列である」とき，E_lとE_dの文字列の長さをそれぞれ，L_l，L_dとし，ある正整数kを用いて$|L_l - L_d| \leq k$と表せるときだけ共通であるとみなす．

例えば，E_lが「980円」，E_dが「980円(税込み)」，$k = 5$であった場合，E_dがE_lを含み，その文字数の差が5なのでこの2つは共通のextractとみなすことになる．文字数の差に制限を設けたのはほとんど関係のない2つのデータを同じとみなすことを減らすためである．

4.2 連続条件の改善

同じレコードに割り当てるべきextractのブロックのうち両端ではないextractが別のレコードに割り当てられてしまうとその両端は同じレコードに割り当てることができない．また，同じレコードに割り当てるべきextract系列の間にそのレコードと無関係なextractが割り込み，そのextractがどのレコードにも割り当てられなければ，その前後のextractは同じレコードに割り当てられなくなってしまう．これらを改善する方法として以下のように連続条件を変えることを提案する．この変更によって，連続した番号のextract系列しか1つのレコードに割り当てられないのではなく，番号は連続していなくとも，他のレコードに割り当てられるextract系列と入れ子になったり交わったりしない連続したブロックのextract群を1つのレコードに割り当てることができるようになる．

改善2 $k < n < i$を満たす，全てのextractの番号n，ページjに現れるextractの番号k，iに対して，

$$x_{ij} + x_{kj} - x_{nj} + \sum_{l \neq j} x_{nj} \leq 2$$

$\sum_{l \neq j} x_{nj}$は，extractE_nがレコードrj以外のレコードに割り当てられるとき1になる．つまり，E_nがレコードrj以外のレコードに割り当てられるときは，もとの連続条件と同じものを表し，どのレコードにも割り当てられない（$x_{nj} = 0$かつ$\sum_{l \neq j} x_{nj} = 0$）のとき，$E_i$，$E_k$がともにレコード$rj$に割り当てることができるようになる．

5 実験

前節の改善法をJavaで実装し，いくつかのサイトで実行して実験を行った．実験には，対象のサイトからリストページとディテールページをダウンロードしてきて行っている．CSPはフリーウェアのソルバーであるGLPK [3]を用いて，できるだけ多くのextractが割り当てられるように線形計画問題として解き，レコードへの割り当てを求めている．

表1: 連続条件改善の結果

ディテールページ	No	extract	No	extract	No	extract	No	extract	No	extract
1.htm	1	49920728 11355	2	XGA 対応…	4	LCM- …	5	発売日: 2005年 12月下旬	6	1024ドット ×768…
2.htm	7	49571800 59068	8	電源 内蔵…	10	LCD- …	11	発売日: 2005年 11月下旬	12	XGA対応 15型…

5.1 部分一致の導入の結果

実験の結果，部分一致した文字列も共通のextractとして抽出するようにしたことにより，より多くのデータをレコードに割り当てることができるようになった．

しかし，制限を緩めたことであまり関係のないデータも共通のextractとしてレコードに割り当ててしまうという場合もあった．文字数の差kでこれをこれを改善しようとしたが，更に他の改善方法の考案が今後の課題である．

5.2 連続条件の改善の結果

表1はディスプレイを販売するある企業のサイトでの結果である．一番目，二番目のレコードそれぞれ，3,9番目のextractが割り当てられていないが，その前後のextract系列が同じレコードに割り当てられ，正しく動作していることが確認できた．

6 終わりに

提案する改善法を実装し，実験により自動抽出精度が上がった事を示した．今後の課題としては，さらにレコード分割の精度を上げる事を考える．

参考文献

[1] Kristina Lerman,Lise Getoor,Steven Minton,Craig Knoblock. Using the Structure of Web Sites for Automatic Segmentation of Tables, *Proceedings of the 2004 ACM SIGMOD international conference on Management of data* 119-130.

[2] K.Lerman, S.Minton,and C.Knoblock. Wrapper maintenance: A machine learning approach. Journal of Artificial Intelligence Research, 18:149-181, 2003.

[3] GLPK <http://www.gnu.org/software/glpk/glpk.html>