属性の共起関係に着目したWWWからの効率的なXMLデータ抽出

Ngo Sy Viet Phu† 天笠俊之†‡ 北川博之†‡
†筑波大学大学院システム情報工学研究科 ‡筑波大学計算科学研究センター

1 はじめに

近年，WWWの普及により，膨大な情報がウェブ上に蓄積されている．しかし，これらの情報が構造を持たないため，ウェブの情報源から構造化されたレコードを抽出することは重要である．このため，ウェブからのレコード情報を抽出する手法が注目され，この数年活発に研究されている．

一方，XMLは標準データフォーマットとして広く普及し，多くの情報がXML形式でネットワーク上を流通している．このため，さまざまな情報源からXML形式のデータを抽出することが求められている．既存のレコード抽出手法は基本的に二項関係，あるいは単純な構造を持つレコード情報に特化しており，XMLのような複雑な構造を抽出する手法はあまり議論されていない．このため，我々は，WWWを情報源とした単純なレコード抽出手法を組み合わせることによってより複雑なXMLデータを抽出する手法を提案した[4]．しかし，この手法では，レコード抽出手法が独立に処理を行うため，最終的にXMLデータを再構成することができるレコード数が少ないという問題点がある．そこで，複数のレコードを組み合わせる際のキーとなる属性値を積極的に利用し，XMLデータとして再構成可能なレコードを効率的に取得する手法を提案する．

2 関連研究

2.1 レコード抽出手法の概要

DIPRE[1]は，少数の種(Seed)レコードを入力として与え，まず種レコードが含まれる文書を検索エンジンにより，取得する．得られた文書から，種レコード周辺の特徴的なパターンを抽出し，次はそのパターンを使って検索エンジンに再度問合せを行う．得られた文書から新たなレコードを抽出するとともに，新たなパターンを取得する．この処理をパターンとレコードの双方を増やしながら繰り返すことによって，大量のレコード情報を抽出する．

Snowball[2]はDIPREを拡張したアプローチである．Snowballでは，固有表現抽出器を利用することにより，文書に含まれる固有表現に対してタグ付けを行う．次に．固有表現を用いた抽出パターンを生成することによりた精度の高いレコードを抽出することができる．

また，McDonaldら[3]は，多項関係の抽出の問題に対して，多項関係から二項関係に分解し，既存のレコード抽出を行う．得られたレコード集合から実体グラフを作成し，最大クリークを探索する．次に最大クリークを元に多項関係を再構築する．

2.2 ウェブからのXMLデータ抽出手法

ウェブからのXMLデータ抽出手法の手順として以下のようになる．

(1)システムの入力として，XMLデータとそのデータに関するスキーマ情報(DTD, W3C XML Schema, など)をシステムに入力する．図1 に種XML データ、図2 に種XML データの文書型定義(DTD) を示す．

```
<CompanyInformation>
  <Company>
  <CompanyName>Apple</CompanyName>
    <CEO>Steve Jobs</CEO>
    <Location>Cupertino</Location>
    <Products>
      <Name>MacBook</Name>
      <Name>MacBook Pro</Name>
    </Products>
  </Company>
  <Company>
    <CompanyName>Microsoft<CompanyName>
    <CEO>Steve Ballmer</CEO>
    ...
  </Company>
  ...
</CompanyInformation>
```

図1 種XMLデータの例.

```
<!ELEMENT CompanyInformation (Company*)>
<!ELEMENT Company
(CompanyName,CEO?,Location?,Products?)>
<!ELEMENT CompanyName (#PCDATA)>
<!ELEMENT CEO (#PCDATA)>
<!ELEMENT Products (Name*)>
<!ELEMENT Name (#PCDATA)>
```

図2 種XMLのDTD.

スキーマ情報からスキーマグラフを導出する．スキーマグラフは，各ノードが要素あるいは属性を表し，枝は要素と要素の親子関係，もしくは要素と属性の所属関係を表している．図3 に文書型定義から導出されたスキーマグラフの例を示す．

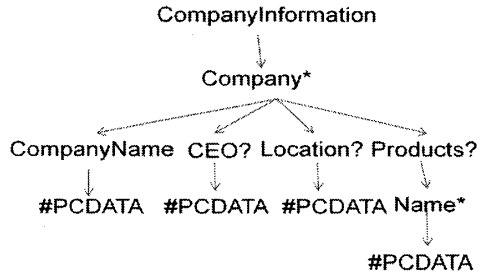

図3 スキーマグラフ.

(2) XML データを（スキーマ情報を利用しながら）関係表に写像する．XML データを関係表に写像する際に既存手法を利用する．例えば，図2 のXMLデータに変換された結果は図4に示される．

Company

CompanyName	CEO	Location
Apple	Steve Jobs	Cupertino
Microsoft	Steve Ballmer	Redmond

Products

CompanyName	Name
Apple	MacBook
Apple	MacBook Pro

図4 種XML データから写像して得られた関係表.

(3) 得られた関係表を二項関係に分解する．分解では各関係表の主キーを軸に，主キー属性以外の属性をそれぞれペアにして分解する．上記の例ではCompanyの関係表を以下の二つの二項関係に分解する．

Company_CEO (CompanyName, CEO)

Company_Location (CompanyName, Location)

(4) 分解して得られた二項関係を例示レコードとして，レコード抽出アルゴリズムに入力し，多数のレコードを取得する．得られた二項関係を，元の関係表を経由してXML データとして再構成し，利用者に返却する．

しかし，この手法では，レコード抽出手法が独立に処理を行うため，最終的に XML データを再構成することができるレコード数が少ないという問題点がある．

3 提案手法

本手法では，XMLデータの抽出を行う際に各二項関係に対して，属性の共起関係に着目して抽出を行う．これにより再構築可能なレコード数を改善する．

(1) 2.2節で説明したウェブからのXMLデータ抽出手法のステップ1から3までを実行する．

(2)パターンのランキング：ブートストラップ型のレコード抽出法ではパターンとレコードの双方を増やしながら繰り返すことによって，大量のレコード情報を抽出する．そこで，良いパターンは多くのレコードを抽出できることが考えられるため，多くのレコードを抽出できるパターンに対して，高いランクを与えるようなランキングを行う．

(3)再構築可能なレコード数の取得:2.2節で説明した例では，例えば，Company_CEO (Google, Eric Schmidt)が抽出されている．しかし，GoogleのCompany_Locationのレコードが抽出されない場合はGoogle会社のレコードは再構築不可能である．その時に，Company_Locationの二項関係で抽出されたパターンを用いて，Googleの所在地を検索する．Company_Locationの二項関係のパターン例として，以下のパターンが存在とする．

[at, .?*, headquarters in, .?*, and]

Googleをパターンに追加し，Googleの所在地を探す．検索エンジンに投入するクエリーは以下のようになる．

"at Google headquarters in .?* and"

この処理により，2.2で説明したウェブからのXMLデータ抽出手法の再構築可能なレコード数が改善できると考えられる．

すべてのパターンについてこの処理を行うと膨大な時間が予測できる．そのため，パターンをランキングして，上位のパターンだけを利用する点がポイントである．

4 評価実験

提案手法の妥当性を評価するために実験を行った．検索エンジンにはYahoo Search Boss を用いた．実験に使用したPC のCPU はCore Duo (2.33GHz) であり，主記憶容量は2GB である．本実験では，会社名，所在地，最高営業責任者を記録したXML データを例示レコードとして与えた．例示レコードを図5 に示す．この種データを関係スキーマに写像した結果，以下の二つの表が得られた．

Company_CEO (ComName, CEO)

Company_Location (ComName, Location)

本実験では，ウェブからのXMLデータ抽出手法の再構築可能なレコード数と改善した手法の再構築可能なレコード数に比較する．また，個別の関係表についてDIPRE によってレコード抽出を行う．

```
<CompanyList>
  <Company>
    <ComName>Microsoft</ComName>
    <CEO>
      <CEOName>Steve Ballmer</CEOName>
    </CEO>
    <Location>Redmond</Location>
  </Company>
  <Company>
  ...
<CompanyList>
```

図5 実験で用いた種XMLデータ

4.1 結果

4 回の抽出を行った際に抽出されたパターン数とレコード数を図6に示す．

抽出回数	パターン数	レコード数
1回目	24	232
2回目	56	532
3回目	158	1794
4回目	173	1805

抽出回数	パターン数	レコード数
1回目	13	97
2回目	27	285
3回目	89	486
4回目	101	504

図6 抽出パターン数とレコード数（左：(ComName, CEO)，右：(ComName, Location)）

4.2 評価

・(ComName, CEO) の二項関係については，合計1805 レコードが抽出された．結果を人手で精査した結果18.2% (340 レコード) はノイズであった．

・(ComName, Location) の二項関係については，合計258 のレコードが抽出され，そのうち21.6% (109 レコード) はノイズであった．

・個別に抽出された二項関係を元の関係表に復元する際，CEO, Location 双方の情報を兼ね備えていたレコード数は109 であった．

・また，改善手法では，CEO, Location 双方の情報を兼ね備えていたレコード数は452 であった．この結果により，最善手法は既存手法に比較すると約4倍の再構築可能なレコードを抽出できた．

5 まとめと

本稿では，属性の共起関係に着目したWWWからの効率的なXMLデータ抽出手法について提案し，評価実験により，提案した手法の有効性を確認した．

謝辞 この研究の一部は，科学研究費補助金特定領域研究 (#21013004)，若手研究(B) (#21700093) によるものである．ここに記して謝意を表す．

参考文献

1) Brin, S.：Extracting Patterns and Relations from the World Wide Web, Technical Report 1999-65, Stanford InfoLab (1999).

2) Agichtein, E. and Gravano, L.：Snowball: extracting relations from large plain-text collections, Proc. ACM DL, pp. 85-94 (2000).

3) McDonald, R., Pereira, F., Kulick, S., Winters, S., Jin, Y. and White, P.：Simple algorithms for complex relation extraction with applications to biomedical IE, Proc. ACL, pp. 491-498 (2005).

4) Amagasa, T., Phu, N. and Kitagawa, H.：A Scheme of XML Data Extraction from Web Pages, SIGDD(2009).