WebAPIから得られるXML文書統合手法の検討

秋永　智†　　木村　昌臣††

芝浦工業大学大学院工学研究科†　　芝浦工業大学工学部情報工学科††

1. はじめに

近年, XMLを利用したデータ交換が頻繁に利用されている. XMLはデータ構造やタグ名などが自由に定義できる反面, 構造やデータの内容を把握するにはXMLの中身を直接閲覧する, 提供元の仕様を参照するなどの必要があるが, コンピュータだけで自動的に処理を行うのは困難である. また, XML中に存在する要素の内容は様々であり, 内容を考慮した上で複数のXMLを統合するのは難しいといえる. しかしXMLの統合をすることで, Webサービスを複数組み合わせることで新たなサービスを作りあげるマッシュアップを作成する際にXML別にプログラムを列挙する手間が無くなることや, 新たに出来たWebサービスをマッシュアップに加える場合などに容易に組み込むことが可能になると考えられる. 横田ら[1]は複数のXML文書間の類似度を高速に検出し, 類似度が高い場合にはXML文書を統合する手法を考案しているが, 同じ内容を有するか否か等の意味的な点を考慮しておらず, 十分な統合が出来ているとは言い難い.

そこで本研究ではWebAPIから得られるXMLを対象とし, XML中に含まれる同じ内容の要素をまとめて一つのXMLに再編成することで, XML別に処理を変える手間を省くとともに, XMLの情報密度を高めてより有用なデータとする手法の提案を行う. また, それを行うために複数のXML中から同じ内容の要素を同定するための手法を提案する.

2. 研究内容及び提案手法

上に述べたように, XMLはデータ構造の定義が自由なため, 全てのXMLに存在する要素もあれば一つにしか存在しない要素もある. 例としてぐるなびWebサービス[2]から提供されるXMLと ホットペッパーWebサービス[3]から提供されるXMLを挙げる. これらは共にレストラン情報を扱っているが仕様が異なるため別物と見なしてXMLごとに処理を変える必要がある. また各XMLに存在する要素の違いとして, レストラン名を示すタグはぐるなびでは<name>としているがホットペッパーでは<ShopName>としていたり, ぐるなびではファックスの情報を示す<fax>があるが, ホットペッパーでは存在していない. XMLを統合する際に内容が重複してしまわないようにnameとShopNameを一つにまとめる必要があり, もちろんファックス情報であるfaxも項目の一つとして定義する必要がある. 片方にしかないからといって項目を定義しないと情報量が減ってしまうためである. このようにして同じ内容の項目は一つにまとめ上げ, そうでない項目をそれぞれ定義しそれを要素として新たなXMLを生成する. 図1のように同じ内容の要素が同じ色で塗り分けられているとするとこの時新たにXMLを生成するときに同じ内容の要素を定義するのは無駄であるため必要なデータは計6種類になると考えられる.

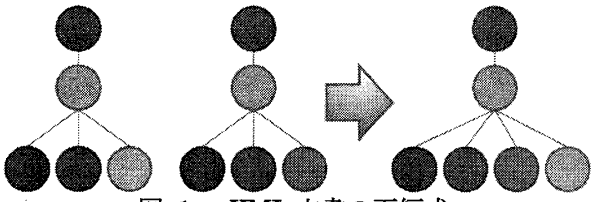

図1　XML文書の再編成

このため, 新たにXMLを定義する際には要素の重複を避けるために同じ内容の要素を同定する必要がある. そこでXMLにある各要素のテキストを解析して要素の特徴量を算出し, それを用いてテキスト間の類似度を算出することで様々な種類の要素の中から類似要素を同定する. 同じ内容の要素の同定手法を以下に示す.

2.1. 文字種走査

本研究における文字種とはひらがな, カタカナ, 漢字, アルファベット, 数字, そして記号の全6つの種類を指す. 例えば電話番号なら数字と場合によっては記号で構成され, URLならば記号とアルファベットで構成されているといった具合である. 同じ内容の要素が他のXML中でも存在していた場合それも同じ文字種で構成されていると予想される. そこで, XML中のテキストを1文字ずつに分け, その文字の種類を判別し, 各文字種の個数をカウントしたものをベクトルの各要素とする. 二つのベクトルのなす角度を求めるコサイン尺度（式1）を用いてテキスト間の類似度を算出し, 類似した特徴を持つ要素を見つけ出す.

$$\cos\theta_{a,b} = \frac{\vec{w}(a)\cdot\vec{w}(b)}{\|\vec{w}(a)\|\|\vec{w}(b)\|} \quad \cdots(1)$$

数式1　ベクトルのコサイン尺度

2.2. N-gramを用いた頻出文字列の抽出

An inquest of integration method for XML obtained from WebAPI

†Satoshi Akinaga, ††Masaomi Kimura

†Graduate school of Shibaura institute of technology

††Shibaura institute of technology

N-gram とは検索対象を単語単位ではなく一定の N 文字単位で分解し，その分解した文字列の出現頻度を求める方法である．WebAPI では図 2 のように取得した XML 中に 1 件分の商品情報やレストラン情報など複数格納されている．例として図 2 では books の子要素に book という情報が複数あり，<book>タグの子要素として<title>，<author>，<isbn>，<publisher>という四つのタグが存在していたとする．

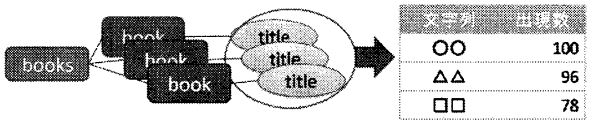

図 2　N-gram を利用した解析

あるタグ（図 2 では title）が持つテキストを全て N-gram で解析し，テキスト中の頻出文字を求める．それを他のタグの解析結果と比較することで類似した文字が頻出しているタグを求める．上記の手法を組み合わせて同じ内容を示す要素の同定を行う．

3. 評価方法

提案手法を用いてプログラムを実行し，再現率及び精度と実行速度の評価を行う．ここで言う再現率とは重複しているデータがあった場合，それが重複しているデータとしてコンピュータが認識できているか，精度とは同定したデータが正しいか，つまり同定したデータが同じ内容を示しているかを指す．なお，利用する WebAPI としてぐるなび Web サービス [2]とホットペッパーWeb サービス [3]から得られる XML を利用した．

4. 評価及び考察

類似項目の取得率は図 4 の通りである．

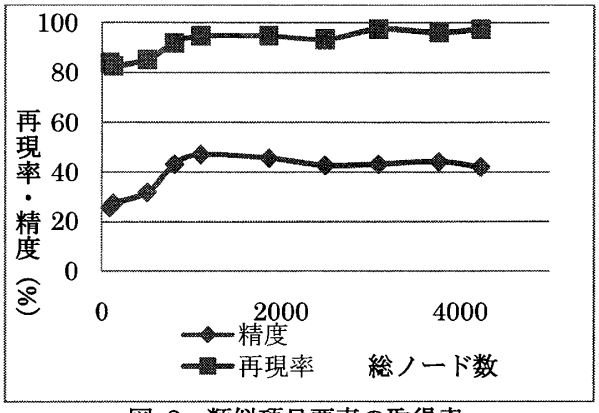

図 3　類似項目要素の取得率

再現率に関しては 100%に近い値が．精度は 50%弱で最も良いときで 46.9%の取得率であった．ノード数が少ないと解析対象であるテキストも少なくなってしまうため十分な再現率・精度共にふるわなかったがノード数 1500 程度から一定値を得られた．しかしある点を境にほとんど変わらなくなり，何らかの別の手法を考える必要がある．評価に利用した XML はレストラン情報を扱う物であったが，住所，緯度，経度など表記に大きな差がない情報は正しく同定が行われていた．レストラン名などは様々な表記があるためベクトルの値がある程度均一に振り分けられ，その結果正しい同定が行われたと思われる．一方，表記揺れが激しい営業日や祝日を示す要素，たまたま表記が似通っていた fax 情報と tel 情報が同じと見なされるなどの点も見られた．また，Web 上でリクエストするために実装されているコード類が存在し，これは"CT12"など文字数が短い上に提供元によって表記がまちまちなため正しい同定が行われなかった．また，XML の各要素の同定にかかった時間を図 3 に示す．同定結果の正誤についてはここでは問わない物とする．なお，実行環境は Pentium Dual-Core 1.8GHz，メモリ 1.5GB，WindowsXP SP2 である．

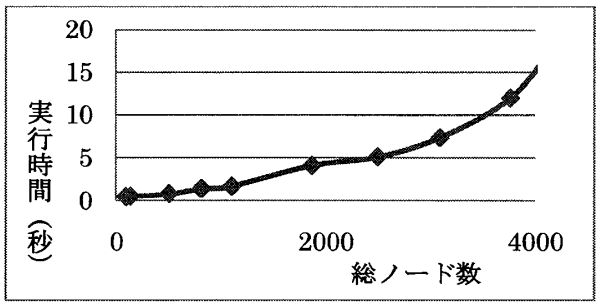

図 4　プログラム実行時間

実行時間のグラフはゆるやかな放物線を描いて上昇している．原因として要素中のテキストを全て解析しているため，要素が増えるほど時間がかかってしまうと考えられる．本評価では計測した最大要素数 4000 程度でファイルサイズ 250KB 程度であるが実際の XML のサイズがここまで肥大することはあまり多くないと想定され，多くの場合 100KB～150KB 程度と考えられるため，実行時間は 5 秒強程度だと思われる．

5. 終わりに

本研究では複数の XML 文書を統合し，一つの XML にまとめ上げるための手法及びそれを行うための要素の同定手法についての検討を行った．最も大きな課題として精度の向上が挙げられる．しかしテキストの表記揺れや特殊な要素の存在により全てを自動で判断するのは困難であるため，一部をユーザーに選択させて統合を行うような半自動的なシステムの構築や，WebAPI から得られるもの以外の様々な XML に対応した手法の検討も予定している．

参考文献

1) 横田治夫，梁文新：複数の XML 文書の類似度検出方法および類似性検出システム，ならびに複数の XML 文書の統合方法 J-STORE 公開特許(2006)
2) ぐるなび Web サービス http://api.gnavi.co.jp/api/service.htm
3) ホットペッパーWeb サービス http://api.hotpepper.jp/