ネットニュース用XMLタグセットの検討とその構造解析への応用

浅野 久子　　永田 昌明
NTT情報通信研究所

1．はじめに

近年、インターネットの急速な広がりとともに、WWWやネットニュース、電子メールなど、多くの電子化情報が流通するようになり、これらの膨大な情報を処理するために、情報抽出や要約、重要文抽出などの技術が注目を集めている。ここで、これらのインターネットを流通する情報のうち、WWWのHTML文書は、HTMLにより情報の構造化が行なわれている。しかし、ネットニュースや電子メールは、構造化が行われておらず、また、文字で描いた図や引用を表す記号等、通常のテキストとは異なる文字の用法が存在し、改行位置や句読点等の表現も多様である。このため、重要文抽出などの技術の基盤となる言語処理の基本単位（例えば「文」など）を把握するのも難しい。

我々はこれらのテキストに対する情報抽出（例えば住所録情報[1]やスケジュール情報[2]の抽出）や要約等を容易にするために、自動的にその構造を解析することを目標としている。本稿では、その第一ステップとして、データの収集が容易なネットニュースを対象に構造化用XMLタグセットを検討して、タグ付きコーパスを作成し、引用行の自動認定を行ったので報告する。

2．構造化の方針

2．1 構造化の目的とそのための対応

本稿で提案するネットニュースの構造化は、ネットニュースを対象とした重要文抽出や要約、および情報抽出を主なターゲットに想定し、これらで利用可能な自動構造解析ツールを提供することを目的としている。そこで、これらの利用ターゲットに対応するために、以下の3項目を考慮し、構造化を検討した。

(1) 処理区分領域の設定

ネットニュースは、ヘッダとボディから構成され、ボディには投稿者が書いた文章の他に、他のニュース記事の引用や、送信者の署名（signature）、文字で描いた図といった領域をもつ場合がある。ここで、前述の利用ターゲットにおいて、全ての領域を同様に扱うことは少ないと考えられる。例えば重要文の抽出では、最も重要な文は投稿者の記述した文章の中にあると考えられるので、そこに含まれる文章の重みを他のニュース記事から引用されている文章より大きくする、signatureは記事の内容は表さないので対象としない等、領域の種類により処理内容が変わるであろう。そこで、このように処理を区分すると考えられる単位で領域を設定する。

(2) 処理単位の設定

前述の利用ターゲットでは、通常、段落や文などの言語単位を処理単位とする。しかしネットニュースでは、改行位置や句読点等の表現が多様であるため、この処理単位を把握するのが困難である。そこで、これらの処理単位の設定を行う。

(3) 元テキストへの再現性

ネットニュースでは、文字による装飾や、空白文字の挿入によるセンタリング等のレイアウト的表現により、強調などが行われる場合がある[3]。そこで、レイアウト的な意味のみをもつ文字も削除せずに保存し、元テキストのレイアウトを復元可能とする。ただし言語解析においては、これらの文字は解析誤りなどの原因になる等の悪影響を及ぼすので、言語的に意味をもつ文字と区別できるようにする。

上記3項目を考慮した構造化において、ヘッダはそのフォーマットが規定されており、容易に構造化が可能である。しかし、ボディにはフォーマットの規定がなく、個々のニュース記事における表現が多様であるため、一部のレイアウト的表現以外は自動的に構造を解析することは難しく、その手法を検討する必要がある。

2．2 タグ付け言語

タグ付け用の言語としては、XML（eXtensible Markup Language）[4][5]を用いた。従来、テキストコーパスのタグ付け言語としては主にSGMLが用いられてきた。このSGMLとXMLは、独自タグを導入することによってマーク付け言語を作り出すための枠組みという点では同一であるが、XMLには以下の2つの利点があるため、XMLを用いることにした。

・ツールの充実性

SGMLのツール（パーザやブラウザなど）は数も少なく高価である。これに対し、XMLは発表されてから約1年半と間もないが、多くの会社がツールの開発を急いでおり、フリーのツールも出現している。また、WWWブラウザの一部にXMLが使われるようにもなってきている。これらのことから、XMLは多くの安価なツールが利用できるようになると考えられる。コーパスの作成や構造化されたタグ付けテキストを利用する上で、ツールの充実は重要である。

・インターネットへの適応性

XMLでは、ハイパーリンクにより、世界中のインターネット上の資源が表示・利用でき、インターネットに対する適応性が高い。これは、ネットニュース、電子メールといったインターネットを流

通するテキスト情報を扱う上で有用である。

3．文書型定義（タグセット）

XML文書は、1つ以上の要素からなり、要素の境界は開始タグと終了タグによって区切られ、要素はいくつかの属性をもつことができる。そしてXMLでは、文書型定義（DTD）によりタグセットを規定する。DTDは、主に次の2つの宣言により構成される。

・要素型宣言：要素の型を定義して、定義された要素型に属する要素がどんな内容をもつか（内容モデル）を記述する。

・属性リスト宣言：ある要素型に属する要素に対し、それがどんな属性を持ちうるかを定義する。

本稿では、1つのニュース記事が1つのファイルより構成されることを前提とし、2.1節で述べた3つの対応項目を考慮して、1ニュース記事を、ニュース記事を識別するための記事情報要素と、ネットニュースの構造に対応したヘッダ要素、ボディ要素から構成することにした。以下、この3要素およびその配下の要素の型と内容モデル、属性について説明する[1]。ここで、内容モデルの表記法を表1に示す。

表1　内容モデルの表記法

A,B	要素Aに続いて要素Bが出現
A|B	要素Aまたは要素Bが出現
()	まとめて扱う範囲を指定
*	0回以上の繰り返し
+	1回以上の繰り返し
?	0回または1回の繰り返し
#PCDATA	通常の文字列
空要素	内容をもたない要素

3．1 記事情報要素

記事情報要素は、ニュース記事の基本情報を表す。

・要素型：記事情報

・内容モデル：空要素

・属性：ニュースグループ、ファイル名

3．2 レイアウト保存用要素

表2に示す5種類の要素は、言語的には意味を持たず、レイアウト的な意味のみをもつ要素である。これは、2.1節（3）元テキストの再現性に対応するための要素であり、ヘッダ要素、ボディ要素の配下の多くの要素内に現れる。

表2 レイアウト保存用要素

要素型	内容モデル
改行	空要素
自動改行	空要素
インデント	#PCDATA
タブ	#PCDATA
飾り	#PCDATA

改行、自動改行要素は、ほぼすべての要素内に現れる。改行要素は、通常の改行位置、自動改行要素は、ニュースリーダで引用した場合などに自動的に挿入される改行位置（元記事では改行が入らない）を表す。例えば、

>ネットマスクが違ったコンピュータ同士は「マイクロソフト共有サー

>ビス」が使えないんでしょうか？

では、2行目と3行目の行末は通常の改行であるが、1行目の行末は引用時に自動的に挿入された自動改行である。

インデント要素は、主に文要素内に現れる。これは行頭部分の空白を表す。

タブ要素は、リスト、表要素における空白を表す。

飾り要素は、上記以外の言語的な意味を持たない文字を表し、段落要素内の多くの要素に現れる。

3．3 ヘッダ要素

ヘッダ要素は、ニュース記事のヘッダを表す。

・要素型：ヘッダ

・内容モデル：フィールド+, 改行

このフィールド要素は、ヘッダの各フィールドに対応する。

・要素型：フィールド

・内容モデル：フィールド名, フィールド値

・属性：ID（通し番号）

フィールド名、フィールド値要素は、フィールド名、フィールド値に対応し、内容モデルは、共に(#PCDATA|改行|自動改行)*である。

3．4 ボディ要素

ボディ要素は、ニュース記事のボディを表す。ボディは、RFC2045で規定されたMIMEのマルチパート構造の場合と、シングルパート構造または非MIME構造（RFC1036で規定）の場合がある。本稿では説明の簡略化のため、現在のところ、ニュース記事の大半を占めるシングルパート構造と非MIME構造のニュース記事のみを考慮したDTDについて説明を行う。MIMEの各パートを1つのニュース記事と見なすことにより、このDTDをマルチパート構造へ対応させることは、容易に可能である。

・要素型：ボディ

・内容モデル：領域種別[2]+

領域種別

領域種別は、2.1節で述べた（1）処理区分領域の設定に対応している。領域種別の各要素の内容モデル、属性を表3に示す。

引用要素は、他のニュース記事を引用した領域を表す。このため、ニュース記事のすべての要素が含まれる可能性をもつ。引用要素内の文字列からは、

[1]要素は実際には英語表記であるが、本稿では直感的に理解できるように、日本語表記に置き換えて記述した。

[2]領域種別は以下の内容モデルの別名である。
引用|添付書類|signature|図|自動挿入文章|通常文章

表3 領域種別の要素

要素型	内容モデル	属性		
引用	(ヘッダ¦領域種別)*	ID	引用先記事	引用記号
添付書類	外部アンパースト実体		元記事	エンコードタイプ、ファイルタイプ
signature	(#PCDATA¦改行¦自動改行)*			タイプ
図	(#PCDATA¦改行¦自動改行)*			
自動挿入文章	(文¦リスト¦改行¦自動改行)*			
通常文章	段落*			

引用記号（例：>>）を除き（元記事と同じ表記とする）、その引用記号を引用記号属性の値として保持する。文字列から引用記号を除くことにより、引用の領域を他の領域と同様に扱うことが可能となり、また、引用記号を属性値として保持することにより2.1節の（3）元テキストへの再現性にも対応する。

添付書類要素は、バイナリデータ等をuuencodeやBinHex等でエンコードした領域を表す。これらはデコードし、適切なアプリケーションで利用することで意味をもつ。そこで、この領域の文字列をすべて外部アンパースト実体、すなわちXMLで解析されない外部ファイルとする。また、属性としてエンコードタイプ、ファイルタイプをもつ。

signature要素は、ニュース投稿者の署名の領域を表す。タイプ属性は、signatureが定形（例えば署名ファイル等に保存してあるもの）であるか非定形であるかを表す。

図要素は、文字ベースで表される図の領域を表す。

自動挿入文章要素は、ニュースリーダで自動的に挿入する文章、例えば、「○○の記事において、××さんは書きました。」などの領域を表す。この領域は、重要文の抽出等では通常の文章より利用価値が低いと考えられる。

通常文章要素は、上記以外の、投稿者が書いた文章を表す。利用ターゲットにおいては、この領域が最も重要な領域であると考えられる。そこで、2.1節（2）処理単位の設定に対応し、次の段落要素の項で述べる詳細な構造化を行った。

領域種別の全要素がもつID属性は、領域種別共通の通し番号であり、引用により入れ子構造となった場合には、子番号を付与する。

また、引用要素の引用先記事属性は、その領域を最初に引用した記事を表す。例えば、記事Aを、最初に記事Bが引用し、さらに記事Bのその部分を記事Cが引用した場合には、引用先記事は記事Bとなる。その他の領域種別がもつ元記事属性は、その領域が記述されたオリジナルの記事を表す。つまり、引用以外の部分では、当該記事そのものをさす。これらの引用先記事、元記事属性の値は当該記事のMessage-IDで表す。

段落要素

段落は、文章のまとまった区切りを表す。本稿では、空行で区切られ話題が変わる部分、または、空行がないが完全に話題が変わる部分（挨拶と本論の境界など）を段落の境界として扱った。

・要素型：段落

・内容モデル：(リスト¦表¦コメント¦文¦飾り¦改行¦自動改行)+

・属性：ID（通し番号）

以下、段落を構成する要素のうち、言語的に意味をもつ要素の概要を説明する。

リスト要素は、個条書きなどの項目が列挙される構造を表し、主な内容として、ラベル要素と項目要素をもつ。例えば、

日時　4月1日
場所　東京ドーム

では、*日時*、*場所*がラベル要素、*4月1日*、*東京ドーム*が項目要素となる。

表要素は、表を表す。ネットニュースにおいては、表とリストのどちらとも考えられる表現があるが、その場合にはリストとして扱った。主な内容は、表ラベル要素と表エレメント要素である。例えば

No.	*製品名*	*価格*
1	*ABC*	*10,000*
2	*DEF*	*20,000*

では、*No.*、*製品名*、*価格*が表ラベル要素、*1*、*ABC*、*10,000*…が表エレメント要素となる。

コメント要素は、ネットニュースや電子メールで慣習的に行頭に「#」をつけ、本論から外れたことを示すコメントを表す。文要素は、文を表す。

4．コーパスの作成

現在、自動構造解析を行うための基礎データとして、3節で述べたDTDに基づくネットニュースコーパスを作成中である。対象としたニュースグループはfj.os.ms-window（Microsoft Windowsに関連する議論、情報）、fj.fleamarket.tickets（チケット関係等の個人的な売りと買い、または、交換しようとする物についての情報を提示する場所）、fj.life.health（健康に関する話題）の3種類であり、形式（討論型、アナウンス型）の異なる様々なニュース記事を網羅している。現在のところ、

fj.os.ms-windows：295記事
fj.fleamarket.ticket：290記事
fj.life.health：185記事

の計770記事分が作成されている。

5．コーパスを利用した引用領域判定

ネットニュースの自動構造解析の第一ステップとして、4節で述べたコーパスを対象に、C4.5[6]を用いて引用領域の判定を行った。ここで、3節で述べた引用時の自動改行に対応するため、通常の引用行、自動改行された引用行、非引用行の3クラスに各行を分類した。属性は表4に示す20の属性を用いた。

表4 引用領域判定のための属性

属性	値
行長（バイト数）	連続値
直前行長（バイト数）	連続値
引用記号候補種類	1, 2, 3, なし
引用記号候補長（バイト数）	連続値
引用記号候補第i文字種(i=1～9)	41種, なし
直前行の引用記号候補と一致?	yes, no
直後行の引用記号候補と一致?	yes, no
直前引用記号候補	1, 1', 2, その他
直後引用記号候補	1, 1', 2, その他
空白文字で区切られた文字列数	連続値
行末の文字種	41種, なし
ニュースリーダ	11種

表4において、引用記号候補は3種類存在し、種類1は、そのニュース記事内のある連続する2行で重複した行頭文字列と一致する行頭文字列、種類2は、種類1の末尾空白が欠落した文字列のみが存在する行全体、種類3は、種類1,2以外の、行頭に連続する記号、空白文字列（例：'>'）を表す。

文字種は、英字、数字、漢字、記号等で大文字と小文字を区別したものであり、記号については、細かく30種に分類している。

直前（直後）引用記号候補では、1は直前（直後）行全体が引用記号候補の種類1のみ、1'は種類1+任意の文字列、2は種類2の場合を表す。

ニュースリーダはマイクロソフト製ニュースリーダ、mnews、NewsWatcher等である。

4節で述べたネットニュースコーパスの561記事を学習用、209記事を評価用とし、ボディの各行を表4の属性により分類した。表5に評価データと誤り率を示す。ここで、

誤り率＝正しく分類された行数／全行数

である。

表5 評価データと誤り率

	C4.5		'>'=引用
	Close	Open	
行数	14678	5623	20301
誤り率	0.6%	1.2%	2.4%

本手法の精度を検証するため、引用記号として最もよく用いられる'>'が行頭から10バイト以内に存在した場合に、その行を引用行と判定した場合の誤り率も表5に示した。この結果、この'>'をキーとした手法と比較して、クローズデータで1/3、オープンデータで1/2に誤り率が減少し、本手法の有効性が確認された。

表6に、オープンデータにおける分類結果を示す。これにより、データ数の少ない自動改行引用行に対する分類精度が低いことがわかる。

表6 オープンデータにおける分類結果

正解＼推定値	非引用行	通常引用行	自動引用行
非引用行	4415	9	0
通常引用行	36	1137	3
自動改行引用行	4	16	3

6．おわりに

ネットニュース構造化のためのXMLタグセットを検討し、コーパスを作成して、引用領域の自動判定を行った。今後は、すべての処理区分領域の自動判定法を検討し、ネットニュースや電子メールを自動的に構造化するツールを開発する予定である。

参考文献

[1]浅野, 大山: 電子メールからのパーソナル情報抽出方法の検討, 第52回情処全大4J-4, 1996

[2]長谷川, 高木: 電子メールコミュニケーションにおけるスケジュール情報抽出, NL123-10, 1998

[3]佐藤, 佐藤, 篠田: 電子ニュースのダイジェスト自動生成, 情処論, Vol. 36, No. 10, 1995

[4]Extensible Markup Language (XML) 1.0 : http://www.w3.org/TR/REC-xml, 1998

[5]村田: XML入門, 日本経済新聞社, 1998

[6]Quinlan: C4.5 Programs for machine learning, Morgan Kaufmann Publishers, 1993

付録　タグ付けテキスト例

```
<?xml version='1.0' encoding='EUC-JP' ?>
<!DOCTYPE ニュース記事 SYSTEM "newsarticle.dtd" [
<!ENTITY a1 "12345@sample.mail.jp">
<!ENTITY a2 "67890@sample2.mail.jp">
]>
<ニュース記事>
<記事情報 ニュースグループ="fj.os.ms-windows" ファイル="0901.xml" />
<ヘッダ>
中略
<フィールド ID="8"><フィールド名>Message-ID:</フィールド名>
<フィールド値> &lt;12345@sample.mail.jp><改行/></フィールド値></フィールド>
<フィールド ID="9"><フィールド名>References:</フィールド名>
<フィールド値> &lt;067890@sample2.mail.jp><改行/></フィールド値>
</フィールド><改行/></ヘッダ>
<ボディ><通常文章 ID="1" 元記事="&a1;"><段落 ID="p1"><文 ID="s1">
<インデント>　</インデント>山田と申します。
</文><改行/><改行/></段落></通常文章>
<自動挿入文章 ID="2" article="&a1;">
<文 ID="s2">HanakoSuzuki wrote:</文><改行/></自動挿入文章>
<引用 ID="3" 引用先記事="&a1;" 引用記号=">">
<通常文章 ID="3-1" 元記事="&a2;"><段落 ID="p2"><改行/><文 ID="s3">
ブラウザ（ＮＮやＩＥ）の起動時にパスワードを聞いてこない方法って<改行/>
ありますか?</文><改行/></段落></通常文章></引用>
<通常文章 ID="4" 元記事="&a1;"><段落 ID="p3"><改行/>
<文 ID="4"><インデント>　</インデント>このパスワードって、メールパスワードのことでしょうか?</文><改行/><改行/></段落>
<段落 ID="p4"><文 ID="5"><インデント>　</インデント>ＮＮだったら、メール&ニュースの設定でメールパスワードの保存にチェッ<改行/>
<インデント>　</インデント>クを入れれば、聞いてこなくなりますけど…。
</文><改行/><改行/></段落></通常文章>
<signature ID="5" 元記事="&a1;">
--<改行/>
山田　太郎　(Taro Yamada)<改行/>
E-mail : yamada@sample.mail.jp<改行/>
HP URL : http://www.sample.mail.jp/~yamada/<改行/>
</signature></ボディ></ニュース記事>
```