自然言語処理基盤としてのウェブ文書標準フォーマットの提案

新里圭司[†]　橋本力[‡]　河原大輔[‡‡]　黒橋禎夫[†]
[†]京都大学 大学院 情報学研究科　[‡]山形大学 工学部　[‡‡]情報通信研究機構
{shinzato, kuro@nlp.kuee.kyoto-u.ac.jp}　ch@yz.yamagata-u.ac.jp　dk@nict.go.jp

1　はじめに

近年，HTML 文書やブログなどの World Wide Web (WWW) 上のテキストデータ (以下，ウェブ文書と呼ぶ) が多くの自然言語処理タスクで利用されるようになってきた．これに伴い，様々な研究機関でウェブ文書の収集・解析が行われている．しかし，WWW 上には膨大な量のウェブ文書が存在するため，1 つの機関単独で収集・解析することにはおのずと限界があるように思われる．また，現状では，文書の収集・解析は各機関で独立に行われており，その管理方法も異なるため，他の機関で収集・解析されたウェブ文書をシームレスに利用することは難しく，ウェブ文書の再利用性はかなり低いと考えられる．仮に，ウェブ文書が共通のフォーマットに従って管理されていれば，異なる機関のものであってもシームレスに利用することが可能になり，1 つの機関単独では入手できない量のウェブ文書の利用が期待できる．

そこで本稿では，再利用性の向上をはかるため，ウェブ文書を共有するためのフォーマットとして**ウェブ標準フォーマット**を提案する．そして，ウェブ文書から標準フォーマット形式のデータを生成するツールおよび，我々が生成した大規模な標準フォーマットデータについて述べる．

2　ウェブ標準フォーマット

ウェブ標準フォーマットとは，図 1 に示した DTD で定義されるフォーマットであり，表 1 に示す情報を含んでいる．本稿では，ウェブ標準フォーマットに従ってアノテーションされた XML 文書を**標準フォーマットデータ**と呼ぶ．図 2 にオリジナルの HTML 文書を，図 3 に図 2 の文書から生成される標準フォーマットデータをそれぞれ示す．

標準フォーマットは大きく分けると，<Header>タグで囲まれるヘッダー部と<Text>タグで囲まれるボディ部から構成される．ヘッダー部は，対応する HTML 文書のタイトル (<Title>)，HTML 文書へのインリンク情報 (<InLinks>，<InLink>) および，HTML 文書からのアウトリンク情報 (<OutLinks>，<OutLink>) などのメタ情報を含んでいる．

```
<!ELEMENT StandardFormat (Header,Text+)>
<!ATTLIST StandardFormat
    OriginalEncoding CDATA #REQUIRED
    Time CDATA #REQUIRED
    Url CDATA #REQUIRED>

<!ELEMENT Header (Title?,InLinks?,OutLinks?)>
<!ELEMENT Title (RawString,Annotation?)>
<!ELEMENT InLinks (InLink+)>
<!ELEMENT InLink (RawString,Annotation?,DocID+)>
<!ELEMENT OutLinks (OutLink+)>
<!ELEMENT OutLink (RawString,Annotation?,DocID+)>
<!ELEMENT DocID (#PCDATA)>

<!ELEMENT Text (S+)>
<!ATTLIST Text
    Author CDATA #IMPLIED
    Date CDATA #IMPLIED
    Title CDATA #IMPLIED
    Type (default|blog|comment) "default">
<!ELEMENT S (RawString,Annotation?)>
<!ATTLIST S
    Id CDATA #REQUIRED
    Length CDATA #REQUIRED
    Offset CDATA #REQUIRED>
<!ELEMENT RawString (#PCDATA)>
<!ELEMENT Annotation (#PCDATA)>
<!ATTLIST Annotation
    Scheme CDATA #REQUIRED>
```

図 1 標準フォーマットの DTD

その一方で，ボディ部は，対応する HTML 文書から抽出された日本語文および，日本語文を自然言語処理ツールを使って解析した結果を含んでいる．より具体的には，<S>タグで囲まれた範囲が，オリジナルの HTML 文書から抽出された日本語文 1 文に対応しており，その子要素である<RawString>タグにより実際に抽出された日本語文が，<Annotation>タグにより抽出された日本語文を自然言語処理ツールで解析した結果がそれぞれ囲まれている．また，同一の日本語文を複数の自然言語処理ツールで解析する場合を考慮し

表 1 標準フォーマットに明示されている情報

種類	情報	XML タグ／属性	備考
メタ情報	HTML 文書のタイトル	Title	構造を持つため属性ではなくタグで管理.
	HTML 文書へのインリンク	InLinks	構造を持つため属性ではなくタグで管理.
	HTML 文書からのアウトリンク	OutLinks	構造を持つため属性ではなくタグで管理.
	URL	Url	
	エンコーディング	OriginalEncoding	
	ページ取得日時	Time	「yyyy-mm-dd hh:mm:ss」形式.
本文情報	著者	Author	任意.
	更新日	Date	任意.
	タイトル	Title	任意.
	種類	Type	通常の Web ページかブログ，またはブログのコメント.
文情報	文 ID	Id	
	バイト長	Length	
	開始位置	Offset	ファイル先頭からのバイトオフセット.
	文字列	RawString	要素が日本語文であることを意味するタグ.
	NLP ツールの解析結果	Annotation	要素が解析結果であることを意味するタグ.
	NLP ツールの名称	Scheme	Juman，Knp など.

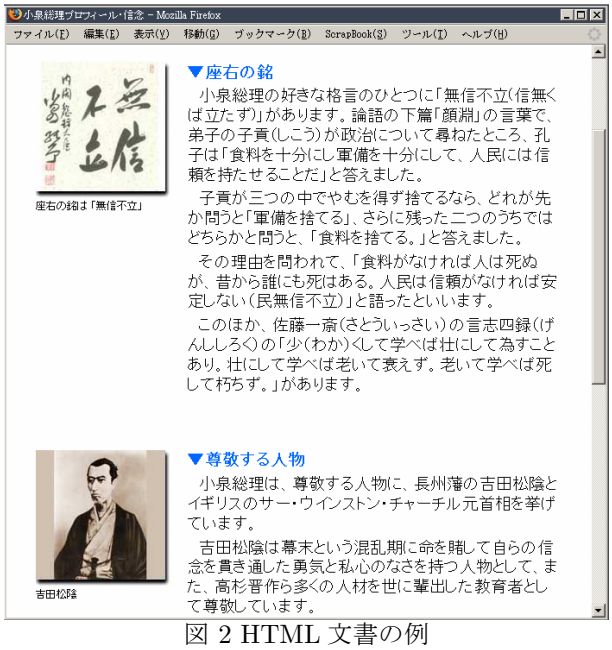

図 2 HTML 文書の例

て，<Annotation>タグの Scheme 属性の値を異なる値 (基本的にはツール名) にするだけで，複数の解析結果を同一の標準フォーマットデータに埋め込めるようになっている．

3 標準フォーマットデータの生成方法

以下に，HTML 文書から標準フォーマットデータを生成するための手順を示す．

Step 1: HTML 文書からの日本語文抽出
Step 2: 抽出された日本語文の解析
Step 3: 日本語文およびその解析結果に基づく標準フォーマットデータの生成

標準フォーマットデータの生成において最大の問題となるのは，HTML 文書からの日本語文の抽出方法である．以下では，日本語文抽出処理について述べる．

3.1 日本語文抽出処理

以下に，HTML 文書から日本語文を抽出するための手順を示す．

1. **文字コードによる日本語ページの判定:** HTML 文書中の charset 属性，または perl の Encode::guess_encoding() 関数を用いて HTML 文書の文字コードを調べる．文字コードが euc-jp，x-euc-jp，iso-2022-jp，shift jis，windows-932，x-sjis，shiftjp，utf-8 であれば，その文書を日本語ページと見なす．

2. **言語情報を用いた日本語ページ判定:** 1 では，より多くの日本語ページを取得するために，utf8 コードでエンコードされているページも日本語ページとして見なしている．しかし，utf8 コードは日本語以外の言語でも用いられるため，日本語以外のページが含まれている可能性がある．ここでは言語情報を用いて utf8 でエンコードされたページのうち，日本語以外のページを排除する．具体的には，日本語の助詞（「が」「を」「に」等）に注目し，助詞が一定以上の割合で含まれていないページは日本語ページでないと判定する．

3. **HTML タグと句点を用いた文抽出:** HTML タグおよび句点を手がかりに日本語ページと判定されたページから文を抽出する．文認識のための手かがりとして用いる HTML タグとしては，例えば，
や<p>を利用する．また<pre>タグ中の改行は，そのまま改行として扱う．

4. **日本語文の判定:** 抽出文の中から日本語の文だけを抽出する．これは，日本語ページと判定されて

```
<?xml version="1.0" encoding="UTF-8"?>
<StandardFormat
Url="http://www.kantei.go.jp/jp/koizumiprofile/1_sinnen.html"
OriginalEncoding="Shift_JIS" Time="2006-08-14 19:48:51">
<Text Type="default">
<S Id="1" Length="70" Offset="525">
  <RawString>小泉総理の好きな格言のひとつに「無信不立（信無くば
立たず）」があります。</RawString>
  <Annotation Scheme="KNP">
    <![CDATA[* 1D <文頭><サ変><人名><助詞><連体修飾><体言><係:
ノ格><区切:0-4><RID:1056>
小泉 こいずみ 小泉 名詞 6 人名 5 * 0 * 0 NIL <文頭><漢字><かな
漢字><名詞相当語><自立><タグ単位始><文節始><固有キー>
 ... 中略...
ます ます ます 接尾辞 14 動詞性接尾辞 7 動詞性接尾辞ます型 31 基
本形 2 NIL <表現文末><かな漢字><ひらがな><活用語><付属><非独立
無意味接尾辞>
。 。 。 特殊 1 句点 1 * 0 * 0 NIL <文末><英記号><記号><付属>
    EOS]]>
  </Annotation>
</S>
<S Id="2" Length="160" Offset="595">
  <RawString>論語の下篇「顔淵」の言葉で、弟子の子貢（しこう）が
政治について尋ねたところ、孔子は「食料を十分にし軍備を十分にして、
人民には信頼を持たせることだ」と答えました。</RawString>
  <Annotation Scheme="KNP">
    <![CDATA[* 1D <文頭><助詞><連体修飾><体言><係:ノ格><区
切:0-4><RID:1056>
論 ろん 論 名詞 6 普通名詞 1 * 0 * 0 "漢字読み:音 代表表記:
論" <漢字読み:音><代表表記:論><文頭><漢字><かな漢字><名詞相当
語><自立><タグ単位始><文節始>
 ... 中略...
ました ました ます 接尾辞 14 動詞性接尾辞 7 動詞性接尾辞ます
型 31 タ形 5 NIL <表現文末><かな漢字><ひらがな><活用語><付属><
非独立無意味接尾辞>
。 。 。 特殊 1 句点 1 * 0 * 0 NIL <文末><英記号><記号><付属>
EOS]]>
  </Annotation>
</S>
... 中略...
</Text>
</StandardFormat>
```

図 3 図 2 の HTML 文書から生成される標準フォーマットデータの例 (KNP による解析結果有)

いても，文単位で見ると日本語以外の文字列 (例えば，記号英数字など）から構成されている場合があるためである．そこで，ひらがな，カタカナ，漢字のいずれかが 60%以上含まれる文のみを日本語文として抽出する．

4 標準フォーマット生成ツール

本節では，HTML 文書から標準フォーマットを生成するツールである，html2sf および www2sf について述べる[1]．

単一の HTML 文書から標準フォーマットデータを生成する場合は html2sf を用いる．html2sf は，引数として与えられた HTML 文書から，3.1 節で述べた手順により日本語文を抽出し，抽出した日本語文の形態素解析結果もしくは構文解析結果を基に標準フォーマットを生成する．現在は，形態素解析ツールとして JUMAN[2]を，構文解析ツールとして KNP[3]をそれぞれサポートしている．

一方で，大量の HTML 文書からまとめて標準フォーマットを生成したい場合には www2sf を用いる．これは，あるディレクトリ以下にある HTML 文書に対して，順次 html2sf を適用するラッパープログラムである．そのため，html2sf 同様，JUMAN と KNP の解析結果を埋め込むことが可能である．

以下に，html2sf および www2sf の使い方を示す．

html2sf の使い方

書式 `html2sf [-j|-k] htmlfile`

-j: JUMAN の解析結果を埋め込む
-k: KNP の解析結果を埋め込む

例 sample.html から JUMAN の解析結果が埋め込まれた標準フォーマットデータを生成

```
html2sf -j sample.html > sample.sf
```

www2sf の使い方

書式 `www2sf [-j|-k] dir1 dir2`

-j: JUMAN の解析結果を埋め込む
-k: KNP の解析結果を埋め込む
`dir1`: HTML 文書が置いてあるディレクトリ
`dir2`: 標準フォーマットデータを書き出すディレクトリ

例 ディレクトリ `hdir` 以下にある HTML 文書から KNP の解析結果が埋め込まれた標準フォーマットを生成し，ディレクトリ `xdir` に保存

```
www2sf -k hdir xdir
```

5 大規模標準フォーマットデータの生成

NTCIR5-WEB タスクで利用された HTML 文書約 9,600 万ページ [1, 2] に対して，4 節で述べたツールを適用し，標準フォーマットデータ (KNP の解析結果付き) の生成を行った．具体的には，Intel CPU Xeon 3.8GHz × 2，メモリ 2GB のスペックを持つ計算機 48 台を用い，96 並列で生成処理を行った．その結果約 5,500 万件の標準フォーマットデータが得られた[4]．これはつまり，残りの約 4,100 万ページについては，日本語ページとして見なされなかったことを意味する．「日本語ページでない」と判定されページから，ランダムに 30 件選び出し人手で調査した．その結果，6 件 (20 %) については日本語を含むページであることが確認された．今後は，日本語ページ判定処理を調整す

[1]両プログラムは http://tsubaki.ixnlp.nii.ac.jp/tools/よりダウンロード可能.
[2]http://nlp.kuee.kyoto-u.ac.jp/nl-resource/juman.html
[3]http://nlp.kuee.kyoto-u.ac.jp/nl-resource/knp.html
[4]標準フォーマットデータの生成には約 2 週間かかった.

表 2 オリジナルの HTML 文書と標準フォーマットデータのファイルサイズ (約 5,500 万文書，gzip 圧縮時)

ファイルの種別	サイズ [TB]
オリジナルの HTML 文書	0.01
標準フォーマットデータ	1.4

表 3 TSUBAKI API で指定可能なリクエストパラメータ (一部)

パラメータ	値	説明
query	string	検索クエリ (utf8) を URL エンコードした文字列．検索結果を得る場合は必須．
start	integer	取得したい検索結果の先頭位置
results	integer	取得したい検索結果の数
id	string	個別の文書を取得する際の文書 ID．オリジナルのウェブ文書，または標準フォーマット形式の文書を得る際は必須．
format	html/xml	オリジナルのウェブ文書，または標準フォーマット形式のウェブ文書のどちらを取得するかを指定．id を指定した際は必須．

ることで，これらのページに対応したい考えている．表 5 に標準フォーマットデータが生成された約 5,500 万ページ分のオリジナル HTML 文書と標準フォーマットデータのファイルサイズを示す．

生成された標準フォーマットデータは開放型検索エンジン基盤 TSUBAKI の API [5]を利用することで入手可能である．TSUBAKI API では表 3 に示したリクエストパラメータを提供している．以下に，API にアクセスするためのリクエスト URL の例を示す[6]．

例 1: 「京都」について検索した結果の上位 20 件を取得したい場合

http://tsubaki.ixnlp.nii.ac.jp/api.cgi?query=%E4%BA%AC%E9%83%BD&starts=1&results=20

例 1 の場合について，TSUBAKI API より返される検索結果を図 4 に示す．検索結果には，検索語，ヒット件数，検索した日時などのメタ情報に加え，検索語を含む HTML 文書の ID，タイトル，オリジナル HTML 文書の URL などの情報が含まれている．標準フォーマットデータを取得したい場合は，図 4 に示した検索結果から文書 ID を抽出し，再度 API に問い合わせればよい．以下に，文書 ID が 01234567 である標準フォーマットデータを取得する場合のリクエスト URL を示す．

例 2: 標準フォーマットデータ (ID=01234567) を取得したい場合

http://tsubaki.ixnlp.nii.ac.jp/api.cgi?id=01234567&format=xml

[5]http://tsubaki.ixnlp.nii.ac.jp/api.cgi

[6]紙面の都合上，“query=” の後に改行を挿入しているが，実際は不要なので注意．

```
<?xml version="1.0" encoding="utf-8"?>
<ResultSet time="2007-02-01 04:55:01" query="京都"
totalResultsAvailable="1390900" totalResultsReturned="20"
firstResultPosition="1" rankingMethod="OKAPI"
logicalCond="AND">
<Result Id="09405221" Score="10.46451">
  <Title>京都府ホテル</Title>
  <Url>http://okasoft.ddo.jp/pasokon/z_kyouto.html</Url>
  <Cache>
    <Url>http://tsubaki.ixnlp.nii.ac.jp/se/index.cgi?URL=
    INDEX_NTCIR2/09/h0940/09405221.html&KEYS=%B5%FE%C5%D4</Url>
    <Size>3316</Size>
  </Cache>
</Result>
<Result Id="37461679" Score="10.43856">
  <Title>ＪＴＢおすすめ・京都の宿！国内旅行／激安旅行／トリップサイト！</Title>
  <Url>http://www.tripsite.jp/jtb/kinki/kyoto1.html</Url>
  <Cache>
    <Url>http://tsubaki.ixnlp.nii.ac.jp/se/index.cgi?URL=
    INDEX_NTCIR2/37/h3746/37461679.html&KEYS=%B5%FE%C5%D4</Url>
    <Size>4503</Size>
 ... 中略...
  </Cache>
</Result>
</ResultSet>
```

図 4 TSUBAKI API より返される検索結果の例

6 おわりに

本稿では，自然言語処理のためのウェブページ用標準フォーマットを提案した．標準フォーマットには，HTML 文書から抽出された日本語文だけでなく，日本語文の解析結果についても埋め込み可能である．これにより解析済みデータの再利用性が高まり，結果として計算機資源の有効活用にもつながると考えられる．

今後は，さらに多くの HTML 文書から標準フォーマットデータを生成し，TSUBAKI API にて随時提供していく予定である．

参考文献

[1] Keizo Oyama, Masao Takaku, Haruko Ishikawa, Akiko Aizawa, and Hayato Yamana. Overview of the ntcir-5 web navigational retrieval subtask 2 (navi-2). In *Proceedings of the Fifth NTCIR Workshop Meeting on Evaluation of Information Access Technologies*, pp. 423–442, 2005.

[2] Masao Takaku, Keizo Oyama, Akiko Aizawa, Haruko Ishikawa, Kengo Minamide, Shin Kato, Hayato Yamana, and Junya Hayashi. Building a terabyte-scale web data collection ”nw1000g-04” in the ntcir-5 web task. In *NII Technical Report, No.NII-2006-012E*, 2006.