未知のサイトに含まれる Web ページからの主要部分抽出手法

鶴田 雅信, 増山 繁

豊橋技術科学大学 知識情報工学系

tsuruta@smlab.tutkie.tut.ac.jp, masuyama@tutkie.tut.ac.jp

1　はじめに

Web ページを有効利用する為に, 全文検索, 情報抽出などの自然言語処理技術は不可欠となっている．しかし, Web ページには, そのページにおいて伝えたい情報とは直接関係のない情報が多く含まれている．たとえば, 図 1 のような Web ページでは, ナビゲーション, 広告, リンク集などが存在する．人間が Web ページを閲覧する上ではこれらの情報は必要であるが, 機械的に文書を処理する用途においてはノイズとなることが多い．このような Web ページに含まれるノイズを削除する, もしくは主要な部分のみを抽出することで, 全文検索, 情報抽出の精度を向上させることができる．

図 1　Web ページに含まれるノイズの例

Web ページからの主要部分抽出というタスクには, 既に多くの手法が提案されている．Finn らの手法 [1] では, HTML 文書をテキストトークン（たとえば, 単語など）, および HTML タグの混在したトークン列とみなした上で,「本文部分のテキストの割合を最大化し, 本文以外の部分のタグの割合を最大化する」部分トークン列を主要部分として抽出するというアプローチを取っている．この手法には, 画像が主要部分の大半を占めるような Web ページには適用できないという問題点がある．

Debnath ら [2] は, 同じサイトに所属する Web ページを大量に収集し, ページ中の部分のサイト内における出現確率を用いて, 主要部分を抽出する手法を提案している．しかし, 未知の Web ページに対しては, 同じサイト内に含まれる Web ページを大量に取得する必要がある．そのような場合に, クロールを行うことがコストに見合わないアプリケーションが存在する．例えば, モバイル機器において転送量を削減させるためのプロキシサーバなどでは, 多くの「ただ 1 度だけ 1 ページしか見られることがないサイト」へのリクエストすら存在する．そのようなアプリケーションにおいて, 未知の Web ページに対して適用可能な手法が必要であると考えられる．

Cai ら [3] が提案した VIPS は, ブラウザによってレンダリングされた DOM [*1] ノードの背景色, フォントサイズ, 大きさなど, 様々なレイアウト情報を用いて文書のセグメンテーションを行う．Song らは, VIPS によってセグメンテーションされた結果に対して, セグメントごとの重要度を人手で付与し, SVM を用いた機械学習ベースの手法によって, セグメントの重要度を判定する手法を提案している [4]．この手法は, VIPS によるセグメンテーションの結果に依存する．

Gupta ら [5] は, ルールベースによって広告, リンクリストを除去することで, 手法を提案している．この手法も, Tsuruta らの手法と同様にクロール, セグメンテーションを必要としない．しかし, この手法で削除できるノイズのパターンは限られている．

Tsuruta ら [6] は, 平均的な Web ページにおいて, 主要な DOM ノードがウィンドウのどの位置を覆っているかという情報を用いて, それを抽出する手法を提案している．この手法は, サイト全体のクロールを必要とせず, セグメンテーションに依存しない．しかし, 1 つの DOM ノードのみを主要なものとして抽出することしかできないため, その DOM ノードの中にノイズが含まれる場合に, 削除する方法が存在しない．本論文では, Tsuruta らの手法によって抽出された主要な DOM ノードに対して, Gupta らの手法を適用することで, 主要部分を抽出する手法を提案する．

2　手法

2.1　主要 DOM ノードの抽出

Tsuruta ら [6] によって提案された主要な DOM ノードの抽出手法の概要について説明する．最初に, 人間によって作成された学習データ N_g [*2] を集約し,「一般的

[*1] http://www.w3.org/DOM/

[*2] 学習データのアノテーションは,「ある Web ページにおいて, 主要な部分を完全に含み, かつ分割不可能なものを除いてノイ

な Web ページにおいて主要な DOM ノードがウィンドウ内のどの位置を多く被覆するか」という知識の一表現として weight-map を生成する．weight-map の生成は以下の手順で行う．

Step 1　N_g に含まれる Web ページ d を一般的な Web ブラウザを用いてレンダリングし，その Web ページに含まれる全ての DOM ノード n から位置情報を抽出する．位置情報は，ブラウザとして Firefox [*3] を使用する場合であれば，nsIDOMNSHTMLElement [*4] などのインターフェイスから取得する．取得した位置情報が親ノードの左上端を始点とした相対座標系のものであれば，ウィンドウの左上端を原点とした絶対座標系のものに変換する．

Step 2　DOM ノードのオフセット情報，および学習データから，weight-map を生成する．weight-map は Web ブラウザのウィンドウを縦 $x\times$ 横 y 個のセルで分割した後，それぞれのセル c に対して，正解データにおいて主要な DOM ノードとされたノード A がそのセルを被覆する確率 $W(c)$ を割り当てたものである．

$$W(c) = \frac{\sum_{d \in N_g} E(A, c, d)}{|N_g|}, \qquad (1)$$

ここで，$E(A, c, d)$ は Web ページ d において，セル c が A に被覆されていた場合は α，A の兄弟ノード，および直系でない先祖ノードに被覆されていた場合は -1 ，それ以外の場合は 0 とする．

weight-map の例を図 2 に示す．中心近くの白い部分は主要部分，両端の黒い部分はノイズ部分を多く被覆していることを示している．

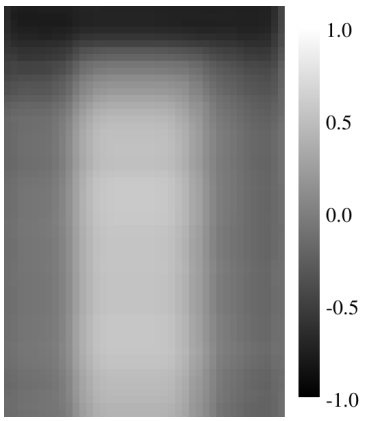

図 2　weight-map

次に，識別対象の Web ページ d_t に含まれるすべての DOM ノード e に対して，weight-map を用いて「主要な DOM ノードらしさ」のスコア $score(e)$ を算出する．手法は以下の通りである．

Step 1　d_t に含まれるブロックレベル要素 [*5] である DOM ノードのうち，縦幅，横幅ともに weight-map に含まれるセルのサイズよりも大きなものの集合を，主要 DOM ノード候補集合とする．

Step 2　主要 DOM ノード候補集合に含まれるすべての DOM ノード e に対して，$score(e)$ を求める．

$$score(e) = \sum_{c \in C(e)} W(c), \qquad (2)$$

ここで，$C(e)$ は weight-map において e によって被覆されているセルの集合とする．

Step 3　$score(e)$ が最大である候補を，その Web ページの主要 DOM ノードとして抽出する．

2.2　ヒューリスティクスによるノイズ削除

Tsuruta らの手法 [6] では，主要部分として 1 つの DOM ノードのみを抽出する．しかし，この手法では，主要部分として抽出された DOM ノードの子孫にノイズが存在した場合，これを削除することは不可能である．一方，Gupta ら [5] は，ヒューリスティクスを利用したノイズ削除手法を提案している．この手法を Tsuruta ら (2008) によって抽出された主要 DOM ノードに対して適用することで，内部に残されていたノイズを削除できると考えられる．Gupta らの手法を以下に示す．

Step 1　既存の広告リストを取得し，それにマッチする URL に対して参照している DOM ノードを削除する．

Step 2　DOM ノードに含まれているテキストのうち，リンクが張られているテキストの文字数の割合が一定以上の DOM ノードを削除する．Gupta らの論文において，その割合は 5 とされている．

Step 3　Step 1, 2 によって，内容が空となった DOM ノードを削除する．

3　実験

人手で作成された正解データを用い，それぞれの手法の組み合わせによる実行結果に対して被覆率，ノイズ被覆率を算出するという方法で，評価実験を行った．評価対象となるのは以下の手法となる．

BODY+Gupta　BODY タグに囲まれた部分に対して Gupta らの手法を適用する．

WM　Tsuruta らの手法 [6].

WM+Gupta　Tsuruta らの手法 [6] によって抽出された主要 DOM ノードに対して，Gupta らの手法を適用したもの．

正解データとして，「はてなブックマーク」のタグ（music, lifehacks, business, 政治, あとで読む）による

ズを含まない，ただ 1 つの DOM ノードのみを正解とする」という教示のもと行った．

[*3] http://www.mozilla.org/firefox/

[*4] http://xulplanet.mozdev.org/references/xpcomref/nsIDOMNSHTMLElement.html

[*5] http://www.w3.org/TR/html4/struct/global.html#h-7.5.3

検索結果の中から, 重複するサイトのものを削除した 191 ページを使用した. weight-map の学習データとして, 「はてなブックマーク」のタグ（魚, 動物, レシピ）による検索結果の中から, 重複するサイトのものを削除した 62 ページを使用した. また, 学習データに出現する Web サイトは正解データから削除し, さらに, 主要部分が明確に存在する Web ページのみを正解データ, 学習データとして使用した.

正解データ作成時, 以下のようなものしか含まない DOM ノードをノイズ部分としてアノテートした.

- サイト内において頻出するヘッダ, フッタ領域
- テンプレートによって生成された広告, ナビゲーション
- 類似記事リスト
- コメント投稿フォーム

3.1 評価尺度

本実験では, 正解データ中の全文書に対してそれぞれ被覆率, ノイズ被覆率を求め, それらの平均をベースに手法間の比較を行う. ここで, 被覆率とは, 正解データにおいて主要部分とアノテートされている DOM ノードが, 実験対象となる手法によっても主要部分として抽出されている割合である. 被覆率が高ければ, 誤って主要部分を削除することが少ないと示される. また, ノイズ被覆率とは, 正解データにおいてノイズ部分とアノテートされている DOM ノードが, 手法によってもノイズ部分として抽出されている割合である. ノイズ被覆率が高ければ, ノイズを多く削除していることが示される.

ある Web ページ d において, 被覆率 Cov , ノイズ被覆率 Cov_n は以下の式で求める.

$$Cov(d) = \frac{I(e_p(d))}{I(e_p(d) \cup e_{fn}(d))}, \tag{3}$$

$$Cov_n(d) = \frac{I(e_n(d))}{I(e_n(d) \cup e_{fp}(d))}, \tag{4}$$

ここで, $I(\varepsilon)$ を文字数および画像の総数に基づく DOM ノードの重みとし, 以下の式で表す.

$$I(\varepsilon) = Char(\varepsilon) + 10 \cdot Obj(\varepsilon), \tag{5}$$

ε: 任意の DOM ノードの集合,
$Char(\varepsilon)$: ε に含まれる DOM ノードが子として持つテキストノードに含まれる文字数の和,
$Obj(\varepsilon)$: ε に含まれる DOM ノードが子として持つ DOM ノードのうち, tagName が IMG, ENBED または OBJ であるもの (画像など) の総数,
$e_p(d)$: Web ページ d の正解データにおいて主要部分としてアノテートされており, かつ手法によって主要部分として抽出された DOM ノードの集合,
$e_n(d)$: d においてノイズとしてアノテートされており, かつ手法によってノイズとして抽出された DOM ノードの集合,
$e_{fp}(d)$: d においてノイズとしてアノテートされていたが, 手法によって主要部分として抽出された DOM ノードの集合,
$e_{fn}(d)$: d において主要部分としてアノテートされていたが, 手法によってノイズとして抽出された DOM ノードの集合.

3.2 パラメータ

Tsuruta らの手法における, Web ブラウザのウィンドウサイズは縦 800× 横 960 ピクセルとした. また, weight-map のセルのサイズは 20×20 ピクセルとし, $\alpha = 0.6$ とした [*6] .

本実験実施時には, Gupta らの手法において使用されている広告リストが入手不可能であったため, 代替として mozilla.org によって提供されているもの [*7] を使用する. この広告リストは CSS の形式で記述されており, Web ブラウザのユーザスタイルシートとして使用する目的で提供されている. 本実験では, これを使用するために, Gupta らの手法の Step 1 を以下のように変更した.

Step 1' *display: none* のスタイルが指定される DOM ノードを削除する

この広告リストの一部を図 3 に示す.

```
iframe[src*="googlesyndication"],
*[src*="doubleclick"],
*[href*="http://216.92.21.16/"] {
        display: none !important;
}
```

図 3 広告リスト（一部）

実験結果を表 1 に示す.

表 1 実験結果

	平均 Cov	平均 Cov_n
BODY+Gupta	0.968	0.484
WM	0.964	0.801
WM+Gupta	0.934	0.872

4 考察

実験結果より, WM+Gupta は, 主要部分をよく抽出し, かつ WM 単体よりもノイズを多く削除することに成功している. 一方, Gupta らの手法を単体で使用した場合, WM を単体で使用したものよりもノイズ被覆率が低くなっている. これは, 図 4 に示されるように, ヘッダ, フッタなどの, Gupta らの手法によっては削除できないノイズが存在していることを示している.

Gupta らの手法を WM と併用することは有効であったが, 依然として削除できないノイズも存在していた.

[*6] 学習データのみを用いるクローズドテストにより調整した.
[*7] http://www.mozilla-japan.org/support/firefox/adblock

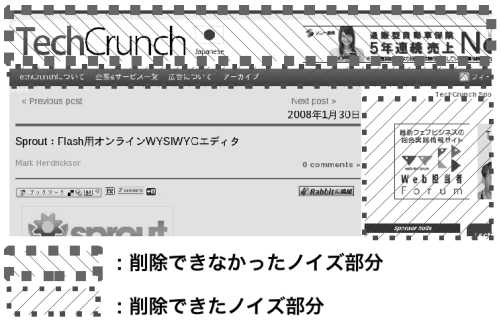

図 4　Gupta らの手法で削除できなかったヘッダ

```
<dl>
 <dt>
  <a><h4>■サーチマーケティングの新情勢</h4></a>
  <a>
    <img />「ナゾトキ」に学ぶ検索窓付き
    広告の成功法（後半）
  </a>
 </dt>
 <dd>
  前回からニンテンドー DS 用ソフト「レイトン教授と不思議な町」
  を事例に検索窓付き広告の成功法について解説してきた．後半
  では，検索エンジン対策（SEM）とウェブサイトのコンテンツ
  という 2 点について説明していく．
 </dd>
 （以下 dt，dd の繰り返し）
</dl>
```

図 5　Gupta らの手法で削除できなかったナビゲーションの例

たとえば図 5 のように, リンクリストであるが, アンカーテキストではない説明文を多く含む部分は削除できなかった．また, 日本語で書かれた Web サイトにおいてよく使用されている広告配信サービスの URL が広告リストに登録されておらず, 削除できなかった事例も存在した.

WM がフッタ, ヘッダの領域に離れて出現する主要部分を, ノイズであると判定してしまう失敗例が存在した．図 6 の例では, タイトル部分をノイズとして誤って削除している．この例の場合, タイトル部分も含む（被覆率 1 となる）DOM ノードはサイドバーの領域も含むため, スコアが低くなり, WM では抽出されなかった. これを解決するには, WM を複数の DOM ノードを抽出できるように拡張するなどの方法が考えられる.

5　まとめ

Web ページのノイズを削除し主要部分を抽出するために, ブラウザによるレンダリング結果を利用する手法, およびヒューリスティクスに基づくノイズ削除手法を組み合わせることで, 高い被覆率, ノイズ被覆率を達成した．しかし, 依然として削除できないノイズも存在しており, これらを高精度で削除する手法の考案が必要である.

図 6　タイトルをノイズと誤判定した例

謝辞

本研究は文部科学省グローバル COE プログラム「インテリジェントセンシングのフロンティア」及び, 文部科学省科学研究費特定領域研究 (B) (2) 16092213 の支援を受けた.

参考文献

[1] A. Finn, N. Kushmerick, and B. Smyth, “Fact or fiction: Content classification for digital libraries,” DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries, 2001.

[2] S. Debnath, P. Mitra, N. Pal, and C. Giles, “Automatic identification of informative sections of web pages,” IEEE Transactions on Knowledge and Data Engineering, vol.17, no.9, pp.1233–1246, 2005.

[3] D. Cai, S. Yu, J.R. Wen, and W.Y. Ma, “Vips: a vision-based page segmentation algorithm,” Microsoft Technical Report MSR-TR-2003-79, 2003.

[4] R. Song, H. Liu, J.R. Wen, and W.Y. Ma, “Learning block importance models for web pages,” WWW2004, pp.203–211, 2004.

[5] S. Gupta, G. Kaiser, D. Neistadt, and P. Grimm, “Dom based content extraction of html documents,” WWW2003, pp.207–214, 2003.

[6] M. Tsuruta, H. Sakai, and S. Masuyama, “An informative dom subtree identification method from web pages in unfamiliar web sites,” to appear in IEICE Trans. Information and Systems, vol.ED, 2008.