論　文

Webページ分割のための決定木学習を用いたタイトルブロック抽出

佐野　博之† 白松　俊† 大囿　忠親† 新谷　虎松†

Extracting Title Blocks for Web Page Segmentation Algorithm Using Decision Tree Learning

Hiroyuki SANO†, Shun SHIRAMATSU†, Tadachika OZONO†, and Toramatsu SHINTANI†

あらまし　本研究で提案するWebページ分割手法では，Webページを細分化ブロックという単位まで分割した後に，Webコンテンツの見出しとなるようなブロック（タイトルブロック）に着目して細分化ブロックの結合を行うことにより，Webページを意味的にまとまりのある単位へと分割する．既存のWebページ分割手法の多くが，面積や子ノード数など，コンテンツ量に依存する情報を用いて結合を行っていた．その結果，同一Webサイト内の同じレイアウトのWebページから異なる分割結果が得られるという問題が存在した．提案手法ではコンテンツ量に非依存な結合を行うために，タイトルブロックとそれに続くタイトルブロック以外のブロック（一般ブロック）を結合していく．そのためには，計算機によるタイトルブロックの抽出が課題となる．計算機によるタイトルブロックの自動抽出を行うために，機械学習によって分類器を生成した．J4.8アルゴリズムによる決定木学習によって生成した分類器により，F値77.8%，89.3%でタイトルブロックと一般ブロックの抽出に成功した．得られたタイトルブロックを用いて細分化ブロックの結合を行った結果，ニュースサイトのニュース記事部分に着目した場合，96.1%の精度でコンテンツ量に依存しない同一の分割結果が得られることを確認した．

キーワード　Webページ分割，Webページレイアウト，Webマイニング，決定木学習

1. まえがき

本研究はWebページ分割に関する研究である．Webページ中に存在する，閲覧者にとって意味的にまとまりのある単位のことを，本研究ではWebブロックと呼ぶ．Webページ分割とは，計算機によってWebページをWebブロック単位へと分割することである．本研究で提案するWebページ分割手法では，Webページを細分化ブロックという単位まで分割した後に，タイトルブロックに着目して細分化ブロックの結合を行うことにより，Webページを意味的にまとまりのある単位へと分割する．タイトルブロックとは，直下に存在するWebコンテンツの見出しとなるような細分化ブロックのことである．

Webページ分割アルゴリズムを確立することにより，様々なWeb技術の精度向上が期待できる．図1はYahoo!ニュースのスクリーンショットである．このWebページの中にはメインコンテンツ（実線）である記事のほかに，サイトロゴや広告，サイトメニュー，関連記事などのサブコンテンツ（破線）が含まれている．一つのWebページには多種多用な情報が記載されており，その中で閲覧者が必要としている情報はわずかである．Webページ検索システム，コンテンツフィルタリングシステム，情報抽出システム等でこのようなWebページを処理対象とする場合，メインコンテンツ以外のテキスト情報がノイズとなり，システムの精度が低下してしまう[1]．システムの精度を向上させるためには，処理対象のWebページをWebブロックへと分割し，Webページ中の主要なWebブロックのみをシステムの処理対象とすればよい．

以下に本論文の構成を述べる．**2.**ではWebページ分割に関する既存研究について言及する．**3.**でタイトルブロックに着目したWebページ分割手法の概要について述べる．**4.**では提案手法のアルゴリズムについて詳しく説明をする．**5.**では評価実験を行い，最後に**6.**で本論文をまとめる．

† 名古屋工業大学大学院工学研究科情報工学専攻，名古屋市
Dept. of Computer Science and Engineering, Graduate School of Engineering, Nagoya Institute of Technology, Gokiso-cho, Showa-ku, Nagoya-shi, 466–8555 Japan

図 1　Web ページ中に含まれる複数の Web コンテンツ
Fig. 1　Multiple Web contents in a Web page.

2. 関連研究

既存の研究で提案されている Web ページ分割手法には，大きく分けて，Web ページを記述している HTML の DOM 構造を用いた分割手法と，Web ページをレンダリングした結果のレイアウト情報を用いた分割手法の 2 種類がある．

2.1 DOM 構造を用いた分割手法

文献 [2], [3] では，DOM 構造を用いた Web ページ分割手法が提案されている．文献 [2] では，各 DOM ノード間の DOM 構造上の距離に着目した分割ルールにより，DOM トリーをブロックに分割している．文献 [3] では，特定の DOM ノードを根とした部分木を生成し，そこから葉ノードまでのパスのエントロピーを用いて，Web ページ中の意味のあるブロックを抽出している．

HTML4 の特徴として，文章の内容と表現の分離が挙げられる．DOM 構造をもつ HTML ファイルには文章の内容のみを記述し，Web ページの見た目（表現）はスタイルシートに記述される．したがって，Web ページを閲覧者の観点から分割するためには，HTML の DOM 構造を解析するだけでなく，HTML をスタイルシートとともにレンダリングして得られるレイアウト情報も用いる必要がある．

2.2 レイアウト情報を用いた分割手法

我々はサポートベクターマシンによる学習を行うことによってレイアウト情報に基づき Web ページを分割する手法を提案した [4], [5] が，この手法を適用して得られる分割結果の粒度は非常に粗い．Web ページ中からメインコンテンツのみを抽出するためには，上記手法を適用して得られる分割結果を更に細かく分割する必要がある．

文献 [6], [7] では，VIPS アルゴリズムと呼ばれるレイアウト情報を利用したヒューリスティクスに基づく Web ページ分割手法が提案されている．フォント情報，面積，背景色，座標など，レイアウトに関する様々なパラメータを用いた 12 個のルールを HTML タグごとに使い分けることで，Web ページをコンテンツ単位へと分割する．文献 [8] では，各 DOM ノードの座標情報をパラメータとして決定木を用いた機械学習を行うことによって，Web ページを九つのブロックに分割する手法が提案されている．

文献 [6]～[8] で提案されている手法では，Web ページを一度非常に細かいブロックまで分割した後，二つのブロックにおけるフォントや背景色の違い，Web ページのレンダリング結果におけるブロックの面積やブロック間の距離などを利用し，フォントが同じである場合や距離が小さい場合に二つのブロックを結合している．しかし，それらフォントの違いやブロック間の距離が Web コンテンツの切れ目を明確に表している Web ページは少ない．また，長い文章の段落ごとには一定間隔の距離を空けることも多いが，実際にはそれらの段落がまとまって一つの Web コンテンツを示している．ブロックの面積は，そのブロック内部に存在するテキストの量や画像の解像度などによって大きく変化する．そのため，同じ Web サイト内に存在する同一レイアウトの Web ページでさえも，メインコンテンツのテキスト量が変化した場合に，異なった Web ページ分割結果が作成されるという問題がある．

3. タイトルブロックに着目した Web ページ分割

既存の Web ページ分割手法では，Web ページを非常に細かい単位まで分割した後に視覚情報や DOM 構造を利用してそれらを結合し，意味的にまとまりのある単位である Web コンテンツへと分割している．本論文で提案する手法においても既存の手法と同様，一度細かい単位（細分化ブロックと呼ぶ）まで分割した後に，細分化ブロックを結合する点では同じである．本手法では結合の際にタイトルブロックに着目した結合を行っており，これが既存研究との差分である．タイトルブロックとは，細分化ブロックの中でも特に，直下の Web コンテンツの見出しとなる細分化ブロッ

クのことである．

タイトルブロックに着目した理由を二つ述べる．一つ目の理由として，Web コンテンツが多数配置されている Web ページには，人が閲覧したときに読解しやすいよう，Web コンテンツの上部にタイトルブロックが配置されていることが多いことが挙げられる．すなわち，タイトルブロックは複数の Web コンテンツ間の仕切りとして利用することが可能であるといえる．我々は予備実験を行い，大半の Web ページに含まれる Web コンテンツが，タイトルブロックとそれに続く本文・画像から構成されていることを確認した．本予備実験に関しては，**5.1** で詳しく述べる．

二つ目の理由として，細分化ブロックのコンテンツ量に非依存な結合が可能になることが挙げられる．既存手法では，二つのブロックにおけるフォントや背景色の違い，また Web ページのレンダリング結果におけるブロックの面積やブロック間の距離などを利用し，フォントが同じである場合や距離が小さい場合に二つのブロックを結合している．しかし，それらフォントの違いやブロック間の距離が Web コンテンツの切れ目を明確に表している Web ページは少ない．例えば，図 2 に示した (b) のような細分化ブロックを結合して (a) のようなコンテンツブロックを作成する場合，細分化ブロック間でフォントサイズや背景色が異なるため，既存手法では結合を行うことができない．また，長い文章の段落ごとには一定間隔の距離を空けることも多いが，実際にはそれらの段落がまとまって一つの Web コンテンツを示している．ブロックの面積は，そのブロック内部に存在するテキストの量や画像の解像度などによって大きく変化する．そのため，同じ Web サイト内に存在する同一レイアウトの Web ページでさえも，メインコンテンツのテキスト量が変化した場合に，異なった Web ページ分割結果が作成されるという問題がある．タイトルブロックを使った結合を行うことにより，このような問題を解決することが可能となる．**5.4** で実験を行い，高い精度でメインコンテンツ量に依存しない Web ページ分割が可能になったことを示す．

図 2　ブロックの粒度
Fig. 2　Block size.

4. Web ページ分割の流れ

4.1 細分化ブロックへの分割

細分化ブロックへの分割には，W3C が定義するブロックレベル要素を用いる(注1)．本論文では細分化ブロックを，“子ノードとしてブロックレベル要素をもたないブロックレベル要素” と定義する．ただしインライン要素であっても，細分化ブロックの兄弟ノードである場合には，そのインライン要素も一つの細分化ブロックとして抽出する．これにより，Web ページ上にレンダリングされる全ての要素がいずれかの細分化ブロックに属することとなる．DOM ノード n_i がブロックレベル要素であるか否かは，表 1 の四つのルールを *Rule*.1 から順に適用して判定する．

表 1 の *Rule*.1 に示した “有効ノード” についての説明を行う．DOM ノードの中には，Web ブラウザで Web ページをレンダリングした際に，実際に Web ページ上に表示されるものとされないものが存在する．表示されるノードを本論文では “有効ノード” と呼び，表示されないノードを “無効ノード” と呼ぶ．DOM ノードが次の四つの条件を満たすとき，その DOM ノードは有効ノードとなる．一つ目の条件は，DOM ノードの横幅 (pixel) と縦幅 (pixel) がともに 1 以上であること，二つ目の条件は，DOM ノードの右下の座標 (x, y) が，$x > 0$ かつ $y > 0$ を満たすことである．三つ目の条件は，DOM ノードの display スタイルが “none” でないことである．display スタイルに “none” が指定された要素は Web ブラウザ上に表示されない．最後に，四つ目の条件は，visibility スタイルが “hidden” でないことである．visibility スタイルに “hidden” が指定された要素は Web ブラウザ上に表示されない．

次に，表 1 の *Rule*.2 と *Rule*.3 についての説明を行う．DOM ノードに CSS で付加されるスタイルのうち，display スタイルではインライン要素とブロックレ

(注1)：http://www.w3.org/TR/html401/struct/global.html

表 1　ブロックレベル要素判定ルール
Table 1　Rules to classify elements as either block-level elements or inline elements.

ルール番号	詳細
Rule.1	n_i が有効ノードでないならブロックレベル要素ではない．
Rule.2	n_i の display スタイルが “block” であるならブロックレベル要素である．
Rule.3	n_i が以下に示すタグであるならブロックレベル要素である． [p, blockquote, pre, div, noscript, hr, address, fieldset, legend, h1, h2, h3, h4, h5, h6, ul, ol. li, dl, dt, dd, table, caption, thead, tbody, colgroup, col, tr, th, td]
Rule.4	*Rule*.2，*Rule*.3 でブロックレベル要素と判定されなかったノード n_i はブロックレベル要素ではない．

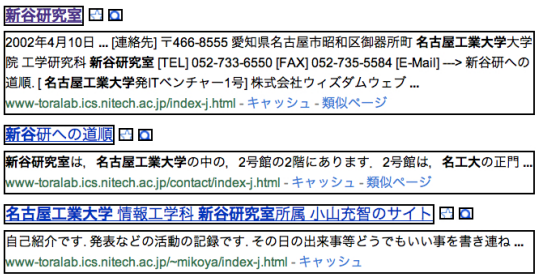


図 3　Web ページから抽出された細分化ブロックの例
Fig. 3　Example of twelve minimum-blocks extracted from a Web page.

表 2　タイトルブロック判定パラメータ
Table 2　Nine parameters to classify minimum-blocks into title-blocks or not.

特徴量	詳細
F_1	テキストノード長
F_2	テキストノードの面積 / ノード全体の面積
F_3	画像ノードの面積 / ノード全体の面積
F_4	ブロックの横幅 / 高さ
F_5	下隣接ブロックの面積がノードの面積より大きいかどうか
F_6	<H1>, <H2>, <H3>, <H4>, <H5>, <H6>, <DT>タグかどうか
F_7	同じ HTML タグ名が上隣接方向に連続している数
F_8	同じ HTML タグ名が下隣接方向に連続している数
F_9	下位 DOM ノードの合計数

ベル要素の指定が可能である．そのため，*Rule*.3 に示したタグであっても，display スタイルを “inline” に指定した場合，インライン要素となる．そこで *Rule*.3 でタグを判定する前に，*Rule*.2 で display スタイルの判定を行う必要がある．

上記のルールに従い，Web ページを細分化ブロックへと分割する．図 3 は Google で “名古屋工業大学 新谷研究室” と検索した結果の Web ページから，検索結果の上位 3 件の部分を切り取ったスクリーンショットである．この図の中には実線で囲った 12 個の細分化ブロックが存在する．

4.2　タイトルブロックの抽出

図 3 の中には三つのタイトルブロックが存在する．“新谷研究室” というテキストをもつ細分化ブロック，“新谷研への道順” というテキストをもつ細分化ブロック，“名古屋工業大学 情報工学科 新谷研究室所属 小山充智のサイト” というテキストをもつ細分化ブロックの三つである．これら三つの細分化ブロックは直下に存在するブロックのタイトルを表しているため，タイトルブロックとみなすことができる．また，図 3 のスクリーンショットは，これらのタイトルブロックを区切りとして意味的に三つに分割できる．

本研究で提案する Web ページ分割アルゴリズムでは，細分化ブロックをタイトルブロックとタイトルブロック以外のブロックへと分類する必要がある．本研究では機械学習によって分類器を生成した．機械学習にはレイアウトに基づく特徴量と，HTML のタグ及び DOM 構造に基づく特徴量を用いた．表 2 に示す F_1 から F_9 の，九つの特徴量を用いた．以下に，それぞれの特徴量の簡単な説明と，これらの特徴量を導入した理由を述べる．

F_1 から F_5 はレイアウトに基づく特徴量である．タイトルブロックは下隣接ブロックの内容を簡潔に表すキーワード・文章で構成されるため，テキストノード長は短くなり (F_1)，ブロック内部でテキストノードの占める面積の割合が大きくなる (F_2)．同時に，画像が占める面積の割合は小さくなる (F_3)．画像をほとんど含まず主にテキストノードで構成されるため，タイトルブロックは高さに比べて横幅が大きくなる (F_4)．タイトルブロックは下隣接ブロックの内容を簡潔に表すキーワード・文章であるため，下隣接ブロックよりも面積が小さくなる (F_5)．

F_6 から F_9 は HTML のタグ及び DOM 構造に基づく特徴量である．<H1>，<H2>，<H3>，<H4>，<H5>，<H6>タグは見出しを記述するために定義されたタグである．また，<DT>は Definition Term の略であり，<DD>タグとセットで利用される．<DT>タグの中に定義語を記述し，<DD>タグの中にはその用語の説明を記

述する．つまり，<DT>タグは<DD>タグに記述した内容のタイトルを表しているといえる (F_6)．上下隣接方向に同じ HTML タグが連続することは，そのブロック自身が隣接するブロックと並列関係にあることを意味する．タイトルブロックは直下に存在するコンテンツの見出しとなるブロックであり，タイトルブロック自身が連続して出現することはない．したがって，同じ HTML タグが隣接して連続する可能性は低い (F_7, F_8)．タイトルブロックは背景色やフォントで装飾するだけの HTML で記述される傾向にあるため，タイトルブロックがもつ DOM ノードの下位ノード数は少なくなる (F_9)．

これらの特徴量を用いて機械学習を行い，タイトルブロックの分類器を作成する．タイトルブロックの判定は，ブロックが “タイトルブロックである” 若しくは “タイトルブロックでない” の 2 クラス分類問題であり，また枝刈りによる過学習の防止が行えるという理由から，決定木学習によって分類器を生成した．比較対象として，サポートベクターマシンによる分類器も生成した．訓練データは，予備実験において，人手で細分化ブロックをタイトルブロックとそれ以外のブロックに分類したものである．訓練データの作成方法や作成した分類器の性能については，**5.** の評価実験で詳しく述べる．

4.3 細分化ブロックの結合

タイトルブロックを用いて細分化ブロックをコンテンツブロックへと結合する．Web ページの閲覧者が Web ページ中で認識する意味的にまとまりのある単位のことを Web コンテンツと呼ぶが，コンテンツブロックとは，Web コンテンツを形成する細分化ブロックの集合である．図 4 に細分化ブロックの結合アルゴリズムを疑似言語で示す．本アルゴリズムに対して細分化ブロックの集合を入力すると，それぞれの細分化ブロックをコンテンツブロック単位へとまとめ，コンテンツブロックの集合を返す．結合の基本的なパターンは，タイトルブロックと下方向に隣接する一般ブロックを，タイトルブロックが出現するまで繰り返し結合していくというパターンである．

アルゴリズムの各ステップに対して詳細な説明を行う．入力された細分化ブロックの集合からタイトルブロックを一つ取り出し（04，05 行目），まずはそのタイトルブロックを一時的なコンテナへと格納する（06 行目）．以降，幅優先探索によって，結合するブロックを決定していく．下方向に隣接している細分化ブロックを取得し（21 行目），取得した細分化ブロックの中にタイトルブロックが含まれているかどうかをチェックする（24〜29 行目）．タイトルブロックが含まれて

```
入力: Web ページ中に存在する細分化ブロックの集合
      MB = {mb_1, mb_2, ..., mb_n}
出力: コンテンツブロックの集合
      CB = {CB_1, CB_2, ..., CB_m}
01: procedure MakeContentBlocks(MB)
02: begin
03:   CB ← {}
04:   foreach mb ∈ MB do
05:     if IsTitleBlock(mb) = TRUE then
06:       Container = {mb};
07:       left ← ∞; top ← mb.top;
08:       x ← −1; y ← −1;
09:       Q = {mb};
10:       while Q.length > 0 do
11:         b ← Q.shift();
12:         if left > b.left then
13:           left ← b.left;
14:         endif
15:         if x <= b.left + b.width then
16:           x ← b.left + b.width;
17:         endif
18:         if y <= b.top + b.height then
19:           y ← b.top + b.height;
20:         endif
21:         Belows ← b.getBelowBlocks();
22:         Q ← Q ∪ Belows;
23:         flag ← FALSE;
24:         foreach b_2 ∈ Belows do
25:           if (IsTitleBlock(b_2) = TRUE) then
26:             flag ← TRUE;
27:             break;
28:           endif
29:         enddo
30:         if flag = FALSE then
31:           Container ← Container ∪ Belows;
32:         elseif
33:           foreach added ∈ Container do
34:             MB.delete(added);
35:           enddo
36:           foreach mb_2 ∈ MB do
37:             if InRect(left, top,
                          x − left, y − height,
                          mb_2) = TRUE then
38:               Container ← Container ∪ mb_2;
39:               MB.delete(mb_2);
40:             endif
41:           enddo
42:           CB ← CB ∪ {Container};
43:           break;
44:         endif
45:       enddo
46:     endif
47:   enddo
48:   return CB;
49: end.
```

図 4　細分化ブロックの結合アルゴリズム
Fig. 4　Algorithm to assemble minimize-blocks into content-blocks.

いなければ，それら細分化ブロックをコンテナへ追加する（30，31 行目）．タイトルブロックが含まれていれば，コンテナへの細分化ブロック追加を終了する．その時点でコンテナに格納されている細分化ブロックの集合が，コンテンツブロックということになる．コンテナに格納されている細分化ブロックを細分化ブロックのリストから削除する（33〜35 行目）．コンテナに格納されている細分化ブロックの集合が作る方形の中に存在する細分化ブロックを，コンテナに追加する（36〜41 行目）．コンテナをコンテンツブロックの集合へ追加し，幅優先探索を終了する（42，43 行目）．上記の処理を，全てのタイトルブロックに対して行う．

本アルゴリズムでは，隣接関係を用いた結合処理の後に，方形情報を用いた結合処理（36〜41 行目）を行っている．具体例を図 5 に示す．図 5 の左のように，タイトルブロックが 2 個 (tb_1, tb_2)，一般ブロックが 5 個 ($ob_1, ob_2, \cdots, ob_5$) の，合計 7 個の細分化ブロックが存在する場合を考える．タイトルブロック tb_1 に着目した場合，まずは tb_1 がコンテナへと格納される．すなわち，$Container = \{tb_1\}$ である．tb_1 の下に隣接する細分化ブロックは，ob_1 と ob_2 である．これらは 2 個とも一般ブロックであるため，結合処理を進める．ob_1 と ob_2 がコンテナへと格納され，$Container = \{tb_1, ob_1, ob_2\}$ となる．ob_1 の下に隣接する細分化ブロックは，ob_4 である．ob_4 は一般ブロックであるため，結合処理を進める．ob_4 がコンテナへと格納され，$Container = \{tb_1, ob_1, ob_2, ob_4\}$ となる．ob_4 の下に隣接する細分化ブロックは，tb_2 である．tb_2 はタイトルブロックであるため，隣接関係を用いた結合処理を終了する．この時点で tb_1, ob_1, ob_2, ob_4 の四つの細分化ブロックの結合を完了した（図 5 中央参照）．ob_3 は tb_1 と隣接していないため，上記の処理を行っただけでは ob_3 は結合されないという問題が発生する．この問題を解決するために，次のステップとして，結合された細分化ブロックの集合が形成する方形内部に位置する細分化ブロックも，同一のコンテンツブロックへ結合する．図 5 では，コンテンツブロック CB_1 の方形内に ob_3 が存在する．したがって ob_3 も CB_1 に結合した後に，結合処理を終了する（図 5 右参照）．

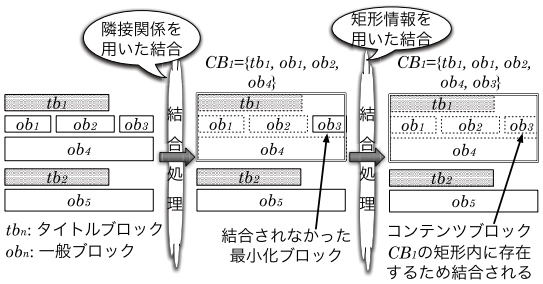


図 5　結合ステップ
Fig. 5　Assemble steps.

図 5 では簡略化のためにコンテンツブロックが 1 段組の図を示したが，本アルゴリズムでは，Web ページが 1 段組であることを仮定しない．コンテンツブロックが複数の段組でレイアウトされた Web ページには，タイトルブロックも複数の段組で存在する．それぞれのタイトルブロックに対して直下に存在する一般ブロックを結合していくという処理を繰り返し行っていくため，全てのタイトルブロックに対して処理を完了したときには，Web ページが複数の段組へと分割される．

5. 実験・考察

5.1 予備実験：タイトルブロックの有無の調査

Web ページ内にタイトルブロックが複数存在し Web コンテンツ間のセパレータとなっていることを確認するため，予備実験を行った．複数の Web ページを収集し，それら Web ページの中にタイトルブロックが存在するかどうかを調査した．

2011 年 9 月 7 日 15 時時点での Google トレンド(注2)上位 10 件のキーワード(注3)をクエリーとして Google で検索を行い，それぞれのクエリーに対して検索結果の上位 5 件の Web ページを取得した．取得した 50 件の Web ページにおいて，本研究で実装したシステム（ブックマークレット）が動作しなかった Web ページが 4 件存在した(注4)ため，それら 4 件の Web ページは実験対象から除外した．すなわち実験対象とした Web ページは全部で 46 件である．これら 46 件の Web ページの中には，企業ページ，ブログ，Wikipedia，2 ちゃんねる，ニュースサイトのような典型的なデザインを有する Web ページだけでなく，個人が Web オーサリングツールを用いて作成した Web

(注2)：http://http://www.google.co.jp/trends
(注3)：1 位から順に，“日向燦”，“武田邦彦”，“築地銀だこ”，“深頚部膿瘍”，“担々麺本舗 辣椒漢”，“王座戦”，“NOPOPO”，“科学技術館”，“渡辺美優紀”，“MH3G”
(注4)：Web ページの JavaScript で宣言されていた変数と本システムで宣言した変数の衝突などが原因

ページも含まれていた．実験対象の Web ページをシステムによって細分化ブロックへと分割した後に，細分化ブロックを被験者 7 名に提示した．被験者にはそれら細分化ブロックがタイトルブロックであるか否かの分類を依頼した．分類を行ったのは名古屋工業大学の情報工学科に所属する学部 4 年の学生 7 名である．同じ細分化ブロックであっても評価者によってタイトルブロックか否かの判定が分かれることがあったため，その場合は多数決によってタイトルブロックか否かを決定した．

実験対象とした Web ページ 46 件の中には，平均して 17 個/ページのタイトルブロックが含まれていることを確認した．タイトルブロックを全く含まないと判断された Web ページはわずか 1 件のみであった．その Web ページは，ページ全体が Flash で構成されていた．上記の実験結果により，タイトルブロックを Web コンテンツ間の区切りとして Web ページ分割するというヒューリスティクスが利用できると判断した．

5.2 タイトルブロックの抽出精度

提案手法によって得られる Web ページ分割結果の精度は，タイトルブロックの判定精度によって左右される．J4.8 アルゴリズムによる決定木学習，ランダムトリーアルゴリズムによる決定木学習，サポートベクターマシンによって 3 種類の分類器を作成した．タイトルブロック判定精度を分類器生成時の 10 分割交差検定で測定する．

5.1 の予備実験で収集したデータを訓練データとして用いた．評価基準は，タイトルブロックを正しく判定した数 (a)，一般ブロックを正しく判定した数 (b)，タイトルブロックを一般ブロックと判定した数 (c)，一般ブロックをタイトルブロックと判定した数 (d) の四つで行う．また，タイトルブロックの判定精度 P_{tb}，一般ブロックの判定精度 P_{ob}，タイトルブロックの再現率 R_{tb}，一般ブロックの再現率 R_{ob} を以下の式で求める．

$$P_{tb} = \frac{a}{a+d}, \quad P_{ob} = \frac{b}{b+c}$$

$$R_{tb} = \frac{a}{a+c}, \quad R_{ob} = \frac{b}{b+d}$$

また，それぞれの F 尺度 F_{tb}，F_{ob} を以下の式で求める．

$$F_{tb} = \frac{2 \cdot P_{tb} \cdot R_{tb}}{P_{tb} + R_{tb}}, \quad F_{ob} = \frac{2 \cdot P_{ob} \cdot R_{ob}}{P_{ob} + R_{ob}}$$

表 3 に，人手で判定した結果を訓練データとして分類器で学習した際の 10 分割交差検定の結果を示す．学習器に入力したブロック数はタイトルブロックが 782 個，一般ブロックが 1542 個の，合計 2324 個である．表 3 の “J4.8”，“RT”，“SVM” はそれぞれ，J4.8 アルゴリズムによる決定木学習，ランダムトリーアルゴリズムによる決定木学習，サポートベクターマシンを表している．

表 3　タイトルブロックの判定精度と再現率
Table 3　Precision and recall in extracting title-blocks.

	J4.8	RT	SVM
a:タイトルブロックを正しく判定した数	588	593	424
b:一般ブロックを正しく判定した数	1401	1338	1395
c:タイトルブロックを誤判定した数	194	189	358
d:一般ブロックを誤判定した数	141	204	147
P_{tb}: タイトルブロックの判定精度	0.807	0.744	0.743
R_{tb}: タイトルブロックの再現率	0.752	0.758	0.542
F_{tb}: タイトルブロックの F 尺度	0.778	0.751	0.627
P_{ob}: 一般ブロックの判定精度	0.878	0.876	0.796
R_{ob}: 一般ブロックの再現率	0.909	0.868	0.905
F_{ob}: 一般ブロックの F 尺度	0.893	0.872	0.847

生成された決定木を観察することで，タイトルブロックの判定には表 2 に示した特徴量 F_6，F_5 が大きな影響を与えることが分かった．決定木の根に最も近いところでは，特徴量 F_6，つまり，`<H1>`，`<H2>`，`<H3>`，`<H4>`，`<H5>`，`<H6>`，`<DT>`タグで記述されているかどうかによって，分岐していた．企業や団体などの Web サイトや，個人ユースでのブログ管理には，コンテンツマネジメントシステムが用いられる．コンテンツマネジメントシステムによって作成された Web サイトではタイトルブロックとしてこれらのタグを用いて記述される場合が大半であるが，個人が作成した Web ページでは，スタイルによってフォントサイズを変更したり，また HTML4 では非推奨とされている`<font>`タグを用いて直下に存在するブロックよりもフォントサイズを大きくすることによってタイトルが表現されていることがあった．そのような場合はタイトルブロックであるにもかかわらず一般ブロックであると誤判定されやすい．今回，特徴量 F_5 で下に隣接するブロックとの面積サイズの大小関係を考慮したが，面積サイズだけではなく，フォントサイズの大小も比較するべきであったといえる．下に隣接するブロックのフォントサイズとの大小関係という特徴量を考慮することにより，精度・再現率が改善する可能性がある．

5.3 Web ページ分割結果

5.2 で作成した決定木によってタイトルブロック

図 6　提案手法による Web ページの分割結果
Fig. 6　Segmentation result by proposed method.

を抽出し，それらを利用して Web ページ分割を行った．実行例を図 6 に四つ示す．方形が分割結果を表している．方形が存在しないところは，システムによって分割されなかったところである．図 6 の (a) は Yahoo!ニュース(注5)のニュース記事，(b) は Ameba ブログ(注6)のブログ記事，(c) は Amazon.co.jp のトップページ(注7)，(d) は新谷研究室のトップページ(注8)である．(a)，(b) は分割が成功していると判断された結果，(c) はほぼ成功していると判断された結果，(d) は失敗と判断された結果である．

決定木学習アルゴリズムには J4.8 アルゴリズムを採用した．一般ブロックをタイトルブロックと誤判定した場合，本来であれば上に隣接するブロックと結合されるべきであるが，図 7 の (1) に示すように，結合されずに別々のコンテンツブロックとして分割されてしまう．逆に，タイトルブロックを一般ブロックと誤判定した場合，本来であれば上に隣接するブロックとは別のコンテンツブロックとして分割されるべきであるが，図 7 の (2) に示すように，分割されずに一つのコンテンツブロックとして結合されてしまう．Web の閲覧者は，図 7 (1) のように分割すべきではないところ

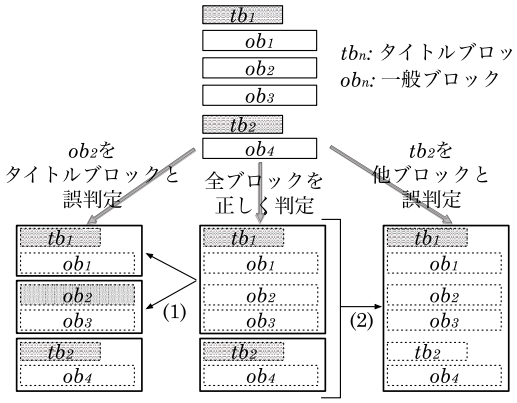


図 7　誤判定によって意図しないコンテンツブロックが生成される例
Fig. 7　Examples of assembling incorrect Web content blocks due to errors in identifying title/non-title blocks.

で細かいブロックに分割されると明らかに間違った分割結果であると判断するが，図 7 (2) のように一つの大きなブロックとして分割された場合には間違った分割結果であるとは判断しないことが多い．そのような分割を行うためには，一般ブロックをタイトルブロックとして誤判定する数を減らすことが重要である．つまり，タイトルブロックの判定精度 P_{tb}，及び一般ブロックの再現率 R_{ob} が高くなるようなアルゴリズムを採用すればよい．表 3 に示した実験結果により，J4.8 アルゴリズムを採用することとした．

提案手法では，Web ページ中のタイトルブロックが存在しないところではコンテンツブロックへの結合処理が行われないため，Web ページ中に分割されない領域が発生する．(d) では，タイトルブロックが二つしか抽出されなかったため，コンテンツブロックも二つしか生成されず，Web ページのほぼ大半が分割されなかった．(d) の Web ページ右上には，画像を用いて表現されているタイトルブロックが四つ存在するが，本研究で生成した分類器ではこれらのタイトルブロックを一般ブロックと誤判定した．これは決定木学習で利用した訓練データの中に，画像を用いて表現されているタイトルブロックがあまり含まれていなかったことが原因である．訓練データを見直し，画像を用いてタイトルを表現しているタイトルブロックの抽出精度を

(注5)：http://headlines.yahoo.co.jp/hl
(注6)：http://ameblo.jp/
(注7)：http://www.amazon.co.jp/
(注8)：http://www-toralab.ics.nitech.ac.jp/index-j.html

上げることによって (d) のような Web ページの分割精度を上げることが可能であり，本研究の今後の課題である．

5.4 コンテンツ量に依存しない分割結果

既存研究で提案されている分割手法の問題点として，同じ Web サイト内に存在する同一レイアウトの Web ページでさえも，メインコンテンツのテキスト量が変化した場合に異なった分割結果が得られるという問題があった．提案手法によってメインコンテンツのテキスト量が変化しても同一の分割結果が得られることを確認するために，実験を行った．

本実験では Yahoo!ニュースを実験対象の Web サイトとした．Yahoo!ニュースでは，“国内”，“海外”，“経済”，“エンターテインメント”，“スポーツ”，“テクノロジー”，“地域” の 7 ジャンルにおいて，アクセスランキングが提供されている．2011 年 10 月 24 日 2 時の時点での，それぞれのジャンルのアクセスランキング上位 20 件のニュース記事ページを対象に実験を行った．すなわち実験対象としたニュース記事ページは合計 140 件である．Yahoo!ニュースで配信されているニュース記事ページにおいて，ニュース記事部分は図 8 に示す (a) から (f) の六つから構成されている．(a) はニュース記事のタイトル，(b) は配信日時，(c) はニュース記事本文，(d) は関連記事，(e) は最終更新日時，(f) は一次配信元サイトのロゴである．提案手法を用いて Web ページ分割を行い，これら六つが一つのコンテンツブロックへと結合されるかどうかを確認した．ただし，(d) が存在しないニュース記事も存在するため，その場合は (d) を除く五つを対象とする．六つが一つのコンテンツブロックへと結合された場合を分割成功とし，二つ以上のコンテンツブロックへと結合された場合を分割失敗とする．また，他の細分化ブロックが対象のコンテンツブロックへと結合された場合も，分割失敗とみなす．

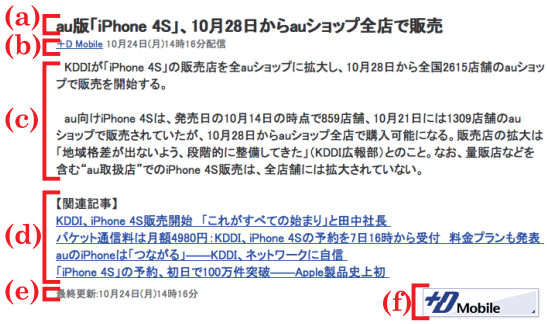


図 8 Yahoo!ニュースのニュース記事部分の構成
Fig. 8 Struct of news article in Yahoo!News Web site.

実験を行った結果，140 件中 135 件の Web ページで分割に成功した．精度は 96.4%であり，十分に実用的であるといえる．分割に失敗した 5 件では，以下のような分割が行われていた．ジャンルが経済のニュース記事ページでは，(f) 一次配信元サイトのロゴの下に，ニュース記事と関連する株価を表す細分化ブロックが配置されているページがあった．これらのページでは，(a) から (f) に加え，株価を表す細分化ブロックも同一のコンテンツブロックに分割された．このような失敗は 3 件存在した．

ニュース記事本文の中に画像と画像のキャプションが配置されるニュース記事ページもあるが，そのようなページの中で，キャプションがタイトルブロックと判定されたニュース記事ページが存在した．画像のキャプションをセパレータとして，二つのコンテンツブロックへと分割された．このような失敗は 2 件存在した．**5.2** で述べたようにタイトルブロックの抽出精度を改善することにより分割精度を向上させることが可能であり，本研究の今後の課題とする．

6. むすび

本論文では Web ページを細分化ブロックと呼ばれる非常に細かい単位まで分割した後に，直下の Web コンテンツの内容を説明するタイトルブロックに着目した細分化ブロックの結合を行うことによって，Web ページを意味的にまとまりのある単位へと分割を行う手法を提案した．計算機によるタイトルブロックの自動抽出を行うために，機械学習による分類器を作成した．J4.8 アルゴリズムによる決定木学習によって生成した分類器により，F 値 77.8%，89.3%でタイトルブロックとタイトルブロック以外のブロックの抽出に成功した．タイトルブロックと下方向に隣接する一般ブロックを結合することで，Web ページをコンテンツ単位へと分割する手法を示した．実験によって，96.1%の精度でコンテンツ量に依存しない同一の分割結果が得られることを確認した．

謝辞 本研究の一部は科研費（22500128）及び総務省 SCOPE の助成を受けたものである．

文　献

[1] L. Yi, B. Liu, and X. Li, “Eliminating noisy information in web pages for data mining,” Proc.

Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '03, pp.296–305, ACM, New York, NY, USA, 2003. http://doi.acm.org/10.1145/956750.956785

[2] G. Hattori, K. Hoashi, K. Matsumoto, and F. Sugaya, "Robust web page segmentation for mobile terminal using content-distances and page layout information," Proc. 16th International Conference on World Wide Web, WWW '07, pp.361–370, ACM, New York, NY, USA, 2007.

[3] H. Guo, J. Mahmud, Y. Borodin, A. Stent, and I. Ramakrishnan, "A general approach for partitioning web page content based on geometric and style information," Proc. Ninth International Conference on Document Analysis and Recognition - Volume 02, IEEE Computer Society, pp.929–933, Washington, DC, USA, 2007.

[4] T. Ito, H. Sano, T. Ozono, and T. Shintani, "A hierarchical web page segmentation algorithm using machine learning," 2008.

[5] 伊藤太樹，浅見昌平，大囿忠親，新谷虎松，"Svm に基づくテンプレートを考慮した web ページの分割手法について（「web インテリジェンス」及び一般)," 信学技報, vol.108, no.119, pp.81–86, June 2008.

[6] 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.

[7] D. Cai, S. Yu, J.-R. Wen, and W.-Y. Ma, "Extracting content structure for web pages based on visual representation," Proc. 5th Asia-Pacific web conference on Web technologies and applications, APWeb'03, pp.406–417, Springer-Verlag, Berlin, Heidelberg, 2003.

[8] S. Baluja, "Browsing on small screens: Recasting web-page segmentation into an efficient machine learning framework," Proc. 15th International Conference on World Wide Web, WWW '06, pp.33–42, ACM, New York, NY, USA, 2006.

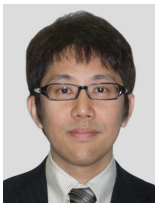
（平成 23 年 6 月 20 日受付，10 月 25 日再受付）

佐野　博之

2010 名古屋工業大学大学院工学研究科博士前期課程了，同年より同大学院工学研究科博士後期課程進学，現在に至る．知的 Web 技術の研究に従事．ACM，IEEE，情報処理学会，人工知能学会，日本ソフトウェア科学会各学生会員．

白松　俊

2003 東京理科大学大学院修士課程了．2003〜2005 にかけて JST CREST 研究補助員として産業技術総合研究所に勤務．2007 日本学術振興会特別研究員 (DC2)．2008 京都大学大学院情報学研究科博士後期課程了．2008 日本学術振興会特別研究員 (PD)．2009 名古屋工業大学大学院工学研究科助教，現在に至る．博士（情報学）．談話文脈のモデル化，議論支援の研究に従事．情報処理学会，言語処理学会各会員．



大囿　忠親　（正員）

2000 名古屋工業大学大学院工学研究科電気情報工学専攻博士後期課程了．同年より同大学工学部知能情報システム学科助手．2003–2004 にかけてマレイシア・マルチメディア大学客員研究員．2006 同大学助教授．2007 同大学准教授，現在に至る．博士（工学）．Web インテリジェンスの研究に従事．AAAI，ACM，情報処理学会，日本ソフトウェア科学会各会員．

新谷　虎松　（正員）

1982 東京理科大学大学院修士課程了．同年富士通（株）国際情報社会科学研究所入所．1993 名古屋工業大学工学部知能情報システム学科助教授．1999 同大学同学科教授．1999〜2000 にかけて米国カーネギーメロン大学ロボティクス研究所客員教授．2004 名古屋工業大学大学院工学研究科情報工学専攻教授．博士（工学），現在に至る．知的 Web 技術，マルチエージェント，知的意思決定支援の研究に従事．2004（株）ウィズダムウェブ設立創業者代表取締役．2004 情報処理学会全国大会優秀賞受賞．2008 情報処理学会フェロー．AAAI，情報処理学会各会員．