CSSセレクタで表現されたコンテンツ抽出ルールの自動獲得

吉田光男[†1] 乾 孝司[†1] 山本幹雄[†1]

近年のWebページの増加により，Webページのコンテンツを利用するサービスや研究が盛んになってきている．本論文では，Webページ集合を用いる事により，CSSセレクタで表現されたコンテンツ抽出ルールを自動的に獲得する手法を提案する．また，本手法のアルゴリズムを実装したソフトウェアを用いて実験を行い，日本語ブログサイトに対して適切な抽出ルールが獲得できた事を示す．

Automatic Generation of Rules on CSS Selector of Primary Content

Mitsuo Yoshida,[†1] Takashi Inui[†1] and Mikio Yamamoto[†1]

In recent years, with the increase in the number of Web pages, researches and Web services using the (primary) content of the pages are actively pursued. In this paper, we propose a simple method to generate special CSS selectors as rules of extracting primary content from a collection of Web pages. We show that the proposed method can accurately extract the primary content from real pages of Japanese blog sites.

1. はじめに

インターネットが普及した今日，様々なユーザがWebページを作成し，インターネット上には大量の情報があふれている．近年のWebページの増加は，ブログの普及に一因がある．我が国では，2004年から2005年ごろにかけてブログ及びその記事が急増しており，現在も増加傾向が見られる[1)]．ブログの普及に伴い，ブログのコンテンツを利用する研究も盛んになってきている．

ブログに限らないが，最近のWebページには，ヘッダ，メニュー，広告，関連記事リストなど不要部分が多々存在する事によりページに占めるコンテンツ（主要部分）の割合が低い．そのため，ブログのコンテンツを利用するためには，コンテンツ抽出が必要になる．

コンテンツ抽出手法の代表例として，人手によって抽出ルールを記述する方法が挙げられる[2)]．しかし，インターネット上には無数のブログページが存在しており，各ブログページに適した抽出ルールを定める事は，ブログサイトごとに定めるにしても，大きな労力を必要とする．情報通信政策研究所は，ブログユーザの85.8%が主要なブログホスティングサービスを利用していると報告しているが[1)]，筆者らによる予備調査では，人気のあるブログサイトの少なくとも30%がブログホスティングサービスを利用していない事が明らかになった[⋆1]．

本論文では，筆者らが過去に提案したコンテンツ自動抽出手法[3)]を拡張し，コンテンツ抽出ルールを自動的に獲得する手法を提案する．抽出ルールを獲得する事により，単一のページWebページを対象としたコンテンツ抽出が可能になり，高速にコンテンツを抽出できるようになる．また，獲得する抽出ルールは，様々なソフトウェアで再利用しやすいようにCSS（Cascading Style Sheets）[⋆2]のセレクタで表現する．過去に提案したコンテンツ自動抽出手法は，HTMLのブロックレベル要素を基にしたブロック（コンテンツと不要部分の最小単位）を用いて，他のページに出現しないブロックをコンテンツとして抽出する．そのため，事前に一切の教師情報を必要としない利点がある．

2. 提案手法

2.1 コンテンツの定義

一般的に，Webページはユーザが必要とする**コンテンツ**（主要部分）と，必要としない**不要部分**から成り立っている．本論文では，ブログページの記事本文をコンテンツとし，記事本文に付随する記事タイトル，記事日時，著者名，写真・図，写真・図の説明文もコンテンツとみなす．また，ブログの読者によるコメント及びトラックバックの本文，それら本文に付随するタイトル，投稿日時，コメント著者名（トラックバッ

†1 筑波大学大学院 システム情報工学研究科
Graduate School of Systems and Information Engineering, University of Tsukuba

⋆1 livedoor Reader（http://reader.livedoor.com/）上位1,000サイトのホスト名から推定した

⋆2 「スタイルシート」とも呼ばれる

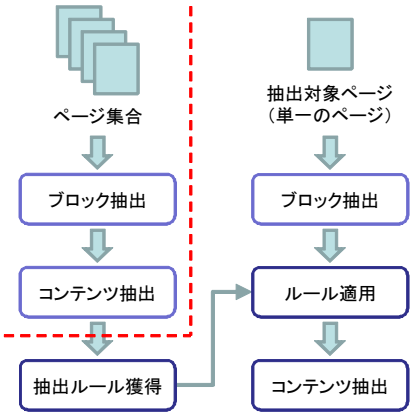

図 1 提案手法の概観
Fig. 1 The overview of the proposed method.

ク送信元ブログ名）もコンテンツとみなす．そして，それら以外の全てを不要部分とする．

2.2 コンテンツ抽出ルール自動獲得手法の概要

本論文で提案するコンテンツ抽出ルールを自動的に獲得する手法（以下，**提案手法**と呼ぶ）は，筆者らが過去に提案したコンテンツ自動抽出手法[3]（以下，**従来手法**と呼ぶ）の拡張である．本小節では，提案手法の概要と，従来手法との差異について述べる．

従来手法は，Web ページ集合を前提とするため，単一のページを与えてのコンテンツ抽出が困難であるという課題があった．この課題を解決するために，従来手法によるコンテンツ抽出結果から抽出ルールを獲得すればよいと考えた．具体的には，コンテンツとして認められるブロックを，Web ページ集合全てで利用できる CSS のセレクタで表現し，その表現を抽出ルールとする．

提案手法の概観を**図 1** に示す．従来手法は左側の破線部分の処理である．提案手法はコンテンツとして認められるブロックを表現する「抽出ルール獲得」,「ルール適用」の処理を加え,「コンテンツ抽出」を行う．

2.3 ブロック抽出・コンテンツ抽出

提案手法は，教師情報を必要としない従来手法によって抽出されたコンテンツ部分を用いる事で，自動的に抽出ルールを獲得する．本小節では，前提となる従来手法の概要を以下の 3 過程にわけて解説する．詳細については，文献 3) を参照されたい．

処理 1 Web ページ集合の準備
処理 2 ブロック抽出
処理 3 コンテンツ抽出

処理 1 では，コンテンツ抽出に必要な Web ページ集合の準備を行う．従来手法は，2 ページ以上の Web ページ集合を用いてコンテンツ抽出を行う．提案手法

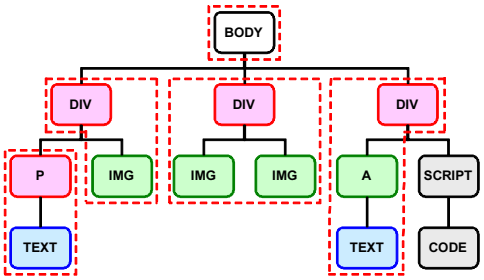

図 2 DOM ツリーから 5 つのブロックを抽出した例（破線部分）
Fig. 2 An example of extracted blocks (dashed-line box) from the DOM tree.

では Web サイトごとに抽出ルールを獲得するため，Web ページ集合は同じ Web サイトのページで構成されている必要がある．

処理 2 では，HTML のブロックレベル要素[4] を基に，コンテンツ及び不要部分の最小単位である**ブロック**の抽出を行う．ブロックレベル要素を用いてブロックを抽出する際，ブロックがコンテンツ及び不要部分の最小単位となるよう，DOM ツリー上の下位ノードにブロックレベル要素が存在しないように抽出する．ただし，ブラウザにレンダリングされない SCRIPT, NOSCRIPT, STYLE の 3 要素及びその下位ノードはブロック内に含めない．また，BODY 要素はブロックレベル要素ではないが，直下にブロックレベル要素以外が存在する HTML 構造にも対応するため，例外的にブロックとして認める．たとえば，**図 2** の DOM ツリー（属性は省略）からブロックを抽出すると，5 つのブロックが抽出される（破線枠部分）．

処理 3 では，ブロックの一致を判断し，他の Web ページには出現しないブロック，すなわち Web ページ集合の中で 1 度だけ出現するブロックをコンテンツとして抽出する．本論文では，このブロックを**コンテンツブロック**と呼ぶ．

2.4 抽出ルール獲得

本研究におけるブロックは，ブロックレベル要素を基にしているため，ブロックの位置は CSS のセレクタで表現できる[5]．また，ブロック内に複数のブロックレベル要素が含まれないため，ブロックに対応するブロックレベル要素は一意に定まる．提案手法で獲得する**抽出ルール**は，要素識別子（Element identifiers）と要素名との組を原則とし，セレクタで表現する．なお，要素識別子とは id 属性または class 属性の事である[4]．

セレクタには様々な構文が存在するが，提案手法では，表現を単純にするためにひな形を準備した．獲得する抽出ルールは，**表 1** のいずれかに合致する．

表 1 抽出ルールのひな形と獲得例
Table 1 The model of the extraction rule.

パターン	パターンが示す意味	獲得した抽出ルールの例（100SHIKI）
E#myid	id 属性の値が myid である E 要素	div#dotcom_logo, ul#flip1, ul#flip2
#myid > E	id 属性の値が myid である要素の子要素が E 要素	#column_box > h4, #column_box > p, #comments > dl, #comments > h2, #dotcom_box > h3, #dotcom_box > p, #main > h1, #main > p, #trackback > dl
#myid * E	id 属性の値が myid である要素の子孫要素が E 要素（ただし子要素は除く）	#comments * p,
E.myclass	class 属性の値が myclass である E 要素	
.myclass > E	class 属性の値が myclass である要素の子要素が E 要素	.textBody > p
.myclass * E	class 属性の値が myclass である要素の子孫要素が E 要素（ただし子要素は除く）	
E	E 要素	

抽出ルールとして使える要素識別子は以下の両条件を満たすものとする．本論文では，両条件を満たす要素識別子を**適合識別子**と呼ぶ．

- 1 ページ中に 1 度のみ出現する
- Web ページ集合全てのページに出現する

1 ページ中に 1 度のみ出現する要素識別子に絞る事で，DOM ツリー上において，要素識別子と関連付けられた要素のサブツリーを 1 カ所に絞る事ができる．すなわち，抽出ルールがマッチする箇所を絞る事ができる．また，Web ページ集合の中で一般化された抽出ルールを獲得するために，個々のページ特有の情報を持つ可能性が高い，Web ページ集合全てのページで出現しない要素識別子を除外する．

本論文では，あるブロックレベル要素から最も近い適合識別子を**近隣適合識別子**と呼ぶ．近隣適合識別子は，あるブロックレベル要素をマッチさせるとき，他にマッチするブロックレベル要素の数を最小にする性質がある．近隣適合識別子は，DOM ツリー上の各ブロックレベル要素を前順（preorder）に走査し，次の優先順位に従って順次獲得する事ができる．なお，id 属性と class 属性の両方を用いる事ができる場合は id 属性を用いる．

(1) 自身の適合識別子を採用する
(2) 親ノードの適合識別子を採用する
(3) 親ノードの近隣適合識別子を採用する
(4) 近隣適合識別子を獲得しない

抽出ルールは，コンテンツブロックの近隣適合識別子と要素名で表現する．コンテンツブロックの近隣適合識別子と要素名は，コンテンツブロックと対応するブロックレベル要素のものとする．抽出ルールは，表 1 で示したひな形に従い，近隣適合識別子を優先順位 (1) で決定した場合は「E#myid」「E.myclass」，(2) の場合は「#myid > E」「.myclass > E」，(3) の場合は「#myid * E」「.myclass * E」，(4) の場合は「E」のパターンで表現される．

2.5 ルール適用・コンテンツ抽出

抽出ルールの適用方法は，セレクタとブロックレベル要素がマッチする[5] のと同様，抽出ルールとマッチしたブロックをコンテンツとして抽出する．例外として，不要部分となる傾向が高い，テキストと IMG 要素のいずれも含まないブロックは，コンテンツと見なさず抽出しない．

3. 実験及び考察

livedoor Reader[⋆1]の登録数ランキング上位からブログ形式の 9 サイト，100SHIKI，Engadget Japanese，ネタフル，404 Blog Not Found，IDEA*IDEA，My Life Between Silicon Valley and Japan，Going My Way，TechCrunch Japan，Life is beautiful から 2009 年 4 月 11 日に収集した計 206 ページ（文献 6) で用いたデータセットと同様）を用いてサイトごとに抽出ルール獲得を試みた．獲得した抽出ルールの例として，100SHIKI[⋆2]から獲得した抽出ルールを表 1「獲得した抽出ルールの例（100SHIKI）」に示す．

獲得した抽出ルールを 100SHIKI で公開されている 2009 年 10 月 30 日の Web ページ[⋆3]に適用し，その結果を**図 3** に示す（破線部分が適用された箇所）[⋆4]．ナビゲーションリンクが過抽出されているものの，概ね適切に抽出されている事がわかる．

古い 3 ページを抽出ルール獲得に用い，それら以外のページに抽出ルールを適用してコンテンツ抽出を試

⋆1 http://reader.livedoor.com/
⋆2 http://www.100shiki.com/
⋆3 http://www.100shiki.com/archives/2009/10/listorious.html (cite 2009-10-30)
⋆4 Firefox 3.5.4 userContent.css を利用した

みる実験をサイトごとに行ったところ，適合率 69.3%，再現率 88.7%の結果（平均値）を得た．従来手法では適合率 92.3%，再現率 88.2%という結果が得られており，適合率の低下があるものの，再現率は維持できる事がわかった．また，従来手法に比べて実行時間を 20 分の 1 に削減できた．性能の計算方法の詳細は文献 3) を参照されたい．

上記実験において，ある 1 サイトで他のサイトよりも大幅な適合率低下が確認され，それが平均値の低下に大きく影響していた．その原因は，構造が乱れた HTML（Valid でない HTML）によるものであると推測され，筆者らが開発した HTML パーサによる実験では，適合率 22.9%，再現率 88.2%の結果を得たが，ルール適用のみブラウザ[⋆1]の HTML パーサを用いた実験では，適合率 69.2%，再現率 78.4%の結果を得た．一般的に，ブラウザは構造が乱れた HTML を柔軟にパースする事が知られており，この事から，構造が乱れた HTML に対する対応が，直近の課題であるといえる．

4. おわりに

本論文では，事前に教師情報を準備する必要のない単純なアルゴリズムで，Web ページ集合からコンテンツ抽出ルールを獲得する手法を提案した．また，日本語ブログサイトを対象とした実験により，提案手法の有効性を示した．

提案手法は，教師情報や閾値を決定するためのデータを必要としないため，非常に小さな労力で Web ページのコンテンツ抽出ルールを獲得する事ができ，また，抽出ルールの適用によって，単一の Web ページからもコンテンツを抽出する事ができる．獲得する抽出ルールは CSS のセレクタで表現されるため，様々なソフトウェアで利用しやすいと考える．また，従来手法に比べ，実行速度が大幅に改善しており，大量の Web ページ集合に対しても適用が容易である事が示唆される．

今後，獲得する抽出ルールのパターンを増やし，より柔軟に抽出ルールを獲得する方法を検討する．また，抽出ルールを XPath などの別書式で表現する方法を検討する．そして，本研究の成果をモジュール及びソフトウェアの形で公開し[⋆2]，Web ページを利用する研究の標準的な手法となる事を目指す．

⋆1 Firefox 3.5.4

⋆2 コンテンツ抽出部分は ExtractUniqueBlock という名前ですでに公開している `http://www.mibel.cs.tsukuba.ac.jp/~m.yoshida/ExtractUniqueBlock/`

図 3 抽出ルール適用例（破線部分）

Fig. 3 An example of a Blog page and the extraction rule application.

参考文献

1) 総務省情報通信政策研究所（IICP）：ブログの実態に関する調査研究 (2008).

2) Hemenway, K., Calishain, T., 村上雅章 (訳者)：Spidering hacks—ウェブ情報ラクラク取得テクニック 101 選，オライリー・ジャパン (2004).

3) 吉田光男，山本幹雄：教師情報を必要としないニュースページ群からのコンテンツ自動抽出，日本データベース学会論文誌， Vol. 8, No. 1, pp. 29–34 (2009).

4) Raggett, D., Hors, A. L. and Jacobs, I.: The global structure of an HTML document, *HTML 4.01 Specification*, World Wide Web Consortium (W3C) (1999).

5) Bos, B., Celik, T., Hickson, I. and Wium, L.H.: Selectors, *Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification*, World Wide Web Consortium (W3C) (2009).

6) 吉田光男，山本幹雄：教師情報を必要としないWeb ページ群の主要コンテンツ自動抽出，第 23 回人工知能学会全国大会 (2009).