情報処理学会論文誌 Vol.55 No.2 874–891 (Feb. 2014)

個別の詳細記事抽出のためのWebページ分割手法の提案

山本 雄平[1,a)] 中村 健二[2] 田中 成典[3] 安彦 智史[1]

受付日 2013年5月14日, 採録日 2013年10月9日

概要：インターネットに流通する有害情報から青少年を守る取り組みとして，ネットパトロールが行われている．ネットパトロールでは，有害情報が含まれる投稿記事を目視により確認しており，この作業を軽減するネットパトロール支援の研究が注目されている．ネットパトロール支援の研究の1つに，投稿者の見守りを目的としたWebクローラ開発の研究がある．この研究では，Webページを解析することで，情報の抽出を行っており，その中でWebページをブロック単位に分割する手法が用いられている．しかし，Webページをブロック単位に分割する場合，投稿記事が複数のブロックに分割される場合や，1つのブロックに複数の投稿記事が含まれる場合がある．そのため，効率的なネットパトロールを実現するには，Webページを投稿記事ごとに分割し，投稿記事を詳細に確認する必要がある．そこで，本研究では，Webページを記事単位に分割できる汎用的なWebページの分割手法を提案する．そして，本提案手法の有用性を検証するため，既存手法との比較実験を実施した．その結果，本提案手法が有用であることを証明した．

キーワード：Webマイニング，ネットパトロール，投稿記事抽出，有害情報，Webページ分割

Proposal Research of Web Page Segmentation Method for Extracting and Describing Each Article in Detail

Yuhei Yamamoto[1,a)] Kenji Nakamura[2] Shigenori Tanaka[3] Satoshi Abiko[1]

Received: May 14, 2013, Accepted: October 9, 2013

Abstract: An Internet monitoring effort called “Net Patrol” is conducted to protect young people from harmful materials circulating on the Internet. In carrying out Net Patrol, post content that contains harmful materials is checked by visual inspection. Researches of supporting Net Patrol to reduce this work are attracting attention. One of the researching of supporting Net Patrol is a research of developing a web crawler for the purpose of watching contributors who submit posts. In this research, materials are extracted by analyzing web pages, using a method of dividing a web page on a block-by-block basis. However, when dividing a web page on a block-by-block basis, one post may be divided into multiple blocks in some cases, and multiple posts may be contained in one block in other cases. In order to patrol the Internet effectively, it is necessary to split a web page on a post-by-post basis and check the post content in detail. In this research, we propose a universal method of dividing a web page on a post-to-post basis. We conducted comparative experiments with the exiting method to verify the usefulness of the proposed method. The results proved that the proposed method is useful.

Keywords: web mining, Net Patrol, post content extraction, harmful material, web page segmentation

1. はじめに

携帯電話やスマートフォンの利用者の低年齢化にともない，子どもに安心安全なインターネット環境の実現に向けた関心 [1] が高まっている．インターネットのコンテンツには，援助交際やネットいじめに関する有害情報も含まれており，青少年の健全育成に悪影響を与える様々な問題 [2], [3], [4], [5], [6] が発生している．そのための対策とし

[1] 関西大学大学院総合情報学研究科
Graduate School of Informatics, Kansai University, Osaka 569–1095, Japan
[2] 大阪経済大学情報社会学部
Faculty of Information Technology and Social Sciences, Osaka University of Economics, Osaka 533–8533, Japan
[3] 関西大学総合情報学部
Faculty of Informatics, Kansai University, Osaka 569–1095, Japan
[a)] yamamoto@kansai-labo.co.jp

て，ネットパトロールによるインターネットの監視などが行われている．ネットパトロールに関する取り組みでは，教育委員会や学校関係者による自主的な取り組み，民間企業を中心とした取り組みと非営利団体やボランティア団体による活動などがある．これらのネットパトロールは，一般的に人手によるサイトブラウジングが中心であり，blog や SNS，掲示板を含む CGM を対象に学校名や学校名の略称で検索して得られた Web ページを確認し，そのページに有害情報が含まれていれば関連機関へ報告する．また，最近のネットパトロールでは，各ユーザの行動を見守る活動もなされており，1 人のユーザのプロフィール情報や投稿内容，友人関係などをひとまとまりにした個人領域を特定し，個人領域内の記事に含まれるリンク関係から現在の交友関係や非行逸脱行動などの有無などを把握する取り組みが行われている．これらの取り組みを行うためには，Web サイトの中のコンテンツを投稿記事単位に分けて，ユーザごとに分類した「データセット」を手動で作成する必要がある．これらのデータセットを用いることで，たとえば，コミュニケーションサイトの投稿記事における，「明日」や「今度の金曜日」などの相対的な時間表現や「同じクラスの××ちゃん」などの間接的な表現と「投稿日がいつであるのか」や「投稿者が誰であるのか」などの情報とを関連付けて，実際の日時や投稿者を推定することが可能である．しかし，これらの取り組みを継続的に行うためには，膨大な人的コストがかかるという問題と，ネットパトロールを行う人物の IT スキルに調査精度が依存するという問題が生じている．そのため，これらの問題を解消し，業務を効率化するためには，ネットパトロールを機械的に支援する次のような技術が必要である．1 つ目は，Web サイトブラウジングを自動化し，blog や SNS，掲示板を含む CGM から有害情報や固有の学校名，または略称などの必要情報を発見・収集する「Web ページの有害度を判定する技術」である．2 つ目は，Web ページから個人領域を抽出し，リンク関係を解析する「Web クローラの技術」である．3 つ目は，Web ページを解析し，必要となるコンテンツを投稿記事単位に抽出する「記事分割の技術」である．

上記の技術を実現するために様々な関連研究が行われている．「Web ページの有害度を判定する技術」については，インターネット上の違法・有害情報を判定する研究 [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18] がある．インターネット上の違法・有害情報を判定する研究には SVM などの識別器を用いて判定する技術 [7], [8], [11], [13], [14], [15], [18] や，Web ページのテキスト構造を解析して，有害な Web ページを判定する技術 [9], [10]，Web ページ中の単語間における共起情報や単語の出現頻度に基づき判定する技術 [12], [16], [17] が提案されている．これらの研究をネットパトロールに用いることで，監視対象の大量の Web ページから自動的に危険な情報が含まれるもののみに絞り込むことができ，ネットパトロール業務の効率化が可能となる．しかし，これらの研究では，Web ページを対象として処理するため，ヘッダ，フッタ，メニューなどの Web サイト共通の部分とメインコンテンツなどの Web ページ個別の部分とを区別せずに有害情報の有無を判定する．そのため，Web サイト共通の部分に有害情報が含まれた場合，そのサイトに属するすべての Web ページが有害情報と見なされ，詳細に確認する必要がない Web ページも監視対象の候補として抽出される．これにより，「Web ページの誤判定が増加するという課題」がある．

「Web クローラの技術」については，インターネット上の情報を収集する Web クローラの研究 [19], [20], [21], [22], [23] がある．これらの研究は，Web ページや Web サイト間のつながり情報を用いてインターネット上を自動巡回し，様々な情報を収集する．この研究をネットパトロールに用いることで，携帯ゲームサイトや SNS などのコミュニティサイトにおけるユーザを個々に識別し，ユーザのサイト上での動向や友人関係などを詳細に把握できるためネットパトロールの高度化が可能となる．ネットパトロールに Web クローラを用いた研究 [23] では，コミュニティサイト内での個人領域を特定しメインコンテンツ部に含まれるリンク関係から友人関係を抽出する技術を提案している．しかし，事前に登録された Web サイトのみを対象としているため，新出のコミュニティサイトが普及した際には，そのサイトに合わせた Web クローラを構築する必要があり，「多様なフォーマットの Web サイトに対応することが難しいという課題」がある．そのため，実際の運用に即した利活用が難しい状況である．

「記事分割の技術」については，Web ページ内のコンテンツを分割する技術 [24], [25], [26], [27], [28], [29], [30], [31] がある．しかし，これらの研究は，Web ページを分割することを目的とした研究であり，コンテンツからヘッダやフッタ，Web ページから投稿記事のみを個別に抽出することはできない．そのため，ネットパトロールを行う際に必要となる「投稿記事単位に分割されたデータセットを作成することが難しいという課題」がある．

そこで，これらの研究成果の課題を解決するために次の 2 つの技術を提案する．1 つ目は，「多様なフォーマットの Web ページからメインコンテンツを特定する技術」である．本技術を実現することで，「Web ページの有害度を判定する技術」において，既存研究の問題点を解決し，ヘッダ，フッタ，メニューなどの Web サイト共通部分により生じる誤判定を軽減させることが可能であると考えられる．また，「Web クローラの技術」においては，ネットパトロールに Web クローラを用いた研究 [23] の課題を解決し，多様な Web サイトに対応することが可能となると考えられる．2 つ目は，メインコンテンツを投稿記事単位に分割す

る技術である．本技術を実現することで，「記事分割の技術」において，現在では人手で作成している投稿記事単位に分割されたデータセットをより効率的に作成できると考えられる．

本研究では，様々な研究成果をネットパトロールに適用するために必要となる「多様なフォーマットの Web ページからメインコンテンツを推定する技術」および，ネットパトロールの高度化に必要となる「メインコンテンツを投稿記事単位に分割する技術」を開発し，ネットパトロールを支援することを目指す．

2. 関連研究

メインコンテンツを投稿記事単位に分割する技術の関連研究として，Web ページ分割の手法が提案されている．Web ページ分割の手法は，大別して「Web ページ内のテキストの内容に基づき分割する手法 [24], [25], [26]」と「Web ページを見た目に基づき分割する手法 [27], [28], [29], [30], [31]」がある．「Web ページ内のテキストの内容に基づき分割する手法」では，テキストセグメンテーション技術などを用いて Web ページの文章を内容単位にグループ化することで，Web ページを分割している．しかし，一般的に文章に着目して Web ページを分割するため，単語や画像のみで構成される部分やメインコンテンツの文章が少なかった場合は，正しく分割できないと考えられる．電子掲示板やコミュニティサイトなどでは，あいさつのみなど非常に短い投稿記事も散見されるため，Web ページ内のテキストの内容に基づき分割する手法を用いることは難しいと考えられる．一方，「Web ページを見た目に基づき分割する手法」は，Web ページのレイアウトの包含関係に基づき分割する手法 [29], [31] や，DOM（Document Object Model）構造に基づき分割する手法 [27], [28], [30] がある．

これらの手法は，Web ページの各 HTML 要素の画面上での表示位置，HTML 要素の表示位置での包含関係と DOM 構造の各要素間の親子関係を用いて Web ページを分割する．そのため，ヘッダ，フッタ，メニュー，メインコンテンツ，メインコンテンツ内の記事，画像，広告など，様々なブロックに分割することができる．これらのことから，本研究では，「Web ページを見た目に基づき分割する手法」と同様に，DOM 構造に基づきメインコンテンツを投稿記事単位に分割する手法の実現を目指す．しかし，現在取り組まれている「Web ページを見た目に基づき分割する手法」は，Web ページを分割することを目的とした研究であり，Web ページから投稿記事のみを個別に抽出することはできない．そのため，Web ページ中の投稿記事が複数のブロックに分割されるという問題や 1 つのブロックに複数の投稿記事が含まれるという問題が発生するなどの課題がみられる．

3. 研究概要

3.1 本研究における提案

人手によるネットパトロールの流れと本研究の位置づけを**図 1** に示す．1 章でも述べたとおり，ネットパトロールでは，有害な Web ページを発見するだけでなく，有害な記事が含まれている Web ページから投稿記事ごとのデータセットを作成することで，個人領域や友人関係の把握などの見守り活動を行っている．本研究では，これらのネットパトロールを効率的に支援する中で，次にあげる課題の解決を目指す．

- Web サイトを対象としたキーワード検索の作業を自動化する際に，ヘッダやフッタ，広告などに含まれる情報から誤判定が増加するという課題
- Web サイトから有害な Web ページを取得する作業を自動化する際に，多様なフォーマットに対応できないという課題
- Web ページを投稿記事単位に分類しデータセットを作成する作業の自動化ができていないという課題

これらの課題を解決するために，本研究では，「多様なフォーマットの Web ページからメインコンテンツを推定する技術」と「メインコンテンツを投稿記事単位に分割する技術」とを開発する．なお，本研究で対象とする Web サイトは，一般的にネットパトロールの巡回先である電子掲示板や SNS などのコミュニティサイトとする．これらの Web サイトの中には，PC での閲覧を前提としたもの以外にも，携帯電話やスマートフォンといったモバイル端末による閲覧を前提としている Web ページが存在する．また，これらの Web サイトは，利用者が投稿して機械的にページが生成・更新される．そのため，生成・更新されるページは，同一テンプレートに従って作成されるという特徴がある．

3.2 本研究における課題と対応方法

本研究では，「多様なフォーマットの Web ページからメインコンテンツを推定する技術」と「メインコンテンツを投稿記事単位に分割する技術」とを開発するために，それぞれ，Web ページのメインコンテンツ推定時の課題とメインコンテンツを投稿記事単位に分割する際の課題とを解消する必要がある．これらの課題への対応方策を検討するため，まず，Web ページの特性を調査する．そして，調査結果に基づきそれぞれの課題への対応方策を提案する．

3.2.1 Web ページの特徴の調査

（1）調査内容

本調査では，多様なフォーマットの Web ページからメインコンテンツを推定するため，メインコンテンツの特徴や Web ページ間におけるメインコンテンツの同一性を調査する．本調査の手順を次に示す．

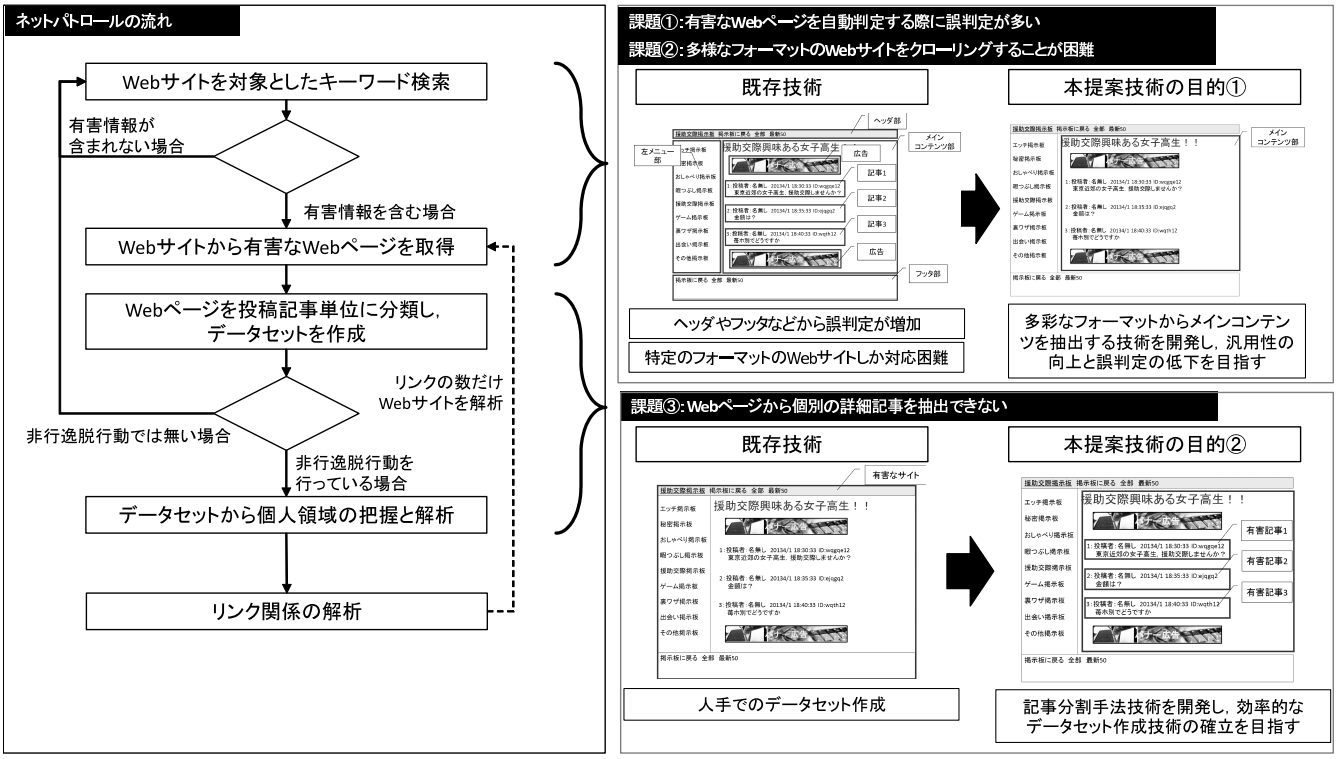

図 1 人手によるネットパトロールにおける本研究の役割
Fig. 1 Role of this research in manual Net Patrol.

表 1 Web ページの特徴の調査結果
Table 1 Survey results of character of Web pages.

項目	平均割合
ヘッダ	8.8%
右メニュー	1.8%
左メニュー	1.4%
フッタ	9.1%
メインコンテンツ	78.8%

STEP 1：任意のキーワードで Google 掲示板検索を行った結果得られた Web ページのうち投稿記事が表示される Web ページ 30 件を収集する．

STEP 2：収集した各 Web ページに対して，Web ページ全体のテキスト長に占めるメインコンテンツのテキスト長の割合を算出する．

STEP 3：収集した各 Web ページに対して，ドメイン名とディレクトリ名が同一でファイル名のみ異なる URL を持つ Web ページやファイル名より後方部分の URL のみ異なる Web ページを 4 件ずつ収集する．

STEP 4：STEP 3 で収集した各 Web ページ間において，ヘッダ，フッタやメニューなどの同一性を目視により確認する．

（2）調査結果

本調査の結果を**表 1** に示す．表 1 は，STEP 2 で算出した 30 件の Web ページのヘッダ，フッタ，左右メニュー，メインコンテンツに含まれるテキスト長の割合の平均である．表 1 の調査結果を確認すると，メインコンテンツに含まれるテキスト長が最長であることが分かる．このことから，「Web ページ内でメインコンテンツ部の内容が最も多いという特徴」があることが分かった．また，STEP 4 でヘッダ，フッタやメニューなどの同一性を目視により確認した結果，30 件中 25 件の Web ページにおいてヘッダ，フッタやメニューなどのフォーマットが共通しており，メインコンテンツに含まれる投稿内容のみが異なっていることが分かった．残りの 5 件の詳細を確認すると，URL のファイル名以降にその電子掲示板の ID が付加された Web ページとなっており，それぞれが別々の電子掲示板サイトの Web ページであることが分かった．そのため，電子掲示板の ID を残して再度調査を行ったところ，これらの Web ページもヘッダ，フッタやメニューなどのフォーマットが共通していることが分かった．このことから，本研究で対象とする Web ページには，「同一ドメイン内の同一フォーマットの Web ページのデザインはヘッダ，フッタやメニューなどが共通するという特徴」と「投稿内容などが含まれるメインコンテンツ部は Web ページごとで異なるという特徴」とがあることが分かった．

3.2.2 Web ページのメインコンテンツ推定時の課題への対応方法

Web ページのメインコンテンツは，1 つの Web ページを確認しただけでは，ヘッダ，フッタ，メニュー，メインコンテンツなどの違いを判別できず，一意に特定することができない状況である．そのため，本研究では，「同一ドメイ

ン内の同一フォーマットの Web ページのデザインはヘッダ，フッタやメニューなどが共通するという特徴」，「投稿内容などが含まれるメインコンテンツ部は Web ページごとで異なるという特徴」と「Web ページ内でメインコンテンツ部の内容が最も多いという特徴」に着目し，複数の同一フォーマットの Web ページを対象として HTML ソースの同一性を解析することで，メインコンテンツを推定する．

3.2.3 メインコンテンツを投稿記事単位に分割する際の課題への対応方法

本研究では，「Web ページを見た目に基づき分割する手法」と同様に，DOM 構造を用いる手法を改良してメインコンテンツを投稿記事単位へ分割する．しかし，これらの手法を用いた場合，Web ページ中の投稿記事が複数のブロックに分割されるという問題や 1 つのブロックに複数の投稿記事が含まれるという問題が発生する．そのため，本研究では，「掲示板，SNS やブログなどの CGM の Web ページのメインコンテンツは，1 つの Web ページ内に複数の投稿記事が含まれるため，一定間隔で同様の HTML 要素が繰り返し出現するという特徴」に着目し，メインコンテンツを一定間隔で繰り返される HTML 要素のパターンを自動的に検出し，グループ化することで対応する．また，PC での閲覧を前提とした Web ページでは，大量の HTML タグを利用して PC で最適な表示がなされるよう工夫されている場合が多いが，モバイル端末向けの Web ページではデータ量を減らすなどの目的から投稿を <hr> タグで区切っただけで余計な HTML タグを利用しない簡素な Web ページが多く存在する．既存手法を用いた場合，これらの Web ページは，HTML タグを Web ページ分割時の区切り候補としているため，HTML タグが Web ページ内にない場合には正確に分割できない．そこで，本提案手法が，この課題に対しても有効な Web ページであり，モバイル端末向けの Web ページにおいても分割できる手法であることを示すため，6 章の実験では，「HR 有」と「HR 無」の区別を行った実証実験を行う．

3.3 処理の流れ

本研究の処理の流れを**図 2** に示す．本研究では，解析対象の Web ページの URL と複数の同一ドメインの Web ページの URL に基づき取得した HTML ソースを解析してメインコンテンツ要素を推定する機能とメインコンテンツの解析結果と解析対象 Web ページから投稿記事を抽出する記事抽出機能とで Web ページを分割し，投稿記事を抽出する．なお，本研究では，Web ページのレイアウトに利用しないタグである <!-- --> タグ，<a> タグ，<script> タグ，
 タグ，<input> タグ，<style> タグを処理対象から除外する．

メインコンテンツ要素推定機能は，DOM 構造構築処理とメインコンテンツ要素推定処理で構成される．DOM 構造構築処理では，入力された複数の Web ページの URL の HTML ソースを取得し，DOM 構造を構築する．この際，多様なフォーマットの Web ページに対応するため，テキスト部分に <text> タグを付与し，DOM 構造に追加する．メインコンテンツ要素推定処理では，複数の Web ページの DOM 構造の類似性に基づき，メインコンテンツを表す HTML 要素を抽出し，Web ページ解析結果 DB に格納する．メインコンテンツ要素推定機能の詳細は，4 章で解説する．

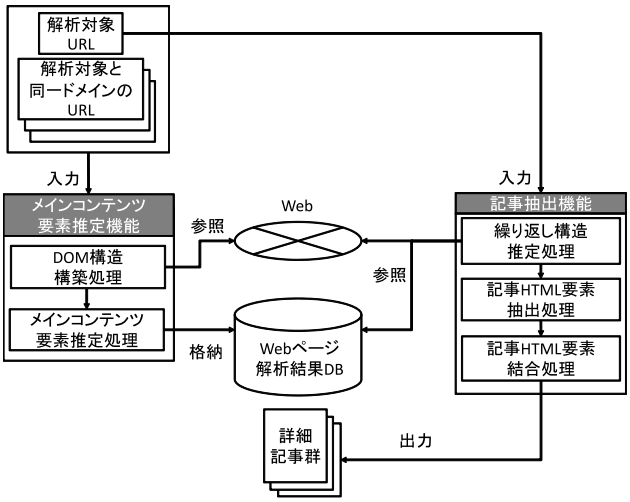

図 2 処理の流れ
Fig. 2 Flow of process.

記事抽出機能は，繰返し構造推定処理，記事 HTML 要素抽出処理と記事 HTML 要素結合処理で構成される．繰返し構造推定処理では，まず，解析対象 Web ページの URL に基づき HTML ソースを取得する．次に，1 つの投稿記事を表現する HTML 要素の集合を推定するため，一定間隔で繰り返される HTML 要素のパターンを DOM 構造の階層ごとに抽出する．そして，階層ごとに抽出した投稿記事の HTML 要素の繰返しパターンの中で，投稿記事を表現する HTML 要素の集合として適切なものを推定する．記事 HTML 要素抽出処理では，HTML 要素の繰返しパターンに基づき，解析対象 Web ページの DOM 構造から HTML 要素を抽出する．記事 HTML 要素結合処理では，まず，各投稿記事を表現する HTML 要素の集合が，実際の各投稿記事を適切に抽出できているかを確認する．次に，投稿記事の一部分（たとえば，投稿記事の投稿日時のみなど）しか抽出できていない場合は，同一階層に含まれる兄弟 HTML 要素との関係を解析する．最後に，同一階層に含まれる兄弟 HTML 要素が同一の投稿記事に含まれる場合は，HTML 要素を結合し，1 つの投稿記事として取得する．記事抽出機能の詳細は，5 章で解説する．

4. メインコンテンツ要素推定機能

本研究では，Web ページ分割を行うための前処理として，

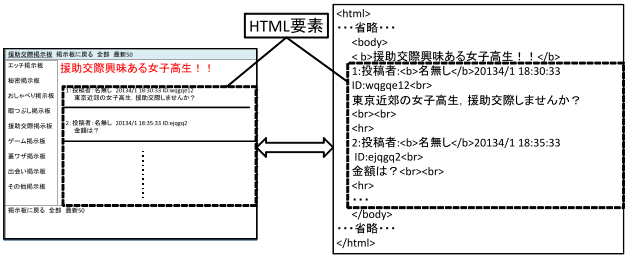

図 3 <hr> タグと
 タグのみで構成される Web ページ
Fig. 3 Web page composed of '<hr>' and '
' tag.

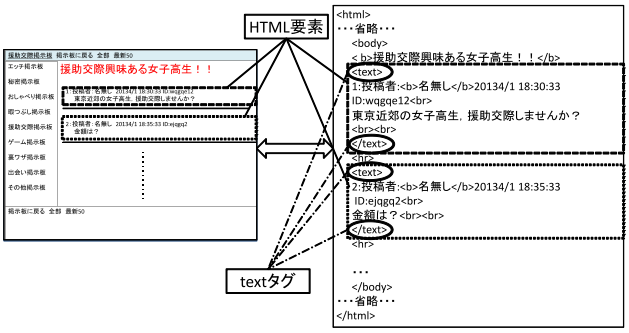

図 4 <text> タグを挿入した後の Web ページ
Fig. 4 Web page of after inserting '<text>' tag.

Web ページの中で投稿記事部分を含む HTML 要素であるメインコンテンツ要素の推定を行う．本研究で対象とする電子掲示板などの Web ページは，「同一ドメイン内の同一フォーマットの Web ページのデザインはヘッダ，フッタ，やメニューなどが共通するという特徴」と「投稿内容などが含まれるメインコンテンツ部は Web ページごとで異なるという特徴」がある．そのため本機能では，同一フォーマットの Web ページにおける Web ページの差異を特定することで Web ページ内のメインコンテンツ要素を推定する．複数の Web ページの差異からメインコンテンツ要素を推定するためには，それぞれの Web ページにおける DOM 構造の共通性を確認する必要がある．しかし，DOM 構造を対象に処理を行った場合，DOM 構造に含まれる各 HTML 要素が処理の最小単位となる．そのため，**図 3** に示す <hr> タグと
 タグで投稿記事が区切られた Web ページなどでは，任意の HTML 要素の子要素に属していないテキスト部分を処理することができず，HTML 要素として DOM 構造の共通性が確認できない．そこで，本研究では，事前処理として Web ページに含まれる単体で取得できないテキスト部分に対し，<text> タグを付与して DOM 構造を構築（**図 4**）する．これにより，Web ページに記述されているすべての内容を考慮した処理が可能であると考えられる．

4.1 DOM 構造構築処理

本処理では，入力された URL に対応する Web ページの任意の親要素のどの子要素にも属していないテキスト部分

Algorithm 1 DOM 構築アルゴリズム
Require:
//HTML 要素 n の HTML ソースを取得
$InnerHtml(n)$
//HTML 要素 n の子要素を取得
$ChildElements(n)$
//入力した HTML 文字列から DOM を再構築
$DomReconstruction(html)$
Ensure:
Function $InsertTextElement(element)$
//入力された HTML 要素に含まれる全体の HTML ソースを取得
$text_{parent} := InnerHtml(element)$
//入力された HTML 要素の子要素の HTML ソース配列を構築
$text_{children} := \{\}$
For Each $childelement$ **in** $ChildElements(element)$ **Do**
$text_{children}.append(InnerHtml(element))$
End For
//全体の HTML ソースから子要素の HTML ソースを除去し
//差分となる HTML ソース配列の取得
$differences := text_{parent} - text_{children}$
//すべての差分文字列に <text> 要素を付与した HTML ソース配列を構築
$differences' := \{\}$
For Each $difference$ **in** $differences$ **Do**
$differences'.append(\text{"<text>"} + difference + \text{"</text>"})$
End For
//構築した <text> 要素を付与した HTML ソース配列と
//子要素の HTML ソースを結合し新しい全体の HTML ソースを構築
$text'_{parent} := text_{children} + differences'$
//新たな HTML 要素 $element'$ を構築し，$element'$ に含まれる子要素に対し処理を実行
$element' := DomReconstruction(textparent')$
For Each $childelement'$ **in** $ChildElements(element')$ **Do**
$childelement' := InsertTextElement(childelement')$
End For
//<text> 要素を付与した HTML 要素を返却
Return $element'$
End Function

に対して <text> タグを付与した DOM 構造を構築する．本処理を事前に行うことで，通常の DOM 構造では処理できなかったテキストを HTML 要素として処理することが可能である．本処理の詳細を **Algorithm 1** に示す．本アルゴリズムは，入力された Web ページ $Page_i$ の見た目を構成する <body> タグ以下の各 HTML 要素に含まれるテキスト部分を特定し，特定したそれぞれのテキスト部分に対して <text> タグを挿入する処理である．本処理では，次に示す 6 つのステップを順次実行することにより，テキスト要素を挿入する．

STEP 1：処理を行う最初の HTML 要素として，Web ページ $Page_i$ に含まれる <body> タグを入力する．

STEP 2：入力した HTML 要素に含まれるすべての HTML ソース $text_{parent}$ とその要素のすべての子要素が保持する HTML ソース配列 $text_{children}$ を抽出する．

STEP 3：STEP 1 で抽出した $text_{parent}$ の中で $text_{children}$ に一致する箇所を探索し，一致していないすべての箇所に対して <text> タグを挿入する．

STEP 4：STEP 2 において，<text> タグの要素内に HTML 要素が存在する場合，HTML 要素の前後で <text> タグを分断する．

STEP 5：$text_{children}$ と <text> タグを挿入した HTML ソースを元の状態に戻し，DOM 構造を構築する．

STEP 6：STEP 2 で入力された HTML 要素に含まれる

<text> タグ以外の子要素が存在する場合，<text> タグ以外のすべての子要素を入力として STEP 2〜STEP 6 の処理を実行する．<text> タグ以外の子要素が存在しない場合，<text> タグが挿入された Web ページを $Page'_i$ として処理を終了する．

4.2 メインコンテンツ要素推定処理

本処理では，DOM 構造構築処理で <text> タグを挿入した $Page'_i$ からメインコンテンツ要素 $element_{main}$ を推定する．本研究で対象とする Web ページには，「同一ドメイン内の同一フォーマットの Web ページのデザインはヘッダ，フッタ，メニューなどが共通するという特徴」と「投稿内容などが含まれるメインコンテンツ部は Web ページごとで異なるという特徴」がある．これらの特徴から，**図 5** に示すとおり，メインコンテンツ要素のみが，そのページに投稿された投稿内容や投稿件数ごとに子要素数が異なり，それ以外の部分は同一になると考えられる．そのため，本処理では，複数の Web ページ $Pages' = \{Page'_1, Page'_2, \ldots, Page'_i\}$ を対象として，各 HTML 要素が持つ子要素数の差からメインコンテンツ要素を推定する．本処理の詳細を **Algorithm 2** に示す．本アルゴリズムは，DOM 構造の中で画面上の表示内容を構成する最上位要素である <body> タグを入力し，各階層において再帰的に処理を行うことによりメインコンテンツ要素を推定する一連の処理である．ここで，本処理における階層 k とは，図 5 に示す DOM 構造における <body> タグからの深さである．また，階層 k の位置にある HTML 要素は，任意の $Page_i$ の階層 k に含まれるすべての HTML 要素 $element_{ikj}$ を指す．本処理では次に示す 5 つのステップを順次実行することによりメインコンテンツ要素を推定する．

STEP 1：階層 $k = 1$ の位置にある HTML 要素として，$Page'_1$ から $Page'_i$ までの各 Web ページに含まれる <body> 要素を入力する．

STEP 2：$Page'_i$ の階層 k に存在する HTML 要素集合のすべての子要素数 $childrencount_i$ を算出し，子要素数の一致率を算出する．

STEP 3：子要素数の一致率が閾値 β を上回っている場合，**図 6** に示す入力されたそれぞれの HTML 要素の子要素集合を次の処理対象とし，STEP 2〜STEP 3 を実行する．子要素数の一致率が閾値 β を下回った時

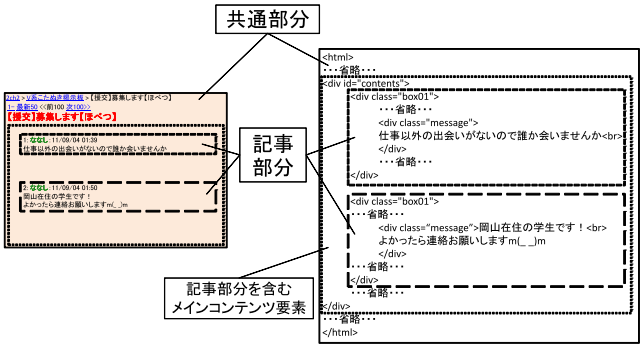

図 5 メインコンテンツと投稿記事の関係

Fig. 5 Relationship between main contents and posting articles.

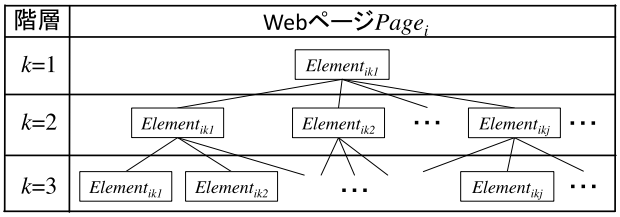

図 6 Web ページにおける DOM 構造の階層

Fig. 6 Layer of DOM tree in Web page.

Algorithm 2 メインコンテンツ要素推定アルゴリズム

Require:
- //HTML 要素 n の DOM 構造に <text> タグを挿入（Algorithm 1）
- $InsertTextElement(n)$
- //任意の $Page_i$ に含まれるすべて HTML 要素の中で最深の HTML 要素の深さを取得
- $MaxElementDepth(Page_i)$
- //配列 $array$ に含まれる要素数を取得
- $Count(array)$
- //配列 $array$ に含まれる要素から重複を削除した配列を取得
- $Distinct(array)$
- //任意の $Page_i$ の階層 k にあるすべての HTML 要素集合を取得
- $LevelElements(Page_i, k)$
- //HTML 要素 n に含まれるテキスト長を取得
- $TextLength(n)$

Ensure:
- **Function** $ExtractMainElement(Pages)$
  - //初期化
  - $k := 1$
  - //$Pages'$ の中で最深の階層 k_{max} を算出
  - $k_{max} := 0$
  - **For Each** $Page'_i$ **in** $Pages'$ **Do**
    - **If** $k_{max} < MaxElementDepth(Page'_i)$ **Then**
      - $k_{max} := MaxElementDepth(Page'_i)$
    - **End If**
  - **End For**
  - //各 $Page'_i$ における階層 k の要素数からメインコンテンツを含む階層を推定
  - **While** $k < k_{max}$ **Do**
    - //$Page'_i$ の階層 k の子要素数を算出し，$childrencount_i$ を構築
    - $childrencount := \{\}$
    - **For Each** $Page'_i$ **in** $Pages'$ **Do**
      - $childrencount_i = Count(ChildrenElements(LevelElements(Page'_i, k)))$
    - **End For**
    - //子要素の一致率が β を下回った場合，その時点の階層 k を k_{main} として定義
    - **If** $Count(Distinct(childrencount))/Count(childrencount) < \beta$ **Then**
      - $k_{main} := k$
      - **Break**
    - **End If**
    - //k を加算して次の階層で再度処理を実行する
    - $k := k + 1$
  - **End While**
  - //最終階層 k_{max} に達するまで k_{main} が決まらなかった場合
  - //k_{main}=1 として処理を続行する
  - **If** $k = k_{max}$ **Then**
    - $k_{main} := 1$
  - **End If**
  - //$Page'_i$ の階層 k_{main} にある各 HTML 要素間においてテキスト長が最長となる
  - //HTML 要素 $element_{ikj}$ を探索
  - $textlength_{kj} := \{\}$
  - **For Each** $Page_i$ **in** $Pages$ **Do**
    - **For Each** $element_{ikj}$ **in** $LevelElements(Page_i, k_{main})$ **Do**
      - $textlength_{kj} := textlength_{kj} + TextLength(element_{ikj})$
    - **End For**
  - **End For**
- **End Function**

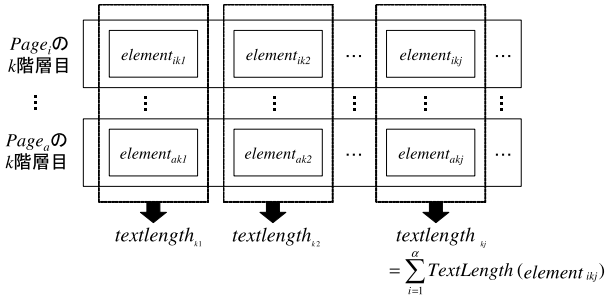

図 7 テキスト長の算出方法
Fig. 7 Calculational procedure of text length.

点の階層 k をメインコンテンツ要素のある階層として STEP 4 を実行する．また，すべての $Page_i$ の中で最深の階層である k_{max} に達した場合は，最上位の階層を設定する．

STEP 4：$Page'_i$，階層 k におけるすべての HTML 要素 $element_{ikj}$ ごとのテキスト長を**図 7** に示すとおり加算した $textlengh_{kj}$ を算出する．

STEP 5：最大の $textlength_{kj}$ を持つ $element_{ikj}$ をメインコンテンツ要素として処理を終了する．このとき，最大の要素が複数存在する場合は，その中で j が最大の $element_{ikj}$ をメインコンテンツ要素とする．

5. 記事抽出機能

本研究では，メインコンテンツ要素推定機能で推定したメインコンテンツ要素以下の HTML 要素を対象として，記事単位に分割する手法を考案し，記事の抽出を行う．本機能では，Web ページ内に含まれる記事部分の HTML 要素が繰返し構造となることに着目し，この繰返し構造を推定することで記事を正確に分割し，抽出する手法を提案する．本手法を用いることにより，多様なフォーマットの Web ページから記事を抽出できると考えられる．

5.1 繰返し構造推定処理

本処理では，メインコンテンツ要素推定処理で推定したメインコンテンツ要素に含まれる各記事を抽出する．メインコンテンツ要素に含まれる記事には，単一または複数の HTML 要素が繰り返し出現しているという特徴がある．そのため，本研究では，繰返し構造のパターンを自動的に推定することにより，記事を抽出する手法を提案する．本研究では，LZW（Lempel-Ziv-Welch）アルゴリズム [32] を用いてこれらの構造を自動的に推定する汎用的な手法を提案する．LZW アルゴリズムとは，データ圧縮技術に用いられるアルゴリズムの一種であり，データに含まれる最長のパターンを短い符号に置換することでデータを圧縮するアルゴリズムである．本研究では，LZW アルゴリズムの最長のパターンを発見する手法を拡張し HTML 要素集合に適用することで，繰返し構造のパターンを推定する．本研究で用いる LZW アルゴリズムを拡張した繰返し構造推定アルゴリズムを **Algorithm 3** に示す．本アルゴリズムは，LZW アルゴリズムを拡張することにより HTML 要素集合から構築した XPath 配列に対して適用するアルゴリズムである．本処理では次に示す 4 つのステップを順次実行することにより繰返し構造を推定する．

Algorithm 3 繰返し構造推定アルゴリズム

```
Require:
  //HTML 要素集合 narray に含まれる各要素の子要素の集合
  ChildrenElements(n_array)
  //HTML 要素 n の XPath を取得
  XPath(n)
  //XPath を符号化
  Coding(xpath)
Ensure:
  Function EstimateArticleStructure(element_main)
    //初期化
    elements := ChildElements(element_main)
    xpaths, xpaths_article := {}
    //処理対象ページのメインコンテンツ要素より下の階層に位置する
    //HTML 要素に対して階層順に LZW 法を適用し，各階層において
    //最長となる XPath 配列のパターンを算出
    While Count(elements) > 0 Do
      xpaths := LZWExtendsElements(elements)
      //繰返し回数が最多となる XPath 配列を抽出
      If PatternLength(element_main, xpaths) >
          PatternLength(element_main, xpaths_article) Then
        xpaths_article := xpaths
      End If
    End While
  End Function

  Function LZWExtendsElements(elements)
    //HTML 要素集合 elements に含まれる HTML 要素を XPath ごとに符号化を
    //行い，符号化した文字と XPath のペア，符号文字列を構築
    encoding := {}
    encoded := ""
    For Each element in elements Do
      xpath := XPath(element)
      If Not ContainsValue(encoding, xpath) Then
        encoding[Coding(xpath)] := xpath
      End If
      encoded := encoded + encoding.ValueOf(element)
    End For
    //符号化した XPath を LZW アルゴリズムを用いてさらに符号化
    code := ""
    table := {}
    num := 0
    p := encoded[num]
    While num < Count(encoded) Do
      c := encoded[num + 1]
      If Contains(table, p + c) Then
        p := p + c
      Else
        code := code + p
        table.append(p + c)
        p := c
      End If
    End While
    code := code + p
    //最も登場回数の多い符号を取得し，符号化された XPath を複合化
    code' := ValueOfUpTo(code)
    xpaths = {}
    For Each sign in code' Do
      xpaths.append(encoding[table[sign]])
    End For
    //構築した xpaths を返却
    Return xpaths
  End Function
```

STEP 1：メインコンテンツ要素推定処理により推定したメインコンテンツ要素の子要素集合から XPath 配列を構築する．

STEP 2：STEP 1 で構築した XPath 配列の先頭から順に XPath を符号化し，符号配列を構築する．

STEP 3：STEP 2 で構築した符号配列を LZW アルゴリズムに入力し，最長となる符号のパターンとその繰返し回数を取得する．

STEP 4：現在処理した子要素集合の子要素が 1 件以上存在する場合，現在処理した子要素集合の子要素集合を対象として STEP 1〜STEP 4 を実行する．最下層 k_{max} に位置する要素のみとなった場合，繰返し回数の最も多かった符号のパターンを複合化し，記事を構成する要素の XPath 配列 $xpaths_{article}$ を繰返し構造のパターンとして抽出する．

5.2 記事 HTML 要素抽出処理

本処理では，繰返し構造推定処理により抽出した XPath 配列 $xpaths_{article}$ に基づいて，メインコンテンツ要素以下の HTML 要素を取捨選択することにより，HTML 要素を投稿記事単位に分割する．XPath に基づき HTML 要素を抽出するアルゴリズムを **Algorithm 4** に示す．本アルゴリズムは，入力されたメインコンテンツ要素 $element_{main}$ と XPath 配列 $xpaths_{article}$ から HTML 要素を取捨選択することにより記事 HTML 要素集合 articles を抽出する処理である．本処理では次に示すステップを順次実行することにより投稿記事を構成する HTML 要素を抽出する．

STEP 1：繰返し構造推定処理で推定した XPath 配列 $xpaths_{article}$ が示す HTML 要素と同一の階層にある HTML 要素 $elements_{article}$ を取得する．

STEP 2：$elements_{article}$ を親要素ごとに分割し，それぞれの分割された要素集合の中から $xpaths_{article}$ と同一の並びとなる箇所を探索し抽出する．

STEP 3：$xpaths_{article}$ と同一の並びとなる箇所を発見するたびに記事 $article_n$ として抽出する．

STEP 4：STEP 1〜STEP 3 を抽出できる要素がなくなるまで実行することで，記事配列 $articles$ を構築する．$articles$ に含まれる投稿記事数が 1 件以上の場合はここで処理を終了し，次の処理を実行する．

STEP 5：STEP 4 において抽出できた記事件数が 0 件の場合，STEP 2 で親要素ごとに分割する前の $elements_{article}$ に含まれるすべての要素を対象として STEP 3〜STEP 4 を実行し，$articles$ を構築する．

5.3 記事 HTML 要素結合処理

本処理では，細分化された各投稿記事の HTML 要素集合を親要素に基づいて結合する．各投稿記事に相当する HTML 要素は，図 7 に示すとおり任意の 1 つの親要素のもとで繰り返すという特徴がある．そのため，本研究では，この特徴に着目し，記事 HTML 要素抽出処理で抽出した各 HTML 要素の親要素が 1 つとなる階層まで遡ることにより最終的な記事 HTML 要素集合を特定する．本処理を行うことにより，繰返し構造の抽出時に漏れてしまった HTML 要素を記事 HTML 要素集合に含めることが可能となる．本処理のアルゴリズムを **Algorithm 5** に示す．本アルゴリズムでは，次に示す 2 つのステップを順次実行することにより記事を抽出する．

STEP 1：記事 HTML 要素抽出処理で抽出した HTML 要素集合 $articles$ のそれぞれの親要素集合 $parents$ を取得する．

STEP 2：$parents$ の要素数が 1 件より多く，$articles$ の要素数が $parents$ の倍数になっている場合，親階層を記事 HTML 要素集合と見なして，再度 STEP 1〜STEP 2 を実行する．それ以外の場合は現在の記事 HTML 要素集合を記事 HTML 要素集合 $articles$ として返却する．

Algorithm 4 記事 HTML 要素抽出アルゴリズム

```
Require:
  //HTML 要素 n の子要素集合から XPath と同じ深度にある HTML 要素集合を取得
  TakeElementFormXPath(xpath, n)
  //HTML 要素集合 n_array の各要素の親要素集合を取得
  ParentElements(n_array)
Ensure:
  Function ExtractArticleElements(xpaths_article, element_main)
    //初期化
    elements_article := TakeElementFormXPath(xpaths_article, element_main)
    articles := {}
    //親要素ごとの子要素集合から XPath と一致する要素集合を抽出
    For Each parent in ParentElements(elements_article) Do
      article := {}
      xpathindex := 0
      text_children.append(InnerText(element))
      For Each element in ChildElements(parent) Do
        If XPath(element) = xpaths_article[xpathindex] Then
          article.append(element)
          xpathindex := xpathindex + 1
        End If
      End For
      If Count(article) = Count(xpaths_article) Then
        articles.append(article)
      End If
    End For
    //親を考慮する手法で記事を一件も取得できなかった場合は
    //親要素の制限を解除して要素を取得
    If Count(articles) = 0 Then
      article := {}
      xpathindex := 0
      For Each element in elements_article Do
        If XPath(element) = xpaths_article[xpathindex] Then
          article.append(element)
          xpathindex := xpathindex + 1
        End If
        If Count(article) = Count(xpaths_article) Then
          articles.append(article)
          article := {}
          xpathindex := 0
        End If
      End For
    End If
    //抽出した記事 HTML 要素集合を返却
    Return articles
  End Function
```

Algorithm 5 投稿記事 HTML 要素結合アルゴリズム

Require:
//HTML 要素集合 n_{array} の各要素の親要素集合を取得
$ParentElements(n_{array})$
Ensure:
Function $ArticleJoin(articles)$
//記事 HTML 要素集合 $articles$ に含まれる
//HTML 要素のすべての親要素集合を取得
$parents := ParentElements(articles)$
If $Count(parents) > 1$ *And* $Count(articles)$ % $Count(parents) = 0$ **Then**
Return $ArticleJoin(parents)$
Else
Return $articles$
End If
End Function

6. 評価実験

6.1 実験計画

本研究で提案した Web ページ分割手法の有用性を検証するため，実際の Web サイトを模して生成したシミュレーションデータを用いて「メインコンテンツ要素の推定精度」と「Web ページ分割精度」の 2 項目について評価実験を行う．その後，他の分割手法との比較と異なるデータでの Web ページ分割とを行い，提案手法の有用性を評価する．本研究の実験計画を**図 8** に示す．図 8 は，評価実験により検証する項目を明確化するため，図 2 の実験内容との対応関係を図に示したものである．実験 1 では，多様なフォーマットの Web ページからメインコンテンツを特定できないという課題を解決できているかを評価するため，人工的に生成した多様なフォーマットの Web ページを用いてメインコンテンツ要素の推定を行い，その結果を考察する．実験 2 では，メインコンテンツを投稿記事単位に分割できないという課題を解決できているかを評価するため，人工的に生成した Web ページのメインコンテンツを用いて既存の Web ページ分割手法を用いた投稿記事抽出手法との比較実験を行い，その結果を考察する．実験 3 では，提案手法の有用性を評価するため，3 種類のデータに対して提案手法を適用して投稿記事の抽出を行い，その結果から提案手法の有用性を評価する．なお，本実験では，従来の手法で対応が困難であった携帯電話向けサイトで多くみられる <hr> タグと
 タグのみでデザインされた Web ページにおいても本提案手法が適用可能であるかを確認するため，すべての実験においてこれらの Web ページと PC 向けのサイトとして様々なタグを用いてデザインされた Web ページとの 2 つに分けて分析を行う．

6.2 実験データの準備

評価実験では，提案手法の精度を評価するため，実際の Web ページを収集したデータセット（以下，「実データセット」）と実際の Web ページを模して人工的に生成したデータセット（以下，「人工データセット」）とを用意する．各データの生成方法を次に示す．

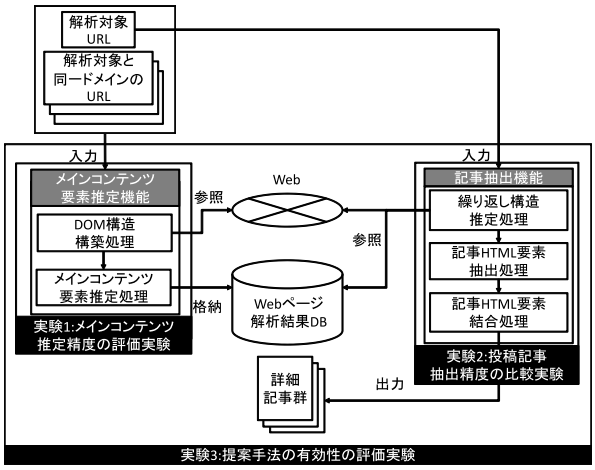

図 8 実験計画
Fig. 8 Plan of experiment.

6.2.1 実データセットの作成手順

実データセットは，人工データセットを実際の Web ページに模して生成するためのデータとして用いる．そのため，実データセットは，実際のネットパトロールで対象となる Web ページを収集する．実データセットの作成手順を次に示す．

STEP 1：隠語・有害語データベース [33] のカテゴリ別用語アクセスランキング Top5 の語句を取得する．

STEP 2：STEP 1 で取得した語句で Google 掲示板検索を行い，上位 200 件の Web ページを取得する．

STEP 3：STEP 2 で取得した Web ページを目視で確認し，有害と判断した場合に実データセットに追加する．ここで，掲示板のフォーマットは，携帯電話向けサイトとして <hr> タグと
 タグのみでデザインされたページ（以下，「HR 有」）と PC 向けのサイトとして様々なタグを用いてデザインされたページ（以下，「HR 無」）とに大別される．この処理をデータセットの件数が，HR 無 50 件，HR 有 50 件になるまで繰り返し実施する．

STEP 4：収集したデータを目視で確認し，**図 9** に示すとおり Web ページの各パートを表現する HTML 要素に，ヘッダ部，フッタ部，右メニュー部，左メニュー部，メインコンテンツ部を示す情報を付与する．なお，掲示板やブログなどの CGM では，1 つのメインコンテンツ内に複数の投稿内容が含まれるため，それぞれの投稿内容を目視で分割し，投稿内容であることを示す情報を付与する．

6.2.2 人工データセットの作成手順

人工データセットは，提案手法のメインコンテンツ要素推定アルゴリズムと記事抽出アルゴリズムの有用性を評価するために用いる．人工データセットは，実データを模した Web ページを生成するため，実データセットに含まれる Web ページを用いて自動的に生成する．人工データセット

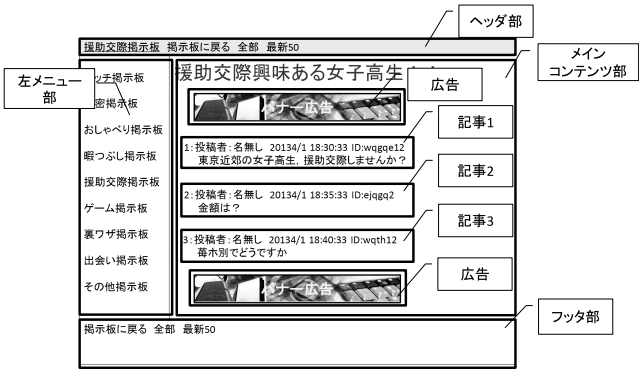

図 9 Web ページの各パートの 2 次情報の例

Fig. 9 Example of secondary information of each part in Web page.

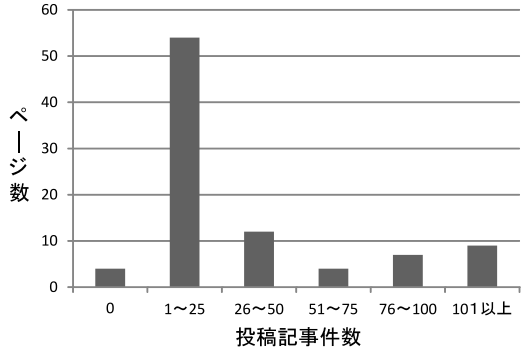

図 10 実データセットにおける投稿記事件数の分布

Fig. 10 Distribution of number of articles in real dataset.

の作成手順を次に示す．

STEP 1：実データセットに含まれる Web ページをテンプレート（ヘッダ，フッタ，メニュー）とメインコンテンツ部に分割し，人工データ生成のための HTML 要素群を用意する．

STEP 2：STEP 1 で生成した HTML 要素群を用いて人工的に Web ページを生成する．生成手順の詳細を次に示す．

STEP 2.1：実データセットからテンプレート部の元となる Web ページとメインコンテンツ部の元となる Web ページとを各 1 件ずつランダムに選択する．

STEP 2.2：実データセットに含まれる 1 ページ内の投稿件数の分布（**図 10**）に従い，ページに含まれる投稿件数をランダムに決定する．

STEP 2.3：STEP 2.1 で選択したメインコンテンツ部の Web ページの HTML 要素を用いて，STEP 2.2 で決定した投稿件数分の HTML を生成する．

STEP 2.4：STEP 2.1 で選択したテンプレート部の Web ページのメインコンテンツ部の HTML を STEP 2.3 で生成した HTML に置換する．このとき，Web ページを構成するそれぞれの HTML 要素には，それぞれの部位（ヘッダ部やメインコンテンツ部など）であることを示す情報が付与されているものとする．

STEP 3：STEP 2 の手順を繰り返し実施し，HR 無 500 件，HR 有 500 件の人工データが生成されるまで繰り返し実施する．

本手順で作成した人工データセットは，実データセットの内容を各テンプレートとメインコンテンツ部に分けたものを無作為に組み合わせているため，Web ブラウザから確認した場合にデザインが崩れている場合や同じ記事が 1 ページ内に複数回登場する場合がある．しかし，いずれの場合においても DOM 構造に問題がない場合，通常の Web ページと同様であると考えられることから，これらの人工データセットを用いて本提案手法の評価実験を行う．

6.3 実験 1：メインコンテンツ推定精度の評価実験

6.3.1 実験内容

本実験では，多様なフォーマットの Web ページからメインコンテンツを特定できないという課題を解決できているかを評価するため，人工的に生成した多様なフォーマットの Web ページを用いてメインコンテンツの推定を行い，その結果を考察する．本実験の手順を次に示す．

STEP 1：人工データセットから実験対象の Web ページを取得する．

STEP 2：STEP 1 で取得した Web ページを用いて，メインコンテンツ要素推定機能で利用する同一フォーマットの Web ページを生成する．このとき，同一フォーマットの Web ページに含まれる投稿件数は，図 10 の分布に従いランダムに決定する．

STEP 3：STEP 2 で用意した同一フォーマットの Web ページ群を用いて，メインコンテンツの推定を行い，メインコンテンツ要素を取得する．

STEP 4：STEP 3 で取得したメインコンテンツ要素と正解データである人工データセットの各 Web ページのメインコンテンツ部の HTML 要素とを比較し評価する．本実験において，正解の判定は，「条件 1：取得したメインコンテンツ要素が正解データの HTML 要素に対して一定の許容範囲内にあること」と，「条件 2：取得したメインコンテンツ要素内にすべての投稿記事が含まれること」の 2 つの条件を満たした場合とした．条件 1 において，完全一致ではなく許容範囲内とした理由は，Ajax などで動的に組み込まれる広告の HTML が表示のタイミングによって異なる事例や正解データとなるメインコンテンツ部の HTML 要素が複数候補存在する事例がみられたためである．条件 2 において，すべての投稿記事が含まれることとした理由は，許容範囲内に収まっていたとしても，投稿記事が含まれていなければネットパトロールに必要なデータセットを取得することができないと判断したためである．本実験では，許容範囲を 5%から 30%まで 5%間隔で設定し，評価結果を算出する．

表 2 パラメータ α の決定
Table 2 Decision of parameter α.

α		2	3	4	5	6	7	8	9	10
許容範囲	5%	0.420	0.400	0.450	0.360	0.430	0.390	0.420	0.470	0.450
	10%	0.530	0.510	0.560	0.470	0.650	0.540	0.510	0.550	0.560
	15%	0.640	0.580	0.680	0.550	0.690	0.610	0.610	0.620	0.660
	20%	0.760	0.650	0.750	0.640	0.720	0.700	0.720	0.720	0.720
	25%	0.780	0.700	0.780	0.680	0.750	0.760	0.760	0.770	0.730
	30%	0.810	0.760	0.820	0.750	0.790	0.790	0.820	0.820	0.810
最大回数		2	0	2	0	2	0	1	2	0
平均推定精度		0.657	0.600	0.673	0.575	0.672	0.632	0.640	0.658	0.655

STEP 5：評価結果を集計し，正解率を算出する．

6.3.2 メインコンテンツ推定精度の評価実験用パラメータの設定

メインコンテンツ推定精度の評価実験では，同一フォーマットの Web ページの URL 件数 α とテンプレート解析処理におけるメインコンテンツ階層選定の閾値 β とを用いる．各パラメータについて，次のとおり設定した．

（1）パラメータ α

パラメータ α は，メインコンテンツ要素推定機能で用いる同一フォーマットの Web ページの URL 件数を表す．本研究では，パラメータ α の値を適切に設定するため，実験で用いる人工データと同様の方法で別途生成した Web ページ 100 件を対象に，α の値を変化させてメインコンテンツの推定精度を評価した．メインコンテンツの推定精度は，正解判定の許容範囲を 5%から 30%まで 5%間隔で設定して算出する．なお，本評価では，α の値を 2 から 10 まで 1 間隔で変化させて実行した．評価結果（**表 2**）を確認すると，平均推定精度が最大となるものは，$\alpha = 4$ の一致率 0.673 であることが分かった．このことから，本実験では $\alpha = 4$ と設定する．

（2）パラメータ β

パラメータ β は，メインコンテンツ要素推定処理におけるメインコンテンツ階層を選定するための閾値であり，各ページの階層ごとの HTML 要素数を比較した際の一致率と比較する．本研究では，パラメータ β の値を適切に設定するため，β の値を変化させてメインコンテンツの推定精度を評価した．メインコンテンツの推定精度は，正解判定の許容範囲を 5%から 30%まで 5%間隔で設定して算出する．なお，本評価では，β の値を 0.1 から 1.0 まで，0.1 間隔で変化させて実行した．評価結果（**図 11**）を確認すると，すべての許容範囲の評価結果において同様の傾向がみられ，0.6 以上の場合に最も高精度であることが分かった．システム試作時に完全一致（$\beta = 1.0$）の場合には，誤判定する事例がみられたことを考慮し，本実験では $\beta = 0.6$ と設定する．

6.3.3 結果と考察

メインコンテンツ推定の評価実験の結果を**表 3** に示す．

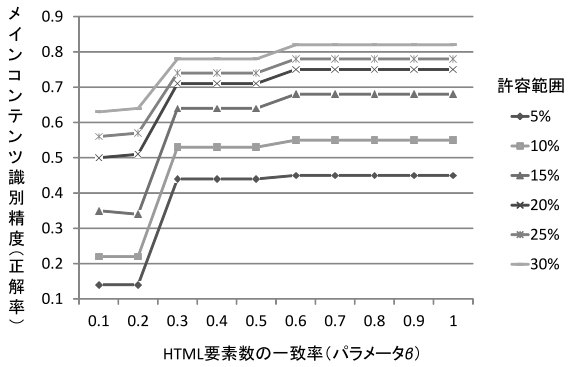

図 11 パラメータ β の決定
Fig. 11 Decision of parameter β.

表 3 メインコンテンツ要素の推定精度
Table 3 Estimate accuracy of element of main contents.

許容範囲	許容範囲内の件数		正解件数（許容範囲内で全記事を含む件数）		正解率（全件に対する正解数の割合）	
	全体	HR 有	全体	HR 有	全体	HR 有
		HR 無		HR 無		HR 無
5%	437	104	434	103	43.4%	20.6%
		333		331		66.2%
10%	554	180	550	178	55.0%	35.6%
		374		372		74.4%
15%	636	238	632	236	63.2%	47.2%
		398		396		79.2%
20%	636	276	684	274	68.4%	54.8%
		413		410		82.0%
25%	749	318	744	316	74.4%	63.2%
		431		428		85.6%
30%	798	356	792	354	79.2%	70.8%
		442		438		87.6%

表 3 における許容範囲内の件数とは，正解の判定の条件 1 を満たすものであり，正解件数と正解率は，正解の判定の条件 1 と条件 2 を満たす件数と割合を示す．表 3 を確認した結果，次に示す 3 つの特徴が明らかとなった．

- **高精度に投稿記事が含まれるメインコンテンツを推定可能であることが分かる**

表 3 の許容範囲内の件数と正解件数を確認すると，許容

範囲が 5%では 437 件中 434 件（99.3%），30%では 798 件中 792 件（99.2%）となり，許容範囲を広げるにつれて正解数は増加していることが分かる．また，許容範囲内の件数と正解件数の差を確認すると，許容範囲が 5%のときの 3 件に対し，30%のときでは 6 件となっており，許容範囲を広げた場合でも件数に大きな差がみられないという結果となった．そこで，許容範囲を 30%とした場合に正解したデータを確認すると，メインコンテンツとして推定した部分の最初や最後にレイアウトのための HTML タグや広告が含まれており，これらの部分が人手で精査したメインコンテンツとの差となっていることが分かった．しかし，これらの Web ページでは，図 9 に示すとおり広告部分を含めた HTML 要素がメインコンテンツ要素となっていたことから，本提案手法は高精度に投稿記事が含まれるメインコンテンツを推定できることが明らかとなった．

- **HR 無のページの方が高精度にメインコンテンツを推定可能であることが分かる**

表 3 の正解率を確認すると，HR 有と HR 無の差が許容範囲 5%で 45.6%，許容範囲 30%で 16.8%であり，許容範囲を広げるにつれて差は縮小しているものの，すべての許容範囲において HR 無のページの方が高精度にメインコンテンツを推定可能であることが分かる．これは，メインコンテンツの推定時に，メインコンテンツに含まれるテキストの量を用いており，HR 有のページは HR 無のページと比較してテキスト量が相対的に少ないため，誤判定につながったと考えられる．このことから，メインコンテンツ推定処理に用いる複数の Web ページを選定する際に，Web ページ内に含まれるテキスト長が他と比べて長いものを優先的に選定するなどの対応が必要であることが分かった．

- **投稿記事の分割の前処理として有用であることが分かる**

表 3 と**表 4** を確認するとメインコンテンツ要素推定機能では，許容範囲 30%における正解の 792 件に加えて，許容範囲外でもすべての投稿記事が含まれている 199 件の合計 991 件（99.1%）が投稿内容を含んだ結果を処理結果としていることが分かる．このことから，提案手法のメインコンテンツの推定は，投稿内容の分割において，不利益になる可能性が低く，前処理として有用であることが明らかとなった．また，失敗した 9 件を確認すると，DOM 構造の上位階層にある広告の影響により，その時点で一致率 β が閾値よりも下回ってしまいメインコンテンツ要素と異なる要素を推定したため，投稿記事の一部が欠けていることが分かった．これらの Web ページに対しては，メインコンテンツ要素はテキスト長が最長となるという特徴を利用して，テキスト長が一定以上となる要素のみを処理の対象とすることで対応できると考えられる．

表 4 失敗事例の内訳
Table 4 Details of failure case.

項目	件数
すべての投稿記事が含まれている	199
<body> タグをメインコンテンツとして推定	135
その他	64
投稿記事の一部が含まれている	9

6.4 実験 2：投稿記事抽出精度の比較実験

6.4.1 実験内容

本実験では，メインコンテンツを投稿記事単位に分割できないという課題を解決できているかを評価するため，人工的に生成した Web ページのメインコンテンツを用いて既存の Web ページ分割手法との比較実験を行い，その結果を考察する．

本実験では，既存の Web ページ分割手法として，Web ページ分割手法として様々な研究で引用されている VIPS [34] を用いた手法（以下，「VIPS」）と Web ページ内の見た目の包含関係に基づき分割する手法 [29]（以下，「包含手法」）とを採用する．VIPS は，Web ページを見た目に基づき分割する手法であり，そのままでは，提案手法および包含手法と比較できない．そのため，本実験では，VIPS により分割された HTML 要素（**図 12**）から投稿内容を特定する処理を加えたものを用いて比較する．VIPS における投稿記事の特定処理は，**図 13** に示すとおり，入力したメインコンテンツの子要素の中から，さらにもう 1 階層下の子要素を 1 件以上持つ要素を記事として判定する．

STEP 1：人工データセット 1,000 件（HR 有 500 件，HR

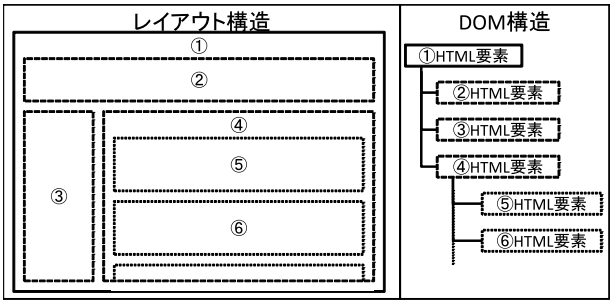

図 12 VIPS による Web ページの分割例
Fig. 12 Example of Web page segmentation by VIPS.

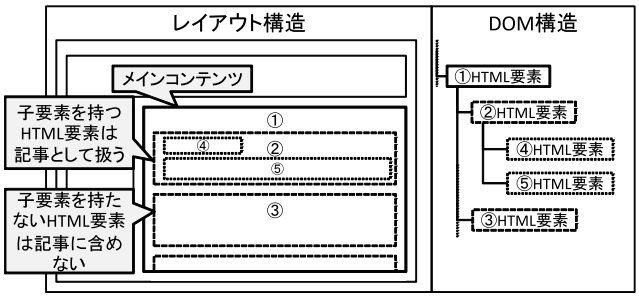

図 13 VIPS を用いた投稿記事の特定手法
Fig. 13 Method for identifying each posting article using VIPS.

無 500 件）から Web ページを取得する．

STEP 2：各 Web ページからメインコンテンツの HTML を抽出する．

STEP 3：メインコンテンツを対象に 3 つの手法で解析し，投稿記事を抽出する．

STEP 4：抽出した投稿記事の HTML 要素とメインコンテンツに含まれる正解の投稿記事の HTML 要素とを比較する．

STEP 5：**図 14**，**図 15** の判断基準に基づき，抽出結果を正常判定と誤判定とに分類して件数を集計する．

STEP 6：正常判定数と誤判定数とを用いて適合率，再現率，F 値を算出する．

6.4.2 実験結果と考察

投稿記事抽出精度の比較実験の結果を**表 5** に示す．比較実験の結果を確認すると次に示す 3 つの特徴がみられた．

- **提案手法は既存手法と比較して高精度に投稿記事を抽出できることが分かる**

既存手法との比較結果（表 5）の全体の精度を確認すると，提案手法は F 値 0.891 であるのに対して，包含手法は F 値 0.539，VIPS は F 値 0.398 であることが分かる．包含手法は，適合率が再現率よりも低いことから，全体の抽出数が多いがその中に正解が含まれている件数が少ないためであると考えられる．また，VIPS は，再現率が適合率よりも低いことから，抽出件数が少なく投稿記事を網羅的に抽出できていないためであると考えられる．このことから，提案手法では，メインコンテンツを投稿記事単位に分割できないという課題を解消できていることが明らかとなった．

- **提案手法は HR 有と HR 無の両方の Web ページから高精度に投稿記事を抽出できることが分かる**

既存手法との比較結果（表 5）の HR 有と HR 無の精度を確認すると，提案手法では HR 有で F 値 0.906，HR 無で F 値 0.877 となり，HR 有の方が 0.029 ポイント低い状態であるが，HR 有と HR 無でほぼ同等の精度で投稿記事を抽出できていることが分かる．それに対して，包含手法では HR 有で F 値 0.374，HR 無で 0.703 となり，HR 有の方が 0.329 ポイント低い状態である．これは，包含手法では HTML の各要素間の包含関係に基づきグループ化しているが，HR 有の Web ページでは HTML の各要素間の包含関係を取得できないため，誤抽出していると考えられる．これらのことから，提案手法は HR 有の Web ページも含めて，高精度にメインコンテンツを投稿記事単位に分割できることが明らかになった．

- **提案手法と比較して，包含手法は過剰分割，VIPS は抽出漏れが多いことが分かる**

各手法で抽出した投稿記事の傾向を分析するため，各手法における抽出数とその詳細を分析（**表 6**）した．分析結果を確認すると，提案手法は包含手法および VIPS と比較して抽出数に対する正常判定数の割合も高く，誤判定数も他の 2 手法と比較して少ないことが分かる．それぞれの手法の詳細な傾向を次に示す．提案手法は HR 有の場合に過剰範囲特定，HR 無の場合に抽出漏れの誤判定が発生する傾向がみられることが分かる．HR 有の場合の過剰範囲特定は，**図 16** に示すとおり，投稿記事を保持するタグが単一の <text> タグであった場合に，投稿記事のタグと同一の階層に，投稿内容以外の <text> タグが複数存在し，それらを誤抽出したためであると考えられる．一方，HR 無の場合の抽出漏れは，**図 17** に示すとおり，投稿記事を保持するタグが複数のタグで構成されている場合に，投稿内容によっては <a> タグの有無や タグの有無などの違いが発生することで，正しくグループ化ができず抽出漏れが発生したと考えられる．これらの課題については，タグの出現パターンを確認する方法に加えて，投稿記事に

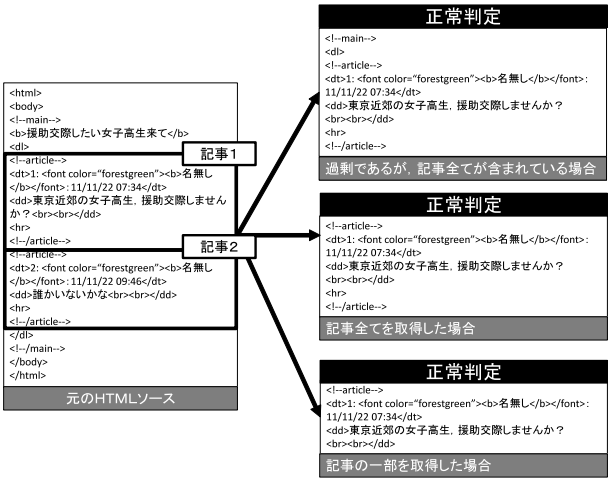

図 14 正常判定の判断基準

Fig. 14 Evaluation criteria of correct decision.

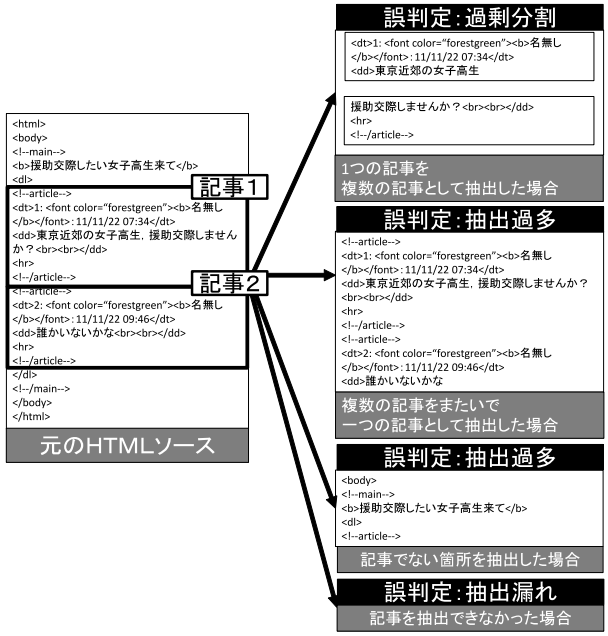

図 15 誤判定の判断基準

Fig. 15 Evaluation criteria of wrong decision.

表 5 各手法における抽出結果
Table 5 Results extracted by each method.

解析手法	VIPS			包含手法			提案手法		
	HR 有	HR 無	全体	HR 有	HR 無	全体	HR 有	HR 無	全体
適合率	0.685	0.556	0.621	0.302	0.606	0.450	0.888	0.885	0.886
再現率	0.344	0.246	0.292	0.491	0.836	0.673	0.925	0.869	0.896
F 値	0.458	0.341	0.398	0.374	0.703	0.539	0.906	0.877	0.891

表 6 Web ページ分割の精度比較
Table 6 Accuracy comparison of Web page segmentation.

解析手法		VIPS			包含手法			提案手法		
		HR 有	HR 無	全体	HR 有	HR 無	全体	HR 有	HR 無	全体
投稿件数		19,202	21,366	40,568	19,202	21,366	40,568	19,202	21,366	40,568
抽出件数		9,643	9,453	19,096	31,169	29,479	60,648	20,001	20,983	40,984
正常判定数		6,604	5,259	11,863	9,425	17,872	27,297	17,768	18,563	36,331
誤判定数	過剰範囲特定	1,245	3,361	4,606	5,150	1,448	6,598	1,824	1,332	3,156
	抽出漏れ	11,575	3,189	14,764	8,867	1,811	10,678	961	2,357	3,318
	過剰分割	1,796	831	2,627	16,594	10,159	26,753	409	1,088	1,497

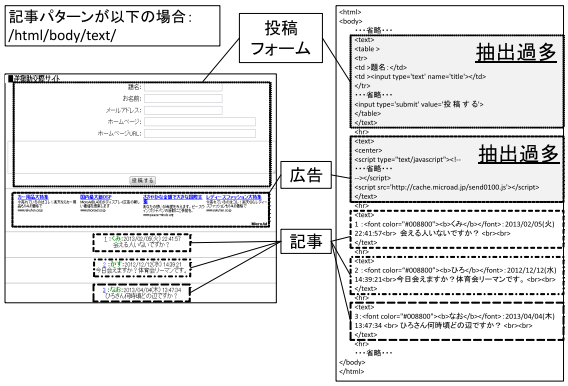

図 16 HR 有における記事の抽出過多の例
Fig. 16 Example of excessive extraction of articles on Web page with '<hr>' tag.

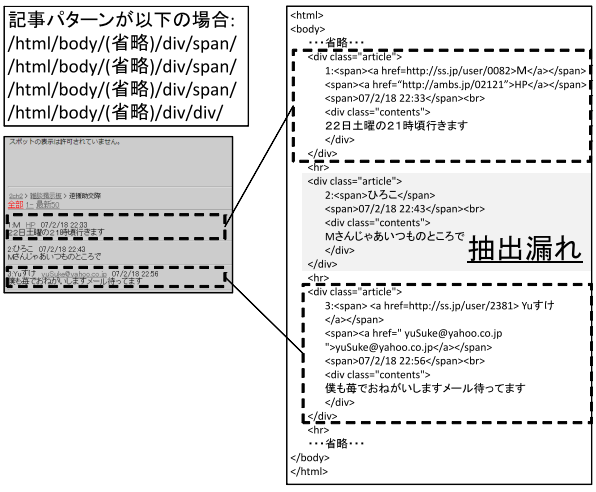

図 17 HR 無における記事の抽出漏れの例
Fig. 17 Example of extraction leakage of articles on Web page without '<hr>' tag.

含まれるテキスト情報（日付やタイトル，投稿者などの文字）の類似性を確認する処理を追加することで，対応可能であると考えられる．

包含手法は，HR 有において抽出数における正常判定数の割合が非常に低く，また，HR 有，HR 無ともに，過剰分割が多くみられることが分かる．これは，ページ分割後に HTML の各要素間の包含関係でグループ化する際に，包含関係を取得できず，グループ化されていないものが多く抽出されたものと考えられる．

VIPS は，投稿記事数に対する抽出数が非常に少なく，抽出漏れが多くみられることが分かる．これは，VIPS で分割した後に投稿記事を特定して判定しているが，子要素の有無で判断しているため，子要素が存在しない場合に抽出漏れが発生していると考えられる．

6.5 実験 3：提案手法の有用性の評価実験

6.5.1 実験内容

本実験では，提案手法の有用性を評価するため，次に示す 3 種類のデータを対象に投稿記事の抽出精度を算出し，その精度を比較する．

- 人工データ 1,000 件を対象にメインコンテンツの推定処理を行い，その推定結果であるメインコンテンツ部の HTML（以下，システム結果データ）
- 人工データ 1,000 件から手作業で抽出したメインコンテンツ部の HTML（以下，正解メインデータ）
- 人工データ 1,000 件の <body> タグ以下の HTML（以下，BODY データ）

システム結果データと正解メインデータとを対象に処理した結果を比較することで，システムの性能限界を評価する．また，システム結果データと BODY データとを対象に処理した結果を比較することで，メインコンテンツの推

定処理の有用性を評価する．

本実験の手順を次に示す．

STEP 1：システム結果データ，正解メインデータと BODY データを提案手法で解析し，投稿記事を抽出する．

STEP 2：それぞれのデータから抽出した投稿記事の HTML 要素とメインコンテンツに含まれる正解の投稿記事の HTML 要素とを比較する．

STEP 3：図 14，図 15 の判断基準に基づき，抽出結果を正常判定と誤判定とに分類して件数を集計する．

STEP 4：正常判定数と誤判定数とを用いて適合率，再現率，F 値を算出する．

6.5.2 実験結果と考察

提案手法の有用性の評価実験の結果を**表 7** に示す．評価実験の結果を確認すると次に示す 2 つの特徴がみられた．

- **システム結果データと正解メインデータとを対象に処理した結果はほぼ同様であることが分かる**

評価実験の結果（表 7）を確認すると，システム結果データを対象とした場合が F 値 0.890，正解メインデータを対象とした場合が F 値 0.891 となり，ほぼ同様の精度で投稿記事を抽出可能であることが分かる．これは，実験 1：メインコンテンツ推定精度の評価実験の考察で述べたとおり，メインコンテンツの推定結果の大半に記事部分が含まれていたため，解析範囲が限定されて高精度に投稿記事を抽出できたと考えられる．このことから，本提案手法は，多様なフォーマットの Web ページからメインコンテンツを推定し，メインコンテンツを投稿記事単位に分割することで，ネットパトロールの効率化を実現するうえで必要となる投稿記事単位に分割されたデータセットを構築することが可能であると考えられる．

- **システム結果データと BODY データとを対象に処理した結果を比較するとシステム結果データを対象とした方が高精度であることが分かる**

評価実験の結果（表 7）を確認すると，システム結果データを対象とした場合が F 値 0.890，BODY データを対象とした場合が F 値 0.829 となり，0.061 ポイントの差がみられた．これは，BODY データを対象とした場合，メインコンテンツ部以外にもフッタ，ヘッダ，メニューなどの繰返し要素がみられる部分が解析対象に含まれるため，それらを投稿記事として誤判定したためと考えられる．

7. おわりに

本研究では，違法・有害情報を判定する研究や Web クローラの研究の様々な成果をネットパトロールの支援へ適用する際に必要となる「多様なフォーマットの Web ページからメインコンテンツを推定する技術」および「メインコンテンツを投稿記事単位に分割する技術」を実現するために，個別詳細記事抽出のための Web ページ分割手法を提案した．そして，本提案手法の有用性を評価するため，実際の Web ページを模して生成したシミュレーションデータを用いて投稿記事の抽出精度の比較実験を行った結果，F 値 0.888 となり，従来の Web ページ分割手法を用いた手法と比較して高精度に投稿記事を抽出できることが明らかとなった．これらの結果から，個別詳細記事抽出のための Web ページ分割手法の有用性を証明した．今後は，Web ページの調査時と同様の手法をシステム化することで，実データにおける同一ドメイン内にある同一フォーマットの Web ページ自動収集手法を自動化し，本提案手法を実データに適用する．そして，本提案手法の有用性と汎用性を評価するとともに，ネットパトロールの支援へ応用する予定である．具体的には，Web クローラにおける記事単位での情報収集手法の考案や，コミュニケーションサイトにおける投稿内容として多くみられる相対的な時間表現などを正しい表現に置き換え可能な高度なネットパトロールシステムを構築する予定である．また，本提案手法は，多様なフォーマットの Web ページを対象としているため，ネットパトロールの分野のみに限定せず，CGM から投稿者の属性や嗜好を分析して EC サイトでのマーケティングに利用するなど，幅広い分野に適用できる技術だと考える．今後は，本手法を利活用した他分野への適用とその検証を行う予定である．

表 7 提案手法の有用性評価

Table 7 Effectiveness evaluation of proposed method.

解析手法		BODY データ			システム結果データ			正解メインデータ		
		HR 有	HR 無	全体	HR 有	HR 無	全体	HR 有	HR 無	全体
投稿件数		19,202	21,366	40,568	19,202	21,366	40,568	19,202	21,366	40,568
抽出件数		19,980	22,618	42,598	20,048	20,959	41,007	20,001	20,983	40,984
正常判定数		17,658	16,827	34,485	17,766	18,527	36,293	17,768	18,563	36,331
誤判定数	過剰範囲特定	1,922	4,938	6,860	1,816	1,332	3,148	1,824	1,332	3,156
	抽出漏れ	985	4,063	5,048	963	2,393	3,356	961	2,357	3,318
	過剰分割	400	853	1,253	466	1,100	1,566	409	1,088	1,497
適合率		0.884	0.744	0.810	0.886	0.884	0.885	0.888	0.885	0.886
再現率		0.920	0.788	0.850	0.925	0.867	0.895	0.925	0.869	0.896
F 値		0.901	0.765	0.829	0.905	0.875	0.890	0.906	0.877	0.891

参考文献

[1] 内閣府：平成 24 年度版子ども・若者白書, 佐伯印刷 (2012).
[2] 小針 誠：学校裏サイトにおける「ネットいじめ」の構造と対策，児童心理，Vol.63, No.13, pp.94–99, 金子書房 (2009).
[3] 群馬大学社会情報学研究センター：青少年の携帯電話等の利用問題に関する全国指定都市・都道府県教育委員会調査結果 (2010).
[4] 桑子博行：我が国における児童ポルノのブロッキングの仕組みと今後の展望（特集 インターネット上に氾濫する違法・有害情報の現状と対策），警察学論集，Vol.64, No.8, pp.67–93, 立花書房 (2011).
[5] 苗村憲司：サイバー防犯ボランティアの意義と育成支援方策について（特集 インターネット上に氾濫する違法・有害情報の現状と対策），警察学論集，Vol.64, No.8, pp.51–66, 立花書房 (2011).
[6] 丸橋 透：「青少年有害情報」と民事責任（特集 ネットワーク社会における青少年保護 第 35 回法とコンピュータ学会研究会報告），法とコンピュータ，Vol.29, pp.65–81, 法とコンピュータ学会 (2011).
[7] Lee, W., Lee, S.S., Chung, S. and An, D.: Harmful Contents Classification Using the Harmful Word Filtering and SVM, *Proc. 7th International Conference on Computational Science*, pp.18–25, Springer-Verlag (2007).
[8] Guermazi, R., Hammami, M. and Hamadou, A.: Combining Classifiers for Web Violent Content Detection and Filtering, *Proc. 7th International Conference on Computational Science*, pp.773–780, Springer-Verlag (2007).
[9] Lee, P.Y., Hui, S.C. and Fong, A.C.M.: Neural Networks for Web Content Filtering, *IEEE Intelligent Systems*, Vol.17, No.5, pp.48–57 (2002).
[10] Du, R., Safavi-Naini, R. and Susilo, W.: Web Filtering Using Text Classification, *IEEE International Conference on Networks*, Vol.11, pp.325–330 (2003).
[11] Chandrinos, K.V., Androutsopoulos, I., Paliouras, G. and Spyropoulos, C.D.: Automatic Web Rating: Filtering Obscene Content on the Web, *Proc. 4th European Conference on Research and Advanced Technology for Digital Libraries*, pp.403–406, Springer-Verlag (2000).
[12] 菊池琢弥，内海 彰：語の共起情報に基づく有害サイトフィルタリング手法，第 9 回情報科学技術フォーラム講演論文集，Vol.FIT2010, No.2, pp.1–6, 情報処理学会・電子情報通信学会 (2010).
[13] 池田和史，柳原 正，服部 元，松本一則，小野智弘，滝嶋康弘：HTML 要素に基づく有害サイト検出手法，情報処理学会論文誌，Vol.52, No.8, pp.2474–2483, 情報処理学会 (2011).
[14] 松葉達明，里見尚宏，桝井文人，河合敦夫，井須尚紀：学校非公式サイトにおける有害情報検出，言語理解とコミュニケーション研究会技術研究報告，Vol.109, No.142, pp.93–98, 電子情報通信学会 (2009).
[15] 中村健二，田中成典，大谷和史，山本雄平：セキュアライフの創出を目指した安全知の獲得に関する研究—電子掲示板からの犯行予告の抽出，土木情報利用技術論文集，Vol.18, pp.269–280, 土木学会 (2009).
[16] 吉村卓也，藤井雄太郎，伊藤孝行：Robinson 型判定手法を用いた単語共起フィルタの検証，第 10 回情報科学技術フォーラム講演論文集，Vol.FIT2011, No.2, pp.85–90, 情報処理学会・電子情報通信学会 (2011).
[17] 柳原 正，池田和史，服部 元，松本一則，滝嶋康弘：単語と単語ペアの統合有害度に基づく特徴量生成手法の提案，言語理解とコミュニケーション研究会技術研究報告，Vol.110, No.142, pp.27–32, 電子情報通信学会 (2010).
[18] Juan, D. and Zhi, A.Y.: A Categorization Algorithm for Harmful Text Information Filtering, *Proc. 4th International Conference on Multimedia Information Networking and Security*, pp.31–34, IEEE (2012).
[19] Akilandeswari, J. and Gopalan, P.: An Architectural Framework of a Crawler for Locating Deep Web Repositories Using Learning Multi-agent Systems, *Proc. 3rd International Conference on Internet and Web Applications and Services*, pp.558–562, IARIA (2008).
[20] Hua, J., Bing, H., Ying, L., Dan, Z. and Yong, G.: Design and Implementation of University Focused Crawler Based on BP Network Classifier, *Proc. 2nd International Workshop on Knowledge Discovery and Data Mining*, pp.44–47, IEEE (2009).
[21] Ibrahim, A., Selamat, A. and Selamat, H.: Scalable E-business Social Network Using MultiCrawler Agent, *Proc. International Conference on Computer and Communication Engineering*, pp.702–706, IEEE (2008).
[22] Hai-ling, X. and Jing, D.: Specific Web Spider Design for the Extraction of Unknown Chinese Words from BBS Corpus, *Proc. 2nd International Conference on Future Information Technology and Management Engineering*, pp.499–502, IEEE (2009).
[23] 中村健二，田中成典，北野光一，寺口敏生，大谷和史：マルチエージェントクローラを用いた有害ユーザの効率的発見手法，情報処理学会論文誌，Vol.53, No,1, pp.90–104, 情報処理学会 (2012).
[24] Orkut, B., Hector, G.M. and Andreas P.: Seeing the Whole in Parts: Text Summarization for Web Browsing on Handheld Devices, *Proc. 10th International Conference on World Wide Web*, pp.652–662, ACM (2001).
[25] 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，人工知能と知識処理研究会技術研究報告，Vol.108, No.119, pp.81–86, 電子情報通信学会 (2008).
[26] Cai, D., Yu, S., Wen, J.R. and Ma, W.Y.: Extracting Content Structure for Web Pages Based on Visual Representation, *Proc. 5th Asia-Pacific Web Conference on Web Technologies and Applications*, pp.406–417, Springer-Verlag (2003).
[27] Hattori, G., Hoashi, K., Matsumoto, K. and Sugaya, F.: Robust Web Page Segmentation for Mobile Terminal Using Content-distances and Page Layout Information, *Proc. 16th International Conference on World Wide Web*, pp.361–370, ACM (2007).
[28] Guo, H., Mahmud, J., Borodin, Y., Stent, A. and Ramakrishnan, I.: A General Approach for Partitioning Web Page Content Based on Geometric and Style Information, *Proc. 9th International Conference on Document Analysis and Recognition*, Vol.2, pp.929–933, IEEE (2007).
[29] 中村健二，田中成典，山本雄平，安彦智史：共起関係の抽出範囲を考慮した有害情報フィルタリング手法，情報処理学会論文誌，Vol.54, No.2, pp.571–584, 情報処理学会 (2013).
[30] Cong, K.N., Likforman, S.L., Moissinac, J.C. and Lardon, J.: Web Document Analysis Based on Visual Segmentation and Page Rendering, *Proc. 10th IAPR International Workshop on Document Analysis Systems*, pp.354–358, IEEE (2012).
[31] 佐野博之，白松 俊，大囿忠親，新谷虎松：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，電子情報通信学会論文誌，Vol.J95-D, No.4, pp.909–918, 電子情報通信学会 (2012).
[32] Ziv, J. and Lempel, A.: A Universal Algorithm for Sequential Data Compression, *IEEE Trans. Information Theory*, Vol.23, No.3, pp.337–343, IEEE (1997).

[33] Jetrun テクノロジ株式会社：隠語・誘導語データベース，入手先 ⟨http://www.kkyg.jp/⟩（参照 2013-05-13）．
[34] Cai, D., Yu, S., Wen, J.R. and Ma, W.Y.: VIPS: A Vision-based Page Segmentation Algorithm, Microsoft Technical Report, MSR-TR-2003-79, Microsoft (2003).

山本 雄平 （学生会員）

2009 年関西大学総合情報学部総合情報学科卒業．2011 年関西大学大学院総合情報学研究科知識情報学専攻博士課程前期課程修了．現在，関西大学大学院総合情報学研究科総合情報学専攻博士課程後期課程在学中．修士（情報学）．Web マイニング，自然言語処理，Web ソリューションビジネスに関連する研究に従事．2007 年（株）関西総合情報研究所入社．現在に至る．システム設計，データモデル設計等の研究開発に従事．

中村 健二 （正会員）

2004 年関西大学総合情報学部総合情報学科卒業．2006 年関西大学大学院総合情報学研究科知識情報学専攻博士課程前期課程修了．2009 年関西大学大学院総合情報学研究科総合情報学専攻博士課程後期課程修了．同年関西大学ポスト・ドクトラル・フェロー，2010 年立命館大学情報理工学部助手，2012 年大阪経済大学情報社会学部准教授，現在に至る．博士（情報学）．知識情報処理，Web マイニング，テキストマイニング等の研究に従事．2002 年から（株）関西総合情報研究所にて活動．システム設計，データモデル設計等の研究開発に従事．電子情報通信学会，土木学会，日本データベース学会，日本知能情報ファジィ学会各会員．

田中 成典 （正会員）

1986 年関西大学工学部土木工学科卒業．1988 年関西大学大学院工学研究科土木工学専攻博士課程前期課程修了．同年（株）東洋情報システム（現在，TIS）に入社．人工知能に関する研究受託開発業務に従事．1994 年関西大学総合情報学部専任講師として着任．1997 年助教授，2004 年教授，2006 年から学生センター副所長，現在に至る．2002 年 8 月から 1 年間，カナダの UBC にて客員助教授．博士（工学）．専門は知識工学と社会基盤情報学．CAD/CG，GIS/GPS，画像処理および Web ソリューションビジネスに関する研究に従事．2000 年（株）関西総合情報研究所を起業，設立当初から現在まで取締役会長．2006～2012 年（株）フォーラムエイトの顧問．主に，ISO に準拠した CAD 製図基準と CAD データ交換基盤の開発に従事．

安彦 智史 （正会員）

2008 年関西大学総合情報学部総合情報学科卒業．2010 年関西大学大学院総合情報学研究科知識情報学専攻博士課程前期課程修了．2013 年関西大学大学院総合情報学研究科総合情報学専攻博士課程後期課程修了．博士（情報学）．画像処理，Web ソリューションビジネスに関連する研究に従事．2006～2013 年まで（株）関西総合情報研究所．