Vol.2009-DBS-149 No.20
2009/11/21

ブログ記事集合を用いた ポストとコメントとの自動分離抽出手法の提案

吉田光男[†1] 乾孝司[†1] 山本幹雄[†1]

近年のブログの普及により，ブログ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．ブログのコンテンツは，ポストと呼ばれるブログの書き手によるコンテンツと，コメントと呼ばれるブログの読者によるコンテンツに大分する事ができる．ブログのコンテンツを利用する場合は，それらが別々に抽出できている事が望ましい．本論文では，ブログ記事集合を用いる事により，ポストとコメントを自動的に分離抽出する手法を提案する．本手法は，ポストはブログ記事集合全てのブログ記事に出現するが，コメントはいずれかのブログ記事にしか出現しないというアイデアが基になっている．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の有効性を示す．

Automatic Extraction for Blog Posts and Comments using a Set of Blog Pages

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

In recent years, with the increase in the number of Blog (Weblog) pages in the Web, the contents of which have attracted a lot of attention from many researchers. Most of the Blog contents are texts, and those can be divided in two parts, posts and comments. A post is a content written by the Blog owner and a comment is piece of text written by readers in response to the owner's post. In this paper, we propose a simple method to extract the posts and comments separately from series of Blog pages, whose posts are all written by the same owner. The proposed method is based on the assumption that although posts appear in all Blog pages, comments do not. We describe experimental results to show good performance of the proposed method using real Web pages of the Blog sites in Japanese.

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[2)–4)]．

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．また，ブログのコンテンツは，ポストと呼ばれるブログの書き手によるコンテンツと，コメントと呼ばれるブログの読者によるコンテンツに大分する事ができる．たとえば，**図 1** のブログ記事[⋆1]には，上部にポストが，下部にコメントが存在する（破線部分）．従来のコンテンツ抽出（本文抽出，主要部分抽出などとも呼ばれる）は，ポストとコメントを区別せずに抽出したり，ポストのみを抽出したりしていた．しかし，ブログのコメントを利用する研究[5)–8)] が行われ始めるなど，今後，ブログのコンテンツ抽出はポストとコメントを分離抽出する事が期待される．

本論文では，ブログ記事集合を用いる事により，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，ブログ記事からポストとコメントを自動的に分離抽出する手法を提案する．本手法は，コンテンツのうち，ポストはブログ記事集合全てのブログ記事に出現するが，コメントはいずれかのブログ記事にしか出現しないというアイデアが基になっている．

以降，2 節で関連研究について，3 節で提案手法について，4 節で評価指標，日本語ブログサイトを対象とした実験結果及び考察について，最後の 5 節でまとめを述べる．

2. 関連研究

前節で述べたとおり，ブログのコンテンツはポストとコメントに大分する事ができる．本節では，ポストとコメントを分離抽出する従来手法について述べる．

ポストとコメントを分離抽出する手法としては，まず，人手によって抽出ルールを記述す

†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 http://www.100shiki.com/archives/2009/08/gscreen.html (cited 2009-10-23)

Vol.2009-DBS-149 No.20
2009/11/21

図 1 ブログにはポストとコメントが存在する
Fig. 1 An example of a Blog page with posts and comments.

る方法が考えられる．この方法の代表例として，正規表現[9] やラッピング言語[10] が挙げられる．しかし，インターネット上には無数のブログ記事が存在しており，各ブログ記事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．

分離抽出をある程度自動化する試みも行われている．Cao ら[11] は，ポストとコメントを分断する 1 本の境界線を発見する事により自動的に分離抽出する手法を提案している．この手法は，視覚情報とテキスト情報を基にブログ記事のコンテンツを特定する処理と，情報量を基にコンテンツの中で境界線を発見する処理の 2 段階にわけられる．彼らの手法には，コンテンツを 1 カ所に特定するため，コンテンツの中に不要部分を多く含む傾向が高いという問題点がある．また，境界線の存在が前提であるため，コメントが付いていないブログ記事には適用が難しいという問題点もある．

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[12] を拡張し，ポストとコメントを分離抽出する手法を提案する．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定義した，コンテンツと不要部分のかたまり（ブロック）を単位として考え，ブロックごとにコンテンツ判定を行うため，コンテンツと不要部分を高精度に分離できる．手法の拡張にあたり，ブロックレベル要素の要素識別子を用いる事により，ブロックに位置ラベルを付与する手法を新たに提案する．そして，この位置ラベルを手がかりに，ブログ記事集合全てのブログ記事に同じ位置ラベルを持つ，コンテンツと認められたブロックが出現するかどうかを判定し，ポストとコメントを分離抽出する手法を提案する．

3. 提案手法

3.1 コンテンツ及びポスト・コメントの定義

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．また，ブログのコンテンツは，さらにポストとコメントに大分する事ができる．本論文では，ブログの書き手による記事本文を**ポスト**とする．また，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・図の説明文もポストとみなす．そして，ブログの読者によるコメント本文及びトラックバック*2本文を**コメント**とする．また，コメント本文及びトラックバック本文に付随するタイトル，投稿日時，コメント著者名（トラックバック送信元ブログ名）もコメントとみなす．なお，ポストとコメントを区別しない場合は，両方をまとめてコンテンツと呼ぶ．

3.2 ポスト・コメント自動分離抽出手法の概要

本論文で提案するポストとコメントを自動的に分離抽出する手法（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[12]（以下，**コンテンツ自動抽出手法**と呼ぶ）の拡張である．本小節では，提案手法の概要と，コンテンツ自動抽出手法との差異について述べる．

*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 ブログ特有のリンク通知システム

Vol.2009-DBS-149 No.20
2009/11/21

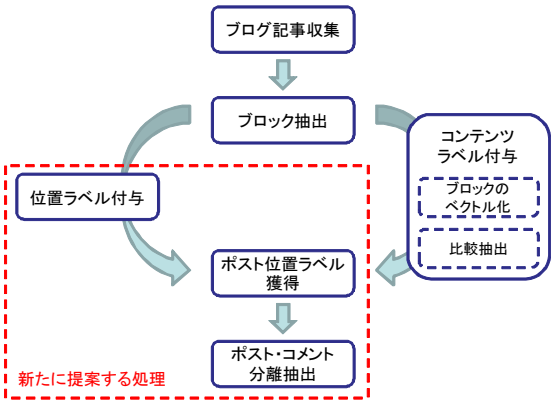

図 2 提案手法の概観
Fig. 2 The overview of the proposed method.

筆者らは，同じブログサイト内であれば，ポストは全てのブログ記事に出現し，出現位置も似ている傾向がある事に着目した．そして，コメントは全てのブログ記事には付かない傾向がある事がわかった．その結果，コンテンツと認められたブロックの位置を特定し，ブログサイト内でどのように出現するかを調べる事で，そのブロックをポストとコメントに分離できると考えた．具体的には，ブログ記事集合全てのブログ記事の同じ場所に出現する，コンテンツとして認められるブロックをポストとして抽出し，残りのコンテンツとして認められるブロックをコメントとして抽出する．

提案手法の概観を**図 2** に示す．コンテンツ自動抽出手法は「コンテンツラベル付与」の結果のみで抽出を行う手法であるが，提案手法はコンテンツと認められたブロックの位置を特定するために「位置ラベル付与」,「ポスト位置ラベル獲得」の処理を加え,「ポスト・コメント分離抽出」を行う手法である．

以降の小節で，図 2 に示した処理の詳細を説明する．3.3 小節から 3.5 小節については，文献 12) も参照されたい．

3.3 ブログ記事の収集

提案手法は，ブログ記事集合を用いて分離抽出を行うため，複数のブログ記事を収集する必要がある．収集する事によって得られる**ブログ記事集合**は，以下の集合であるとする．

(1) 同じブログサイトの記事で構成される
(2) コメントが付いていない記事が含まれる

一般的にブログ記事のコメント率は低い．また，フィード[⋆1]を用いて最新記事を更新直後に収集すれば，少なくとも 1 記事はコメントが付く前に収集できると考える．従って，上の条件（2）は特殊事例とならない．なお，上の条件（2）が満たされていない場合は，前小節で述べた仮説上，コメントはポストとして抽出される事が予想される．

3.4 ブロックの抽出

本論文では，コンテンツ及び不要部分の最小単位を**ブロック**と呼ぶ．本小節では，DOM ツリーからブロックを抽出する手法を述べる．

Web ページで使用されている HTML タグは，WWW で使用される技術の標準化を進める国際団体である W3C によって定められている．W3C の定めた HTML タグは，Web ページ内の見出しや段落など文書の基本構造を構成するためのブロックレベル要素（H1, P, DIV, TABLE など）と，特定の語の修飾やハイパーリンクを設置するためのインライン要素（FONT, STRONG, A, IMG など）に大分する事ができる[13]．

本手法では，ブロックの抽出にブロックレベル要素を用いる．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 3** 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）．

3.5 コンテンツラベルの付与

ポストとコメントを分離抽出する前に，ブロックをコンテンツと不要部分に分類する．本論文では，コンテンツに分類されたブロックを表すラベルを**コンテンツラベル**と呼び，コンテンツラベルが付与されたブロックを**コンテンツブロック**と呼ぶ．

以降の小々節で，コンテンツラベルの付与に必要な 2 過程を説明する．

3.5.1 ブロックのベクトル化

ブロック間の微小な違いを許容して一致を判断するために，ブロックをベクトルで表現する．本論文では，そのベクトルを**ブロックベクトル**と呼ぶ．本手法では，ブロックベクトルのベクトル素性としてブロックに含まれる要素名，テキスト， title, alt, src 属性の属性値

⋆1 RSS, Atom など Web サイトの更新情報を記述した XML ファイル

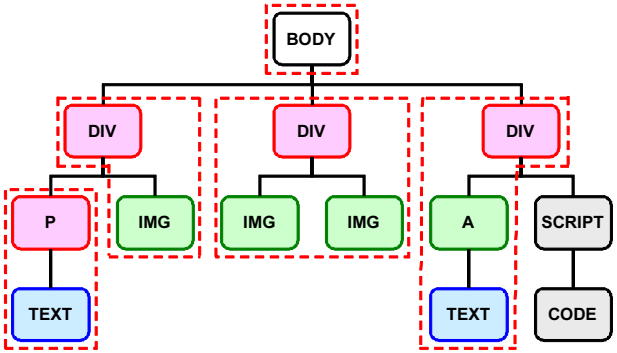

図 3 DOM ツリーから 5 つのブロックを抽出した例（破線部分）
Fig. 3 An example of extracted blocks (dashed-line box) from the DOM tree.

を用いる．素性の値はそれぞれの出現回数とする．なお，テキストの出現回数をカウントする際，テキストを改行ごとに個々のベクトル素性とみなし，英字の場合は全てを小文字に正規化して用い，空白のみのテキストは除外している．

ブロックに含まれる要素名の数は，ブロックのレイアウト情報を表現している．また，ブロックに含まれるテキストの数はブロックの内容を， title, alt, src 属性の属性値の数は画像（IMG）が出現するブロックの内容を表現している．

3.5.2 比較抽出

ブロックの一致を判断し，他のブログ記事には出現しないブロック，すなわちブログ記事集合の中で 1 度だけ出現するブロックにコンテンツラベルを付与し，コンテンツブロックを抽出する．ブロックの一致判断は，前小々節で述べたブロックベクトルを用いて，ブロックベクトル同士のコサイン類似度を計算する事により行う．ここでは，ブロック間の類似度が **0.9**[⋆1] を越えた時，それらのブロックは一致したと認める．コサイン類似度を用いる事により，レンダリングにほとんど影響を与えない微小な違いを許容する事ができる．

3.6 位置ラベルの付与

コンテンツブロックの位置を特定し，同じブログサイト内でコンテンツブロックがどのように出現するかを調べるためには，ブロックの位置を表すラベルを付与する必要がある．本論文では，ブロックの位置を表すラベルを**位置ラベル**と呼ぶ．本小節では，ブロックに位置ラベルを付与する手法について述べる．

3.4 小節で述べたとおり，ブロックの抽出はブロックレベル要素を基に行われる．その際，ブロック内に複数のブロックレベル要素が含まれないように抽出を行うため，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．また，ブロック間には DOM ツリー同様の親子，兄弟関係が存在すると考える事ができ，本小節では，ブロックが DOM ツリー上に存在するかのように取り扱い，説明する．

本手法では，ブロックレベル要素の要素識別子（Element identifiers）を用いてブロックに位置ラベルを付与する．この処理は，以下の 3 過程にわける事ができる．

処理 1 ブロックレベル要素の要素識別子をブロックの識別子に変換する
処理 2 ブロックの識別子のうち特定の条件を満たさないものを除外する
処理 3 ブロックの識別子を基にブロックに位置ラベルを付与する

まず，処理 1 を説明する．ブロックの**識別子**は，そのブロックに対応するブロックレベル要素の要素識別子とする．ブロックの識別子はブロックの位置ラベル候補である．全てのブロックレベル要素に要素識別子が存在するとは限らないため，あるブロックには識別子が存在しないという状態が存在する．なお，要素識別子とは id 属性または class 属性の事である[13]．

次に，処理 2 を説明する．ブログ記事集合に含まれるいずれのブログ記事に対しても統一的にブロックの位置を表現するために，位置ラベルとして用いるブロックの識別子は以下の両条件を満たすものとし，それ以外の識別子は除外する．

- 1 記事中に 1 度のみ出現する
- ブログ記事集合全てのブログ記事に出現する

1 記事中に 1 度のみ出現する識別子を用いるという事は，識別子の対応を 1 ブロックに絞るという事である．後ほど説明する処理 3 では，この識別子を基に周辺のブロックに位置ラベルを付与するが，その結果，ある位置ラベルが付与されたブロックは 1 カ所に集まる．また，全てのブログ記事で出現しない識別子は，個々のブログ記事特有の情報を持つ可能性が高く，ブログ記事集合の統一的な位置を表現していない．そのため，ブログ記事集合全てのブログ記事に出現しない識別子は除外する．

最後に，処理 3 を説明する．処理 1 と処理 2 の説明で述べたとおり，全てのブロックに識別子が存在するとは限らず，識別子をそのまま位置ラベルにしただけでは，全てのブロックに位置ラベルが付与されない場合がある．そこで，各ブロックを前順（preorder）に走査し，次の優先順位に従って順次位置ラベルを付与する．なお，id 属性と class 属性の両方を

⋆1 この値は経験的に決定した

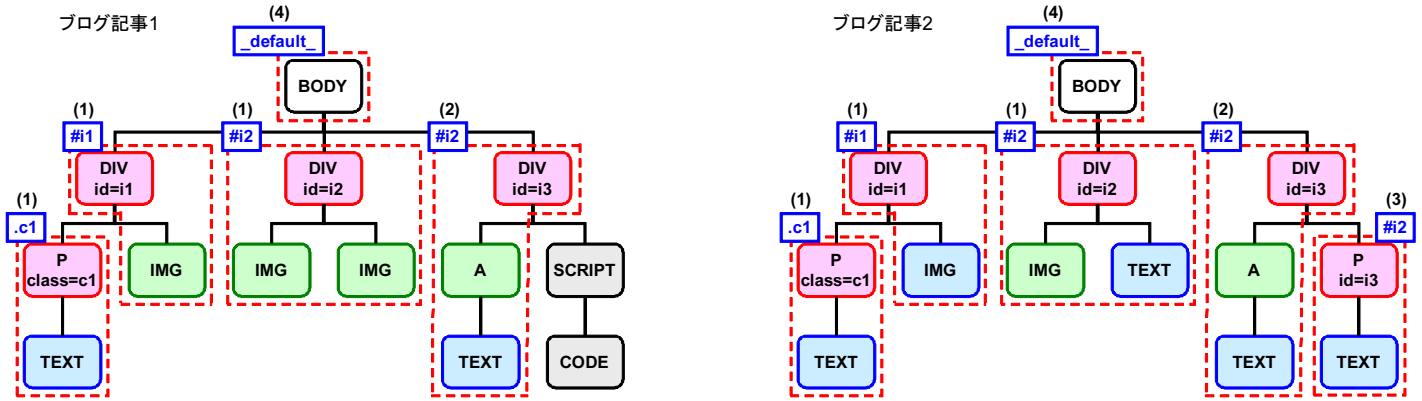

図 4 各ブロックに位置ラベルを付与した例
Fig. 4 An example of blocks with positional labels.

用いる事ができる場合は id 属性を用いる．

（1） 自身の識別子を付与する
（2） 隣接する兄ノードの位置ラベルを付与する
（3） 親ノードの位置ラベルを付与する
（4） 標準値（_default_）を付与する

図 4 は，DOM ツリー（要素識別子は要素内に記載）からブロックを抽出し，位置ラベルを付与した例である．なお，スタイルシート（CSS）のセレクタ表現にならい，id 属性を基に位置ラベルを決定した場合は「#」を，class 属性を基に位置ラベルを決定した場合は「.」を属性値の接頭に付けて区別した．また，括弧付き数字は位置ラベルを決定した際に使われた上の優先順位を表す．たとえば，要素識別子「id=i3」は，両方のブログ記事に出現しているものの，ブログ記事 2 において 2 度出現しているため，ブロックの識別子から除外し位置ラベルとして付与されない．

3.7 ポスト位置ラベルの獲得

前小節の方法でブロックに付与された位置ラベルは，ブロックがコンテンツブロックであるか否かは関知していない．そこで，位置ラベルのうち，ブログ記事集合全てのブログ記事において，少なくとも 1 つのコンテンツブロックに付与された位置ラベルを**ポスト位置ラベル**とする．

3.8 ポストとコメントの分離抽出

各ブログ記事において，コンテンツブロックのうち，ポスト位置ラベルが付与されているブロックをポストとし，残りのコンテンツブロックをコメントとして抽出する．

4. 実験と今後の課題

4.1 評 価 指 標

本実験の評価指標は，3.1 小節の定義に従って人手により作成したポストとコメントの正解データに対する適合率（Precision），再現率（Recall），F 値（F-measure）を使用した．

提案手法によってポストとして抽出されたブロックの数を N，抽出されたポストのうち正解データに適合していたブロックの数を R，正解データに含まれるポストのブロックの数を C とすると，ポストの適合率，再現率，F 値は次のように計算できる．

$$Precision = \frac{R}{N}$$
$$Recall = \frac{R}{C}$$
$$F = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

また，コメントの適合率，再現率，F 値も同様に計算する．

Vol.2009-DBS-149 No.20
2009/11/21

4.2 日本語ブログサイトを対象とした実験結果

使用したデータセットは，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[⋆2]，Engadget Japanese[⋆3]，ネタフル[⋆4]，404 Blog Not Found[⋆5]，IDEA*IDEA[⋆6]，My Life Between Silicon Valley and Japan[⋆7]，Going My Way[⋆8]，TechCrunch Japan[⋆9]，Life is beautiful[⋆10] から 2009 年 4 月 11 日に収集した計 206 記事である（**表 1**「データセット詳細」）．ポスト及びコメントの候補となるブロックは 35,216 ブロック存在し，人手によってポストと認められるブロックは 2,932 ブロック，コメントと認められるブロックは 973 ブロックであった．また，使用したデータセットでは 38%のブログ記事にコメントが付いていた．

実験結果を表 1「ポスト抽出性能」「コメント抽出性能」に示す．「My Life Between...」はコメントが自動抽出されなかったためコメントの適合率が計算できず，また「Going My Way」は正解データにコメントが含まれていなかったためコメントの再現率が計算できず「-」とした．実験結果より，提案手法は全体的に高い性能を示している事がわかる．なお，コメント抽出性能に関しては，正解ブロック数（コメント）が少ないため実験結果が安定していないが，「404 Blog Not Found」「Life is beautiful」からわかるとおり，十分なコメントがあれば高い性能で安定すると考える．**図 5** は提案手法による抽出を行ったブログ記事[⋆11]の例である（破線着色部分がポスト，実線着色部分がコメントを示す）．

3.3 小節で述べたとおり，提案手法には前提が存在する．「My Life Between...」には全記事にコメントが付いており，自動的に分離抽出した結果，全てのコメントがポストとして抽出された．また，「Going My Way」には全記事にコメントが付いておらず，本来であれば全てポストとして抽出されるが，コンテンツ抽出の失敗により，コメントとして抽出されるブロックが存在した．

⋆1 http://reader.livedoor.com/
⋆2 http://www.100shiki.com/
⋆3 http://japanese.engadget.com/
⋆4 http://netafull.net/
⋆5 http://blog.livedoor.jp/dankogai/
⋆6 http://www.ideaxidea.com/
⋆7 http://d.hatena.ne.jp/umedamochio/
⋆8 http://kengo.preston-net.com/
⋆9 http://jp.techcrunch.com/
⋆10 http://satoshi.blogs.com/
⋆11 http://blog.livedoor.jp/dankogai/archives/51185176.html

図 5 実験結果のブログ記事例（分離抽出後）
Fig. 5 An example of a Blog page and the extracted posts and comments.

4.3 分離手法のみの性能評価実験

提案手法の性能は，コンテンツ自動抽出手法の性能に依存している．本小節では，分離抽出のみの評価を行い，その有効性を示す．

まず，正解データを基に分離抽出する事により提案手法の有効性を示す．この実験では，コンテンツを過不足なく抽出できたと仮定した場合の性能がわかる．実験結果を**表 2** に示す．実験結果より，ほぼ理想的に分離抽出できている事がわかる．コンテンツ自動抽出が理想的に行われた場合，提案手法は非常に高い性能を達成できることが明らかになった．

次に，分離抽出の上限値を示す事により提案手法の有効性を示す．コンテンツ自動抽出に失敗したポスト及びコメントは，その後の処理である分離手法では抽出不可能であるため，表 1「ポスト抽出性能」「コメント抽出性能」の再現率には上限値が存在する．コンテンツ自動抽出の結果から正解データに含まれないブロック（不要部分）を除外し，残りのブロックを完璧に分離抽出した場合の再現率は，それぞれ 87.1%と 91.6%であった．提案手法による性能（再現率）は，それぞれ 82.2%と 86.4%である．この事から，提案する分離手法

表 1 提案手法によるポストとコメントの分離抽出性能
Table 1 Data sets and Experimental results of the extraction.

	データセット詳細				ポスト抽出性能（%）				コメント抽出性能（%）			
ブログ名	記事数	ブロック	ポスト	コメント	抽出数	適合率	再現率	F 値	抽出数	適合率	再現率	F 値
100SHIKI	10	725	153	12	150	88.7	86.9	87.8	17	41.2	58.3	48.3
Engadget Japanese	40	6163	200	36	175	90.3	79.0	84.3	63	49.2	86.1	62.6
ネタフル	30	2494	451	4	371	91.9	75.6	83.0	88	4.5	100.0	8.7
404 Blog Not Found	50	19264	1008	750	835	99.2	82.1	89.9	715	95.1	90.7	92.8
IDEA*IDEA	9	569	121	6	129	86.8	92.6	89.6	11	27.3	50.0	35.3
My Life Between...	5	575	166	46	233	71.2	100.0	83.2	0	-	0.0	-
TechCrunch Japan	20	1951	182	8	161	88.8	78.6	83.4	12	50.0	75.0	60.0
Life is beautiful	12	1525	130	111	164	72.0	90.8	80.3	120	91.7	99.1	95.2
Going My Way	30	1950	521	0	441	93.2	78.9	85.4	46	0.0	-	-
合計（平均）	206	35216	2932	973	2659	90.6	82.2	86.2	1072	78.5	86.4	82.2

表 2 分離抽出のみの性能
Table 2 Experimental results of the separation.

	ポスト抽出性能（%）				コメント抽出性能（%）			
ブログ名	抽出数	適合率	再現率	F 値	抽出数	適合率	再現率	F 値
100SHIKI	153	100.0	100.0	100.0	12	100.0	100.0	100.0
Engadget Japanese	200	100.0	100.0	100.0	36	100.0	100.0	100.0
ネタフル	455	99.1	100.0	99.6	0	-	0.0	-
404 Blog Not Found	993	100.0	98.5	99.3	765	98.0	100.0	99.0
IDEA*IDEA	121	100.0	100.0	100.0	6	100.0	100.0	100.0
My Life Between...	212	78.3	100.0	87.8	0	-	0.0	-
TechCrunch Japan	182	100.0	100.0	100.0	8	100.0	100.0	100.0
Life is beautiful	130	100.0	100.0	100.0	111	100.0	100.0	100.0
Going My Way	521	100.0	100.0	100.0	0	-	-	-

は，コンテンツ自動抽出が不完全であっても，その範囲内でかなり正確に分離できている事がわかる．

4.4 考察及び今後の課題

提案手法の性能は，コンテンツ自動抽出手法の性能に依存している．ポストの適合率に比べてコメントの適合率が低くなる傾向があるのは，ポストが完全に抽出されない事，不要部分を誤って抽出する事が影響している．そのため，提案手法の性能を改善するためには，コンテンツ自動抽出手法の性能を改善する必要がある．

ブログ記事集合に構造が乱れた HTML（Valid でない HTML）で構成されるブログ記事が含まれている場合，位置ラベルの付与が適切に行われない場合がある．4.3 小節の実験で「404 Blog Not Found」が理想的な結果とならなかった原因は，そのようなブログ記事が 1 記事含まれていたからである．この問題は，HTML を整形するツール（たとえば「TIDY[⋆1]」）を使用する事により，解決できると考える．

提案手法は，ブログ記事集合にコメントが付いていない記事が含まれる事を前提としている．そのため，全ての記事にコメントが付いていると，コメントもポストとして抽出してし

⋆1 http://www.w3.org/People/Raggett/tidy/

まう．この問題は，3.3 小節で述べたとおり，フィードを用いてブログ記事を収集する事で解決できるが，収集方法に依存せずに解決する事も望まれる．コメント部分の位置ラベルはどのブログ記事集合でも似ている傾向がある事（位置ラベルに「comment」を含むなど）に着目し，コメントが 1 件も抽出できない場合は，他のブログ記事集合でコメントに分類された位置ラベルを加味する事により，適切に抽出できる可能性が高い．

筆者らは，ブログ記事集合において，ポストは全てのブログ記事に出現し，出現位置も似ている傾向がある事に着目し提案手法を考案した．位置を表現する方法としてブロックレベル要素の要素識別子を用いる方法を提案したが，ポストとコメントの位置を DOM ツリーで確認すると，それぞれ別のサブツリーに属している傾向がある．従って，DOM ツリーの構造パターンを獲得する事により，要素識別子に頼らず位置ラベルを付与できる可能性がある．この方法では，スタイルシートを使用せずにデザインしているブログサイトにも柔軟に適用できると考える．

5. 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，ブログからポストとコメントを分離抽出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，人手で作成した正解データを必要としないため，非常に小さな労力でブログ記事からポストとコメントを抽出する事ができる．

提案手法の分離抽出性能は，コンテンツ抽出性能に依存しており，コンテンツ抽出性能の改善のみで分離抽出性能が改善できる事が示唆された．また，対象とするブログ記事集合の全ての記事にコメントが付いている場合，ブログ記事集合に様々なブログサイトが混合している場合は適用が難しいという問題点があるが，収集方法の改善で容易に回避できると考えている．

今後，要素識別子が適切に付与されていないブログ記事集合にも適用できるよう，より柔軟に位置ラベルを決定する方法を検討する．また，抽出ルールの自動獲得，ルール自動選択による抽出を検討し，より高速に分離抽出する手法を検討する．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[*1]，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 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/

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，http://www.soumu.go.jp/iicp/chousakenkyu/data/research/survey/telecom/2009/2009-02.pdf (cited 2009-10-23) (2008).
2) 風間一洋，今田美幸，柏木啓一郎：ブログ空間における情報伝播ネットワークの抽出と分析，Web とデータベースに関するフォーラム（WebDB Forum）2008 (2008).
3) 藤村　滋，戸田浩之，松浦由美子，片岡良治：局所構造を考慮したブログネットワークの分析，Web とデータベースに関するフォーラム（WebDB Forum）2008 (2008).
4) 中島伸介，稲垣陽一，草野奉章：高信頼性情報の提示を目指した熟知度に基づくブログランキング方式の提案，日本データベース学会論文誌， Vol.7, No.1, pp.257–262 (2008).
5) Bhattarai, A., Rus, V. and Dasgupta, D.: Characterizing comment spam in the blogosphere through content analysis, *Computational Intelligence in Cyber Security, 2009. CICS '09. IEEE Symposium on*, pp.37–44 (2009).
6) Hu, M., Sun, A. and Lim, E.: Comments-oriented document summarization: understanding documents with readers' feedback, *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, Singapore, Singapore, ACM, pp.291–298 (2008).
7) Hu, M., Sun, A. and Lim, E.: Comments-oriented blog summarization by sentence extraction, *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, Lisbon, Portugal, ACM, pp.901–904 (2007).
8) Li, B., Xu, S. and Zhang, J.: Enhancing clustering blog documents by utilizing author/reader comments, *Proceedings of the 45th annual southeast regional conference*, Winston-Salem, North Carolina, ACM, pp.94–99 (2007).
9) 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).
10) 澤菜津美，森嶋厚行，杉本重雄，北川博之：HTML ラッパ自動構築手法の提案，日本データベース学会論文誌， Vol.7, No.1, pp.263–268 (2008).
11) Cao, D., Liao, X., Xu, H. and Bai, S.: Blog Post and Comment Extraction Using Information Quantity of Web Format, *Information Retrieval Technology*, pp.298–309 (2008).
12) 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).
13) 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).