サイト内におけるページ間の共通構造を基にしたブログ判定システム

石塚　拓也† 　手塚 太郎† 　木村 文則† 　前田 亮†

†立命館大学　情報理工学部 〒525-8577 滋賀県草津市野路東 1-1-1

E-mail: †cm000069@is.ritsumei.ac.jp, {tezuka, amaeda }@media.ritsumei.ac.jp fkimura@is.ritsumei.ac.jp

あらまし ウェブからブログのみを的確に収集するには，収集するページがブログかどうか判定する必要がある．しかし，ページ単体の特徴では，掲示板など似たような特徴を持つページが存在するため，機械的に判定するのは困難である．そこで，ページ単体ではなく，ページの集合体であるサイト毎の情報を用いてブログ判定を行う手法を提案する．

キーワード Web コンテンツ解析，ブログ，Web 情報の収集・分析

Weblogs Judgment System using Common Structures among Web Pages in a Site

Takuya ISHIZUKA† Taro TEZUKA† Fuminori KIMURA† and Akira MAEDA†

† College of Information Science and Engineering,
Ritsumeikan University 1-1-1 Nojihigashi, Kusatsu-shi, Shiga, 525-8577 Japan

E-mail: †cm000069@is.ritsumei.ac.jp, {tezuka, amaeda }@media.ritsumei.ac.jp fkimura@is.ritsumei.ac.jp

Abstract In order to efficiently collect blogs using a web crawler, it is necessary to distinguish blogs from other types of web content. The task is difficult since there are many web pages with structures similar to that of blogs, for example BBS (bulletin board systems). We therefore propose a method that uses the common structures of pages contained in each web site, and distinguish blogs from other types of web sites.

Keyword contents analysis, Weblog, crawling and analysis

1. はじめに

近年，一般家庭におけるインターネットの普及にともない，ブログ（Weblog）や掲示板（BBS），ソーシャルネットワークシステム（SNS）など多くの人々が自ら情報を発信する機会が増えている．こうした状況を背景に，ユーザが主体的に発信した「生の声」をこれらの情報源から取り出し，企業活動や研究などに利用しようとする試みがみられる．

ユーザが主体的に情報を発信している情報源のなかで，近年注目されているものとしてブログがある．ブログは通常の Web ページと比べ，速報性，リアルタイム性のある新鮮な情報が発信されている点に特徴がある．また，ここ数年においてブログが爆発的に普及してきたことによって，トレンド分析や評判分析，実世界の動向との相関分析など数多くのブログマイニング技術が研究されている[1][2].

これらの技術を活かすためには，インターネット上に存在するブログのデータをリアルタイムかつ自動的に収集しなくてはならない．従来ブログを収集するには，RSS[3][4][5] などのメタデータや，ping.bloggers.jp[6]などをはじめとする，XML-RPC を使用した ping[7]による blog 更新通知サービスなどのツールを利用することで行われてきた．しかし，ユーザの多様化により，現在は RSS などのメタデータの構築や ping による更新情報の配信を行わないブログも多数出てきている．また，これらの技術の普及にともない，ニュースサイトなどブログ以外にもこれらのシステムを利用したサイトが出現している．さらにはこれらのシステムを悪用したスパムサイトまでもが出現してきている．

このため，多くのブログを網羅的に収集するには，特定のツールやメタデータなどに依存しない手法が必要である．そのような手法として，HTML 文書の解析に基づいた手法が挙げられる．しかし，単一の Web ページ内だけを解析した結果から得られた特徴では，掲示板などブログと似たような特徴を持つページが存在するため，機械的に判定するのは困難である．そこで，本研究ではある URL で表されるディレクトリをサイトのトップディレクトリとし，そのトップディレクトリと同一階層，および下階層に含まれる Web ページが

そのサイト内のページであると定義する．そして，同一サイト内の Web ページにおける HTML 文書の構造の一致を用いることによって，ページ単体ではなく，ページの集合体であるサイト毎にブログ判定を行う手法を提案する．

2. 関連研究

南野ら[8]は，HTML 文書を解析し，“Web ページから，ある制約を満たすエントリを切り出すことができるかどうか”をチェックすることで，その Web ページがブログであるかどうかの判定を行っている．Web ページからエントリの抽出を行うために，出現した日付表現を用いてHTML文書中におけるエントリの開始位置と終了位置を決定する．

エントリの開始位置については，日付表現はエントリの上部にあると仮定することによって行われ，終了位置については，次のエントリの開始位置によって決定される．しかし，この手法ではエントリの終了位置の判定に次のエントリの情報を用いるため，エントリが 1 つしかない Web ページについては解析することができない．また，Web ページ単体のみの情報を用いて解析するため，エントリの本文中に出現した日付表現の誤認識などの問題が起こっている．

そこで本研究では，取得した Web ページ単体の情報だけでなく，Web ページをサイトごとに分類し，同一サイト内の Web ページ全ての情報を用いることによって，サイト内から“ある制約を満たすエントリ”が出現するかどうかを調べる．

3. Web ページの収集

本研究では，ブログであるかどうかの判断は，同一サイト内の Web ページにおける HTML 文書の構造の一致を基に行う．そのため，Web ページを取得する際に，サイト毎に取得する必要がある．Web ページをサイト毎に取得するためには，サイトのトップディレクトリを決定し，そのトップディレクトリと同一階層もしくは下階層の Web ページをクローリングすることによって可能である．

サイトのトップディレクトリを決定する手法としては以下の３つを使用する．

1. ping によるブログ更新通知サービスから送られてくる URL のディレクトリ部分をサイトのトップディレクトリとする

2. RSS などのメタデータ内に含まれるサイトのトップページの URL におけるディレクトリ部分をサイトのトップディレクトリとする

3. URL のホストをトップディレクトリとする

図 1 にシステムの概要図を示す． ping.bloggers.jp および RSS，URL のホスト部分を用いてサイトのトップディレクトリを抽出し，Web ページの収集モジュールを用いることによって，サイト内に含まれる Web ページのテキストデータを収集し，それらに対してブログ判定モジュールを用いることによって，ブログのエントリの投稿日もしくは更新日を表す日付表現のみを判別し取得する．ブログのエントリの投稿日もしくは更新日を表す日付表現については，次節にて説明する．

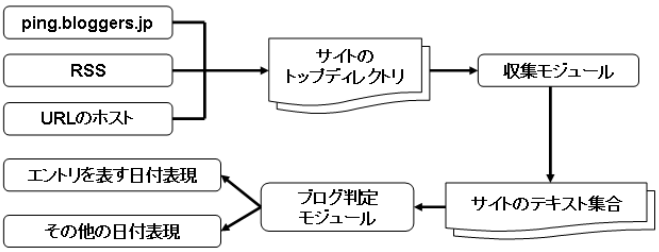

図 1：システムの概要図

4. Web ページのブログ判定

本研究では，ブログはエントリを含む Web ページであると定義している．ある Web ページがブログであるかどうかを解析せずに直接判断することは難しいが，Web ページがエントリを含むかどうかについて解析し，含んでいた場合はそれをブログと判断することは可能である．ブログの性質として以下を仮定する．

- 各エントリ内に，エントリの書かれた日付を表す日付表現が必ず含まれる．
- 同一サイト内のブログページにおける全てのエントリに対して，各日付表現にかかるタグの種類は一定である．
- 同一サイト内のブログページにおける全てのエントリに対して，各日付表現のフォーマットは一定である．
  ex．) “2010/1/4”と”2010-1-4”は別のフォーマットとみなされる

エントリの出現部分とは，エントリの更新日もしくは投稿日を表す日付表現が出現する部分である．今後，エントリの更新日もしくは投稿日を表す日付表現のことを，“ターゲット日付表現”と呼ぶ．

本実験では，これらの条件を用いて，サイト内に出現する全ての日付表現から，ターゲット日付表現を取得する．

4．1　前処理

取得した Web ページに対して HTML Tidy[9]を適用し，HTML 文書を well-formed XML 文書にする．この処理により，開始タグと終了タグのバランスが取れていることが保証される．よって，以降の処理では，全てのエントリにおけるタグの係り方が一定であるという制約を用いることができる．

4．2　日付表現の抽出

次に，前処理を行った HTML 文書から，日付表現を抽出する．

日付の表記法には様々なものがある．区切り文字には，「年」「月」「日」，ハイフン，スラッシュなどが使用されることもあれば，月が英語名で記述されているものもある．これら一般に使われている数種類のフォーマットに分類し，それぞれを表す正規表現と入力 HTML 文書でパターンマッチを行うことで，日付候補箇所を抽出する．表 1 に正規表現にマッチする日付表現の一部を示す．

また，ターゲット日付表現は以下の条件を満たすと仮定し，それ以外の日付表現を除外する．

- ターゲット日付表現は head，style，script タグ内には出現しない
- ターゲット日付表現は 4 階層目以下のタグ内で出現する

まず，HTML 文書において script タグが使われるのは，flash やユーザサイドで動くプログラムを表記する場合である．ターゲット日付表現は著者がサーバにエントリを投稿した時点で決まるので，script タグ内で出現した日付表現は，ターゲット日付表現ではないと判断することができる．また，エントリが画面上に表示されることから，ページのメタデータなどを記述する head タグ内および，CSS を記述する style タグ内に出現した日付表現も除外する．

ターゲット日付表現はブラウザによってレンダリングされて画面に表示されることから，html タグ内および body タグ内に含まれる．また，ヘッダやフッタおよびサイドカラムとエントリを分けられて表示されることから，各エントリもしくはエントリ全体を表すタグが 1 つ出現する．そして，日付表現はエントリのフッタもしくはタイトル部分に出現するため，日付表現の出現部分をエントリの本文などの他の記述と分けるためのタグが 1 つ出現する．これらの理由により，ターゲット日付表現は，4 階層目以下のタグ内で出現するものとする．図 2 に日付表現の出現部分の構造の例を示す．また，図 3 に HTML タグの階層構造について示す．

表 1 ：日付フォーマットの例

2004 年 3 月 5 日	2004.3.5	2004/3/5
2004-3-5	2004 03 05	5 Mar. 2004
5 March 2004.	5-March-2004	3. 5 2004

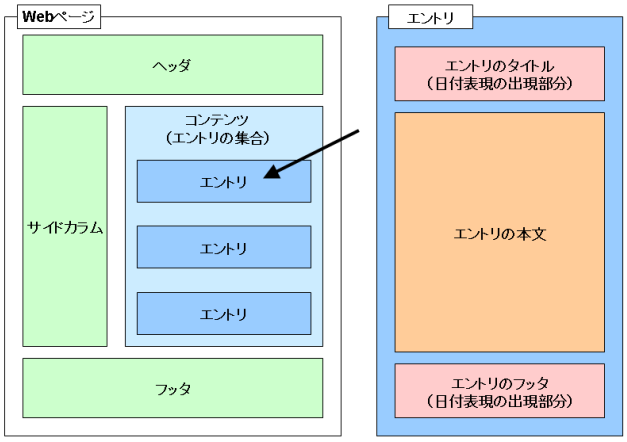

図 2：日付表現の出現部分の構造の例

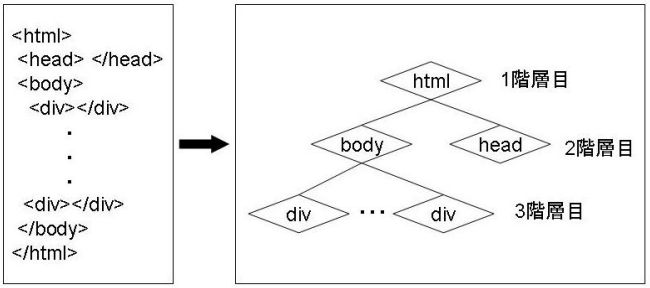

図 3：HTML タグの階層構造

4.3　ブログ判定システム

ある Web ページがブログであるかどうかの判定は，ターゲット日付表現が出現するかどうかを調べることによって，判定することが可能である．

まずサイト内で出現した全ての日付表現を本節で定義する条件に基づいて分類する．これらの分類された日付表現の集合のうち，ある条件に当てはまる日付表現の集合はターゲット日付表現の集合であるとし，フィルタリングによってターゲット日付表現の集合のみを採択する．

4．3．1 日付表現の分類

ブラウザでエントリを表示したときに，各エントリの日付表現（ターゲット日付表現）はすべてのエントリにおいて同様に表示される．そこで，前節の手法に

よって抽出された日付表現を以下の２つの条件を用いて分類することによって，すべてのターゲット日付表現を同一種類の日付表現として，１つの集合にまとめる事ができる．

- タグの係り方が同一である
- 日付表現のフォーマットが同一である

今後，この 2 つの条件に基づいた分類を“基本的な分類”および“分類 A”とし，これらの条件に基づいて分類された日付表現の集合を“基本的な分類による日付表現の集合”もしくは単に“日付表現の集合”とする．また，基本的な分類による日付表現の集合のうち，ターゲット日付表現によって構成されている日付表現の集合のことを“ターゲット日付表現の集合”とする．

タグの係り方で分類する際に，インライン要素のタグは影響しないものとする．HTML の要素はインライン要素とブロック要素の 2 種類に分けることができる．HTML データをブラウザで変換して画面上に表示する際に，表示範囲や表示位置などの構造的な部分に影響する要素がブロック要素であり，色やフォント，リンクなどのように，構造的な部分には影響しない要素がインライン要素である．ブログで日付表現を表示する際に，表示位置などの構造的な部分では一致するが，構造的な部分以外で違いが出てくる可能性がある．例えば，リンクを張るのに使用する要素のタグである <a>タグにおいて，その日付のアーカイブが存在する場合はリンクが張られ，存在しない場合にはリンクが張られないなど，このような場合が考えられる．よって表 2 に示す要素のタグについては，タグ構造による分類に影響しないものとする．

また，フィルタリングによってターゲット日付表現の集合のみを採択するために，基本的な分類による日付表現の集合をさらに,以下の条件によって分類する．

分類B　出現するページが同一である
（出現するページが同一である日付表現の分類）

分類C　タグの係り方だけでなく，出現するタグの位置（兄弟構造）も同一である
（兄弟構造が同一である日付表現の分類）

分類D　日付表現のフォーマットだけでなく，テキストレベルで同一である
（テキストが同一である日付表現の分類）

これらの分類は基本的な分類による日付表現をさらに細かく分類したものであるため，基本的な分類における分類の条件もそれぞれ含まれるものとする．図 4 に各手法による日付表現の分類のイメージを示す．

表 2：除外する HTML のインライン要素

a	span	big	small
b	em	i	font
strong	u		

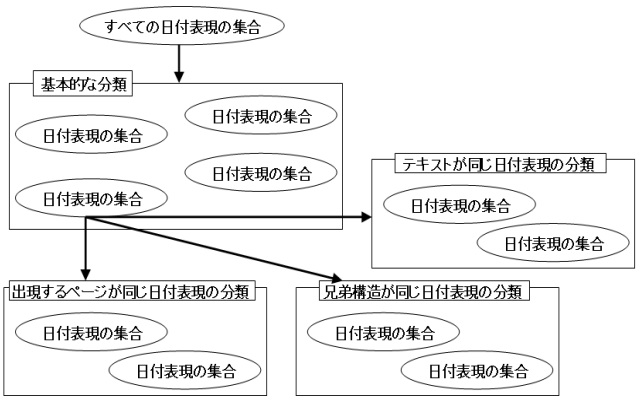

図 4：各手法による日付表現の分類のイメージ

4．3．2 日付表現のタイプによるフィルタリング

ターゲット日付表現の集合は以下の条件を満たすと仮定し，それ以外の日付表現の集合を除外する．

条件1　サイト内の全ページのうち 3 割以上のページで出現する

条件2　兄弟構造およびテキストが同一である日付表現の集合において，その集合に含まれる日付表現が出現するページはサイト内全てのページの 8 割未満である

条件3　出現するページが同一である日付表現の集合において，テキストが同一である日付表現が 3 つ以上出現するページは，サイト内全てのページの 6 割未満である

条件4　出現するページが同一である日付表現の集合において，それに含まれる日付表現の数は 40 個未満である

条件5　9 割以上は過去の日付を表す

条件6　出現するページが同一である日付表現の集合において，それに含まれる日付表現が 1 個だけの集合の日付表現を集め，その集合に対し，テキストが同一である日付表現が 30 個以上である集合全てに含まれる日付表現の数は 4 割

未満である

条件7　ターゲット日付表現は一定の期間ではなく，特定の日を表す

条件8　１つのページ内で日付表現が隣接して出現することはない

図 5 にフィルタリングに使用する日付表現の集合についての関係の図を示す．円で囲まれた部分は，そこに書かれた分類手法で分類された日付表現の集合を表していて，四角で囲まれた部分は，そこに書かれた条件でフィルタリングを行うことを示している．また，実線は日付表現の分類の流れを示し，点線は四角に囲まれた部分の内部に書かれている条件でフィルタリングを行う際に使用する，日付表現の分類の手法との関係を表している．

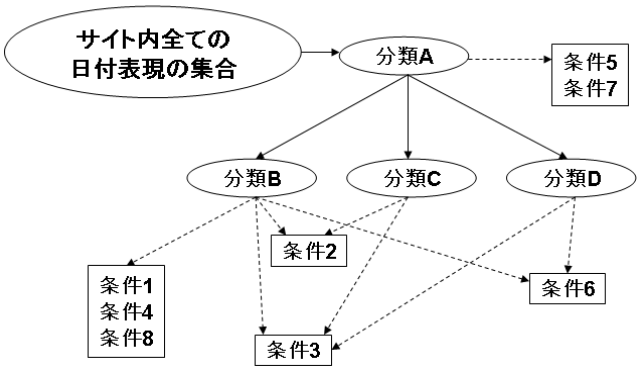

図 5：フィルタリングに使用する日付表現の集合

4．3．2．1 条件 1 によるフィルタリング

ブログにはアーカイブや個別ページとして非常に多くのページが生成されるという特徴がある．エントリはそれらのページ全てに含まれるため，ターゲット日付表現は多くのページで出現する．よって，出現するページ数が少なく，エントリの本文の中で出てきた日付表現など，突発的に出現した日付表現はエントリの日付表現ではないと判断することができる．

4．3．2．2 条件 2 によるフィルタリング

ターゲット日付表現は，そのエントリの投稿日もしくは更新日であるため，エントリ毎に日付が変わる．このことから，基本的な分類による日付表現の集合において，同一の集合内の日付表現であっても，出現する箇所によって日付表現のテキストは違うということになる．このことから，出現するページ全てにおいて，テキストが同一である日付表現はターゲット日付表現でないということが分かる．このような日付表現の例として，サイトの作成日などがある．

4．3．2．3 条件 3 によるフィルタリング

ブログは掲示板やチャットと違い，エントリを投稿するユーザ数が限定されているため，1 日の間に投稿されるエントリは比較的に少ない．さらに，ブログには個別エントリを表示するページが存在するため，日付が同一であるエントリが複数回出現するページは少ない．それを利用することによって，多くのページにおいて，同一の日付が何度も出現するような日付表現はターゲット日付表現でないと判断できる．

4．3．2．4 条件 4 によるフィルタリング

ブログでは掲示板などとは違い，１つのエントリに書かれる内容が長く，それぞれのエントリ間に直接的な関係性は無いため，１つのページに表示するエントリの数が多い場合は複数のページに分けて表示する場合がほとんどである．そのことから，１つのページに多く出現する日付表現の集合は，ターゲット日付表現でないと判断できる．

4．3．2．5 条件 5 によるフィルタリング

ブログにおいて，ターゲット日付表現は基本的に過去のものである．システムによってはブログでも未来の日付表現のエントリを投稿することは可能であるが，そういったシステムのブログでも実際にその日時が来るまで非表示の設定にできる．またブログは基本的に現在起きたことなどを記す物なので，未来の日付であるエントリが何度も出現することは無い．これを利用することによって，未来の日付表現が多く出現するイベントの告知などに出現する日付表現を除外することができる．

4．3．2．6 条件 6 によるフィルタリング

ブログでは，個別エントリのページを集めて，それらのページに含まれる日付表現の中で日付が同一である日付表現がいくつ存在するかを調べることによって，1 日の間に何回エントリが投稿されたかを調べることができる．また，ニュースサイトも基本的に１つのページに付き１つのエントリが書かれているため，同一手法で１日の間に何回エントリが投稿されたかを調べることができる．ブログは基本的に個人が管理しているため，ニュースサイトと比べて１日に投稿されるエントリの数は少ない．そのことから，１日に多く投稿されている日付表現の数が多かった場合は，それはブログではないと判断することができる．

4．3．2．7 条件 7 によるフィルタリング

ブログのエントリに出現する日付表現は投稿日もしくは更新日を表すものである．よって，特定の日付ではなく，一定の期間を表す日付表現はターゲット日付表現ではないと言える．

4．3．2．8 条件 8 によるフィルタリング

ブログのエントリには本文やタイトルが含まれるため，複数のエントリの日付表現が隣接して出現することはない．そこで，アーカイブのリストなど，複数の日付表現が隣接して出現した場合はターゲット日付表現ではないと判断できる．

5. 評価実験

本手法に基づいて作られたシステムの評価実験を行う．

5．1 実験に用いたデータ

本実験は，ターゲット日付表現の集合を正例，それ以外の日付表現の集合を負例とし，各日付表現の集合に対してフィルタリングを行うことによって，各サイト内に出現した正例および負例である日付表現の集合が，採択もしくは棄却のどちらになったかについて調べる．

ここで，正例については，サイト内に複数出現する正例のうち，フィルタリングの結果少なくとも一つ採択できた場合は“正例を採択した”，すべて棄却した場合は“正例を棄却した”したとする．

また，負例については，サイト内に複数出現する負例のうち，フィルタリングの結果少なくとも一つ採択した場合は“負例を採択した”，すべて棄却した場合は“負例を棄却した”とする．

ブログである Web ページを含むサイトの集合として，ping.bloggers.jp を用いて取得したサイトの中からランダムに選んだ 47 サイト，42,721 ページを使用する．これらのサイト内には，正例だけでなく負例も出現するので，正例および負例の両方において，それぞれ採択もしくは棄却のどちらになったかについて調べる．

また，正例を含まないサイトとして，南野ら[8]の研究でブログであると誤認識されてしまった BBS やイベント紹介などの内容によって構成されている Web ページを含むサイトの中からランダムに選んだ 30 サイト，58,141 ページを使用する．これらのサイトには負例のみが出現するため，負例について採択もしくは棄却のどちらであったかを調べる．本実験で用いた正例を含まないサイトの種類を以下に示す．

- 掲示板
- イベントの案内
- 更新情報
- メールマガジン
- Amazon.com のレビュー
- ニュースリリース

5．2 実験結果

正例を含む 47 サイトのうち，フィルタリングによって正例のみを残せたサイトは 39 であった．このことから全体的な認識率は約 83%である．実験の結果を表 3 に示す．

負例を採択してしまったサイトは 47 サイトのうち 4 サイトであった．負例を採択してしまった原因を表 4 に示す．

正例を棄却してしまったサイトは 47 サイトのうち 5 サイトであった．正例を棄却してしまった原因を表 5 に示す．

正例を含まない 30 サイトのうち，フィルタリングによってすべての負例を除外できたサイトは 25 であった．このことから誤認率は約 83%である．実験の結果を表 6 に示す．

負例を採択してしまったサイトは 30 サイトのうち 5 サイトであった．負例を採択してしまったサイトを表 7 に示す．また，負例を採択してしまった原因を表 8 に示す．

表 3：正例を含むデータにおける実験結果

	負例を棄却した	負例を採択した
正例を採択した	39（83%）	3（6%）
正例を棄却した	4（8%）	1（2%）

表 4：負例を採択してしまった原因

原因	データ数
エントリ内に出てくるアマゾンの広告	1
エントリのコメントの日付表現	2
日付アーカイブの日付データ	1

表 5：正例を棄却してしまった原因

原因	データ数
日付表現不足	1
日付表現に「年」が含まれていない	1
ターゲット日付表現が無い	1
ブログであるファイル数が少ない	1
元データのファイル数が足りていない	1

表 6：正例を含まないデータにおける実験結果

負例を棄却した	負例を採択した
25（83%）	5（16%）

表 7：負例を採択してしまったサイトの種類

データの種類	データ数
Wikipedia	1
ニュースサイト	2
掲示板	2

表 8：負例を採択してしまった原因

原因	データ数
Wikipedia のエントリの最終更新時間	1
ニュースサイトの 1 日あたりの エントリの数が少ない	1
元データのファイル数が足りていない	1
掲示板の投稿時間	2

6. 考察

正例を含むサイト集合に対する実験の認識率は 83%という結果が出た．また，正例を含まないサイト集合に対するデータの認識率は 83%であった．

6．1 負例を採択してしまった原因

負例を採択してしまった原因のうち，エントリのコメントを含むデータおよび，掲示板の投稿時間については HTML タグの input 要素で type 属性が text であるタグ，input 要素の type 属性が submit であるタグ，textarea 要素のタグが出現するページのみで出現する日付表現を弾くことで，除去することができると予想される．この方法でコメントの日付表現のみを除去できるのと予想されるのは，もしエントリが 1 ページに 1 つしか表示されないブログであったとしても，アーカイブページではコメントを入力するフォームが存在しないためである．ただし，コメント投稿の機能を利用不可能にしているブログではこのようなフォームは表示されないため，一部のブログでしか適応できないと思われる．

また，エントリ内に何度も出てくるアマゾンの広告や日別アーカイブの日付データ，Wikipedia のエントリの最終更新時間は，HTML の構造を解析し，近くにエントリのタイトルおよび本文と思われる文章データが出現するかどうかを調べることによって，除去することが可能であると予想される．

ニュースサイトの 1 日あたりのエントリ数が不足については，フィルタの閾値を厳しくすることによって除外することが可能である．しかし，その場合は他の正例のデータも除外してしまう可能性があることについて注意が必要である．

元データのファイル数が足りていないものについては，データ数が少ないデータについてはブログ判定をせずに，判定不能という結果を出すという解決策がある．こういったデータについては，正例を逆に弾いてしまうという場合も大いに考えられるため，判定した結果を用いないほうがよいと思われる．

また，正例と負例両方とも採択してしまったデータについては，各タイプの日付表現の出現回数や出現するファイル数などのデータを元に，正例である可能性が高いと思われるものを選ぶことも可能であると思われる．

6．2 正例を棄却してしまった原因

正例を棄却してしまった原因のうち，日付表現不足については，日付表現の種類を増やすことによって解決することができる．ただし，正規表現を増やすと処理時間が長くなるため，その点に注意が必要である．

日付表現に年が含まれてないデータについては，南野ら[8]の研究において，日付表現の年を補完するというものがある．これは，日付表現に年が不足した場合，その日付表現を表す年のデータはそのページの上部に出現するものであるという考えで，これを利用することによって解決することができると思われる．

エントリに日付表現が無いものや，元データのファイル数が足りていないものについては，本研究の性質上解決不可能である．

全体の中でブログであるファイル数が少ないものについては，フィルタの閾値を変更することによって取得することが可能である．しかし，閾値を甘くすることによって他の不要なデータも取得してしまう可能性がある点について注意が必要である．また，すべてのタイプの日付表現のデータを負例であるとしてしまったサイトについて，そのサイトの 1 階層下ごとに分類に分けて，再度ブログ判定を行うことによって，この問題を解決することが可能であると予測される．また，これを繰り返すことによって，最終的にはページ単位の判定を行うことになる．この場合は南野ら[8]の研究における手法を利用することによって判定が可能である．また，Web ページをサイト毎にまとめる際に，最初に決定したトップディレクトリがホスティングサービスによるものであった場合など，１つのサイト内に複数のサイトが含まれることが考えられる．そのような場合についても同一手法を用いることによって，URL のホストをトップディレクトリとした場合，そのディレクトリのトップページがブログでないと判断されたならば，元のトップディレクトリから階層が

１つ下のそれぞれのディレクトリをトップディレクトリとして，再度ブログ判定を行うという手法によって解決することができると思われる．

7. まとめ

本論文ではページ間の構造を用いてサイトがブログであるかどうかの判定を行う手法について述べ，その手法を用いた実験を行った結果について述べた．

多くのデータにおいて，ターゲット日付表現の集合のみを取得することに成功した．また，従来手法と比べて，１ページにエントリが１つしか出現しない場合についても判定が可能である．しかし，すべてのデータではなく，考察で述べたとおりまだ多くの解決策が考えうる．また，南野ら[8]の手法と組み合わせることによって，更なる精度の向上も可能であると思われる．

謝　辞

本研究の一部は文部科学省私立大学戦略的研究基盤形成支援事業「芸術・文化分野の資料デジタル化と活用を軸とした研究資源共有化研究」の支援を受けている．

参　考　文　献

[1] 奥村　学．“blog マイニング――インターネット上のトレンド，意見分析を目指して――，”．人工知能誌，vol.21，no.4，pp.424-429，2006.

[2] 奥村学．ブログマイニング技術の最新動向．電子情報通信学会誌，Vol. 91，No. 12，pp. 1054-1059，2008.

[3] RDF Site Summary (RSS) 1.0 . http://web.resource.org/rss/1.0/.

[4] RSS 0.92 . http://backend.userland.com/rss092.

[5] RSS 2.0 Specification . http://www.rssboard.org/rss-specification.

[6] Daiji Hirata . ping.bloggers.jp. http://ping.bloggers.jp/.

[7] Dave Winer．Weblogs.com xml-rpc interface．http://www.xmlrpc.com/weblogsCom，2001

[8] 南野朋之，鈴木泰裕，藤木稔明，奥村学．Blog の自動収集と監視．情報処理学会研究報告，2004-NL-160，pp.129-136，2004.

[9] Dava Raggett. Clean up your web pages with html tidy. http://www.w3.org/People/Reggett/tidy/.

[10] Tomoyuki NANNO, Suguru SAITO, and Manabu OKUMURA . Structuring web pages based on repetition of elements. In Second International Workshop on Web Document Analysis(WDA2003), 2003

[11] 奥村 学，南野 朋之，藤木 稔明，鈴木 泰裕．blog ページの自動収集と監視に基づくテキストマイニング．第 6 回人工知能学会セマンティック Web とオントロジー研究会，2004.

[12] 灘本 明代，荒牧 英治，阿辺川 武，村上 陽平．Wikipedia エントリとブログサイトの対応付けのための特定トピックのブログサイト検索．電子情報通信学会データ工学ワークショップ，2008.

[13] 黒田 晋矢，福田 直樹，石川 博．Blog 空間探索のための Blog データベースの設計．電子情報通信学会データ工学ワークショップ，2007.