修 士 論 文

ウェブページからのサイト情報・作成者情報の抽出

北陸先端科学技術大学院大学
情報科学研究科情報科学専攻

堀 達也

2015 年 9 月

修 士 論 文

ウェブページからのサイト情報・作成者情報の抽出

指導教員　白井清昭 准教授

審査委員主査　白井清昭 准教授
審査委員　池田心 准教授
審査委員　長谷川忍 准教授

北陸先端科学技術大学院大学
情報科学研究科情報科学専攻

1310067 堀 達也

提出年月: 2015 年 8 月

概 要

ウェブには様々な情報が存在し, その量は膨大である. 我々は, ウェブから知りたい情報を検索することができる. しかし, ウェブには正しい情報だけが存在するのではなく, 虚偽の情報も存在する. そのため, ユーザが虚偽の情報を正しい情報として誤って認識してしまう可能性がある. このような事態を防ぐため, ユーザは検索した情報が正しい情報であるか否かを判断しなければならない. このとき, 情報の正しさを判断する助けになるのが, ウェブサイトに関する情報や, そのウェブサイトの作成者に関する情報である. 例えば, 病気について調べたいときには, ウェブ検索でヒットしたウェブサイトが病院の正式なホームページであるとわかれば, そのサイトは信頼性が高いと判断できる. 同様に, 検索によってブログ記事がヒットしたとき, そのブログの書き手が医者であることがわかれば, そのブログの内容の信頼性が高いと判断できる. そこで, 本研究ではウェブページからウェブサイト情報や作成者情報を自動抽出することを目的とする. ここで, 「ウェブサイト情報」(以下, 単にサイト情報と呼ぶ) とはウェブサイトやブログの内容を説明したテキスト, 「作成者情報」とはウェブページの作成者のプロフィール (年齢, 性別, 職業, 自己紹介など) について書かれたテキストであると定義する. また, ウェブサイトや作成者の情報が記述された別ページへのリンクが存在するときは, そのリンクを抽出する. なお, ウェブには様々なサイトが存在するが, ブログは形式がある程度決まっているため, サイト情報や作成者情報の抽出が比較的容易であると考えられる. そのため, 本研究では手始めにブログページを対象としてサイト情報・作成者情報の抽出を試みる.

関連研究として, 主に情報発信者名や著者名の抽出を試みた先行研究がいくつかある. 百瀬らは, ウェブページのレイアウト情報を利用し, 情報発信者名を抽出している. Kato らは, テキストの特性や, DOM の木構造における深さなどを手掛かりに, 情報発信者名を抽出している. Giuffrida らは, ウェブページの空間特性や, テキストの言語特性を利用し, 科学論文から著者名等のメタデータを抽出している. これらの研究との違いとして, 本研究では作成者の名前だけでなく, ウェブサイトの説明文 (サイト情報) や, 作成者に関する年齢, 性別, 職業, 自己紹介文など (作成者情報) をウェブページから取得する点に特徴がある. すなわち, 先行研究と比べて, ウェブページの信頼性を判断するために有効な情報の抽出に焦点を当てている.

本研究では, HTML ファイルにおける Document Object Model (DOM) の個々のノードに対し, そのノードがサイト情報や作成者情報を含むか否かを判定する. 上記の判定を行う分類器は教師あり機械学習によって獲得する. 機械学習アルゴリズムは Support Vector Machine (SVM) を用いる. 機械学習に用いる素性として, DOM ノードのタグ名, id, class の属性値, テキスト長, 自立語, DOM ノードがブログタイトルの近くに出現するか否か, サイト情報を示唆するキーワード, サイト情報へのリンクを示唆するキーワード, サイトの説明文に頻出する n-gram を使用する. また, 本研究の問題設定では, 正例 (サイト情報や作成者情報を含む DOM ノード) よりも負例 (情報を含まない DOM ノード) の方が圧倒

的に多い. そこで, 本研究では明らかに負例であると考えられる DOM ノードをあらかじめ除外し, 正例と負例のバランスを是正することで, 情報抽出の性能の向上を図った. この手続きを「負例のフィルタリング」と呼ぶ. 本研究では, まずコンテンツ領域のフィルタリングを実行する. ウェブページは, そのページの主な内容が記述されているコンテンツ領域と, 目次, 広告などを表示する非コンテンツ領域に分けることができる. サイト情報や作成者情報は非コンテンツ領域に配置されると考えられる. そこで, ウェブページのコンテンツ領域を負例とみなし, そこに含まれる DOM ノードを削除する. コンテンツ領域の検出には, Kato らの手法を用いる. 次に, テキスト長が 0 のノードのフィルタリングを実行する. テキスト長が 0 の DOM ノードには, 明らかにサイト情報, 作成者情報が含まれていない. そのため, このようなノードを負例とみなし, 削除する.

本研究の評価実験について述べる. ウェブから 500 件のブログページを取得し, 人手でサイト情報と作成者情報を付与した. これを訓練データとし, 10 分割交差検定によりサイト情報・作成者情報抽出の精度, 再現率, F 値を求めた. また, 提案手法との比較のために, 人手で作成した少数のルールに従って情報を抽出するベースラインシステムを構築した. ベースラインシステムの F 値は, サイト情報が 0.384, 作成者情報が 0.258 であったのに対し, 提案手法の F 値は, サイト情報が 0.585, 作成者情報が 0.675 であった. このことから, サイト情報, 作成者情報の抽出において, 提案手法はベースラインシステムより有効であることがわかった. さらに, 新たに別のブログの集合を取得し, これをテストデータとし, 同様の実験を行った. このテストデータにおけるサイト情報, 作成者情報の F 値も, 提案手法はベースラインシステムを上回った. さらに, 提案した素性の有効性の評価を行った. 実験データによって, 素性の有効性に違いが生じたため, 提案手法においてどの素性がサイト情報又は作成者情報の抽出に有効であるかを明確に確認することはできなかった. しかしながら, 全体的には各素性が F 値の向上に貢献することを確認した. さらに, 負例のフィルタリングの有効性の評価も行った. その結果, 負例のフィルタリングがサイト情報, 作成者情報の抽出精度の向上に貢献していることがわかった. 最後に, 情報の抽出に失敗した原因を考察した. 作成者以外のプロフィールを抽出してしまうこと, 負例のフィルタリングで正例を削除してしまうことなどの要因によってエラーが発生したことが判明した.

本研究の提案手法はベースラインシステムより高い評価値を示したため, サイト情報, 作成者情報の抽出には機械学習による手法が効果的であることが確認された. エラー分析の結果を踏まえ, サイト情報, 作成者情報の抽出精度をさらに向上させる手法を検討することが今後の課題である.

目 次

第1章　はじめに

1.1　研究の背景

ウェブには様々な情報が存在し, その量は膨大である. 我々はウェブ上でさまざまな情報を検索することができる. しかし, ウェブにはときには正しくない情報が公開されていることもある. そのため, ユーザが虚偽の情報を正しい情報として誤って認識してしまう可能性がある. このような事態を防ぐため, ユーザは検索した情報が正しい情報であるか否かを判断しなければならない. このとき, 情報の正しさを判断する助けになるのが, ウェブサイトに関する情報や, そのウェブサイトの作成者に関する情報である. 例えば, 病気について調べたいときには, ウェブ検索でヒットしたウェブサイトが病院の正式なホームページであるとわかれば, そのサイトは信頼性が高いと判断できる. 同様に, 検索によってブログ記事がヒットしたとき, そのブログの書き手が医者であることがわかれば, そのブログの内容の信頼性が高いと判断できる. ウェブサイトに関する情報や作成者に関する情報が書かれている場所は, ウェブページによって多様である. そのため, これらの情報を人手で素早く取得することは困難である.

1.2　研究の目的

本研究では, ウェブページからウェブサイト情報や作成者情報を自動抽出することを目的とする. ここで, 「ウェブサイト情報」(以下, 単にサイト情報と呼ぶ) とはウェブサイトやブログの内容を説明したテキスト, 「作成者情報」とはウェブページの作成者のプロフィール (年齢, 性別, 職業, 自己紹介など) について書かれたテキストと定義する. また, ウェブサイトや作成者の情報が記述された別ページへのリンクが存在するときは, そのリンクを抽出する. 将来的には, 抽出したサイト情報や作成者情報は, ユーザがウェブページの信頼性を判断する補助情報として, ウェブ検索エンジンで検索結果とともに掲示することを想定している. これにより, ウェブサイトのサイト情報, 作成者情報を簡易に確認することができるため, ウェブサイトの信頼性を判断する時間を短縮できると考えられる. なお, ウェブには様々なサイトが存在するが, ブログは形式がある程度決まっているため, サイト情報や作成者情報の抽出が比較的容易であると考えられる. そのため, 本研究では手始めにブログページを対象としてサイト情報・作成者情報の抽出を試みる.

1.3　本論文の構成

本論文は全 5 章から構成されている. 第 2 章では, 先行研究の概要および先行研究と本研究の違いについて説明する. 第 3 章では, 本研究で提案するサイト情報および作成者情報を抽出する手法について説明する. 第 4 章では, 提案手法の評価実験について報告し, その結果を考察する. また, エラー分析の結果についても報告する. 第 5 章では, 本論文のまとめと今後の課題について述べる.

第2章　関連研究

本章では, 本研究の関連研究について述べる. 2.1 節では, 情報発信者や著者をウェブページから抽出する手法, あるいは人々の意見をウェブページから抽出する手法 (オピニオンマイニング) に関する先行研究を紹介する. 2.2 節では, これらの関連研究と本研究の違いについて論じる.

2.1　ウェブページからの情報抽出

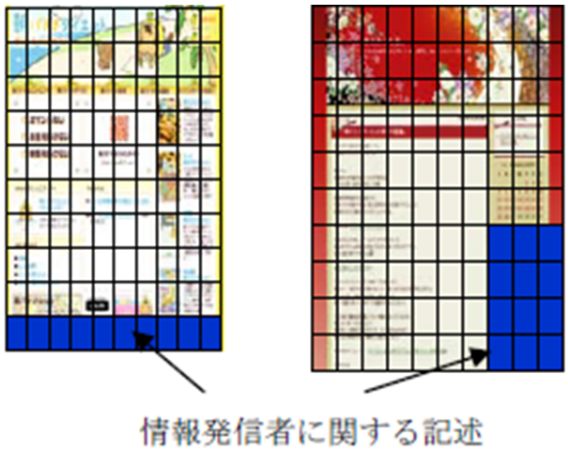

図 2.1: 情報発信者記述部分の例 [1]

百瀬らは, ウェブページのレイアウト情報を利用して, 二段階の手続きで情報発信者を抽出する手法を提案した [1]. 第一段階では, 図 2.1 のように, ウェブページを縦横のグリッドに分割し, グリッドのセルごとに発信者情報が含まれているか否かを判定し, そこに含まれる DOM ノードを抽出する. 発信者情報が含まれているか否かは, 機械学習を用いて判定する. 機械学習に用いる素性を以下に示す.

- グリッドセルの X 座標

- グリッドセルの Y 座標
- グリッドセルに含まれる DOM ノードに現れる HTML タグの出現頻度
- グリッドセルに含まれる DOM ノードに現れる形態素の品詞の出現頻度
- グリッドセルに含まれる DOM ノードに現れる形態素の表層表現の出現頻度
- ページタイトル中の形態素の表層表現の出現頻度
- ページタイトル中の形態素の品詞の出現頻度

実験では, グリッドを 10×10 に分割した場合と 5×5 に分割した場合を比較するため, 5 分割交差検定を行った. 実験の結果を表 2.1 に示す. この結果から, グリッドを細かく分割しても, 精度の向上が見られないことがわかる. これは, 細分化することによりタスクが難しくなっていること, DOM ノードが描画される範囲がそれほど狭い範囲でないことが原因として考えられる.

表 2.1: 1 段階目の抽出の評価 [1]

10×10		5×5	
Precision	Recall	Precision	Recall
0.21	0.52	0.48	0.68

第二段階では, 固有表現抽出などで用いられるチャンキング手法を利用し, 抽出された DOM ノードに含まれるテキストから情報発信者と思われる単語列を特定する. 第二段階の抽出に用いる素性を以下に示す.

- 表層文字
- 品詞
- 形態素原型
- 文節内素性
- 主辞素性
- 角川類語辞典の分類番号
- 活用形の原型
- Juman により付与された単語の代表表記

第二段階の実験は, 第一段階の手法を行わない場合と, グリッドを 10×10 に分割した第一段階の手法が成功したと仮定した場合の 2 つの条件で行った. 実験の結果を表 2.2 に示す. この結果より, 1 段階目の抽出が成功すれば, 2 段階目の抽出精度が向上することが確認できた.

表 2.2: 2 段階目の抽出の評価 [1]

	Precision	Recall
1 段階目の絞り込みを行わない従来方法	0.53	0.47
1 段階目が成功したと仮定した場合	0.84	0.47

Kato らは, ウェブページの情報発信者情報を抽出するためのサブタスクとして, 情報発信者名の抽出を試みた [2]. Kato らの提案手法の概要を図 2.2 に示す. まず, 情報発信者候補を抽出する. その手法を以下に示す.

1. HTML から全テキストを抽出する.
2. テキストを文に分割する.
3. 文に KNP(日本語の文節係り受け解析ツール) を適用する.
4. 以下の条件を満たす文を保持する.
   (a) 文中の“ の ”を除いた助詞の割合が 閾値を超える文
   (b) 動詞につく助詞を含まない文 (“ 〜について ”など)
5. 保持された文から次のいずれかの条件を満たす句を抽出する.
   (a) 句に含まれる名詞の中に個人名や組織名として分類されたものが存在する.
   (b) 句に未知の単語が含まれている.
   (c) 句の最後の形態素が人名接尾辞 (“ ○○氏 ”など) あるいは組織名接尾辞 (“ ××会社 ”など) である.
6. 抽出された句から複合名詞を抽出する.

上記の手法で抽出された複合名詞を情報発信者候補とする. 次に, 情報発信者候補を以下の二つの特性を用いてランク付けする.

文書構造の特性 候補のノードの HTML タグやメインコンテンツからの距離

言語特性 候補の品詞タグ (人名, 組織名など)

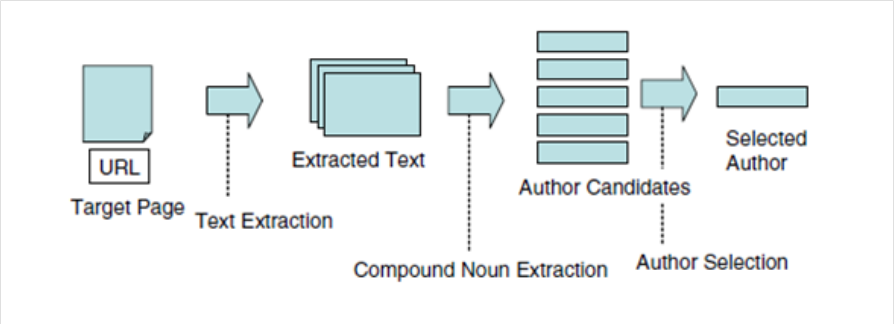

図 2.2: Kato らのシステム概要 [2]

文書構造の特性を用いる際に利用する距離計測の手法を, 図 2.3 を用いて説明する. author name から content までは, ⟨h1⟩ の親ノード ⟨div⟩ を通らなければならないので, この 2 つを結ぶパスは ⟨h1-div-table-tbody-tr-td⟩ となり, 距離は 5 となる.

情報発信者候補のランキングにはランキング SVM を用いる. ランキング SVM の学習のため, 訓練データにおけるラベルを以下のように設定した. なお, 正解の情報発信者を「ABC エレクトリック株式会社」としたとき, 各ラベルに該当する例も示す.

- 完全一致 → ラベル 2

  例: ABC エレクトリック株式会社

- 部分的一致 → ラベル 1

  例: ABC エレクトリック

- 不一致 → ラベル 0

ランキングの上位 k 番目までに正解の情報発信者が含まれている場合, そのページを正解とする. 実験の結果, ランキングの精度は表 2.3 のようになった. All は, 抽出された情報発信者候補をすべて出力した場合の精度である. ランキングが一位の候補のみを出力したときの精度は 58.6%であった. また, 自明ではあるが, k の値が大きいほど抽出の精度が高くなっていることがわかる.

表 2.3: ランキングの精度 [2]

k	Ranking Precision
1	0.586
3	0.720
5	0.752
All	0.847

```
<div>
 <h1>author name</h1>
 <table>
   <tbody>
     <tr>
       <td>content</td>
       <td>content2</td>
     </tr>
     <tr>
       <td>some other content</td>
     </td>
   </tbody>
 </table>
</div>
```

図 2.3: DOM ツリーにおける距離計測のための例 [2]

Giuffrida らは, PostScript で記述された科学論文からタイトル, 著者, 所属, 著者と所属の対応関係, 目次を抽出する手法を提案した [3]. 以下の二つの特性を利用し, メタデータを抽出する.

空間特性 タイトルは最初のページの最上部, 著者名はタイトルの下に記載されているといったレイアウト情報を用いた特性

言語特性 タイトルはそのページにおける最大のフォントで記述されているといったテキスト情報を用いた特性

Giuffrida らの手法で抽出が最も困難なメタデータは, 著者と所属の対応関係である. これを抽出するために Giuffrida らが設計したルールを以下に示す.

1. 各著者のバウンディングボックス中心の xy 位置を求める.
2. 各所属のバウンディングボックス中心の xy 位置を求める.
3. 著者-所属のすべての組合せに対し, その間のユークリッド距離を計算する.
4. それぞれの著者を空間的に最も近い所属にリンクする.

抽出対象とするメタデータの種類毎にこのようなルールを設計する. 各メタデータのルールの数を表 2.4 に示す.

表 2.4: 各メタデータのルールの数 [3]

メタデータ	ルールの数
タイトル	9
著者	12
所属	10
著者と所属の関係	10
目次	8

表 2.5: メタデータ抽出の正解率 [3]

メタデータ	Accuracy
タイトル	92%
著者	87%
所属	75%
著者と所属の関係	71%
目次	76%

抽出したメタデータの正解率を表 2.5 に示す. この結果から, タイトルや著者は書き方がある程度決まっているため, 他のメタデータより抽出しやすいことが確認された.

Kawahara らは, 与えられたトピックとその主要な述語項構造の矛盾についての概観を掲示する手法を提案した [4]. 例えば, トピック「ゆとり教育」について, 「学力が低下する」は主な述語項構造であり, 「学力が向上する」はその矛盾である.

Kawahara らの手法は以下のステップで構成される.

1. 述語項構造を抽出する

   与えられたトピックに対して検索された Web ページに以下の手順を適用し, 述語項構造を抽出する.

   1). 各 Web ページから重要な文を抽出する. 重要な文はトピック語の近くにある文とする.

   2). 重要な文に形態素解析器“ JUMAN ”および構文・構造解析器“ KNP ”を適用し, 述語項構造を抽出する.

   3). Web 全体とターゲットページ間の確率比に基づき, トピックに関係のない述語項構造を除外する.

2. 述語項構造をマージする

   同一の述語項構造は単にマージされ, 同意語もしくは他の構造に含まれる述語項構造は以下のようにマージされる.

- 同意語の述語項構造のマージ
  述語の同意語辞書を用いてマージする.
  例: “ 学力が低下する ” と “ 学力が下がる ” はマージされる.
- 他の構造に含まれる述語項構造のマージ
  包含関係にある述語項構造はマージされる.
  例: “ ゆとりで学力が低下する ” は “ 学力が低下する ” に含まれる.

3. 主要な述語項構造とその矛盾を検出する

   高頻度で出現する述語項構造およびその矛盾を抽出する. 矛盾は次のいずれかの条件を満たすすべての述語項構造を検索することにより得られる.

   - 否定フラグの不一致
     主要な述語項構造の述語に否定フラグがある (ない) 場合, 述語に否定フラグがない（ある) ものは矛盾する.
     例: “ 学力が低下しない ” は “ 学力が低下する ” の矛盾として抽出される.
   - 述語の反意語への置換
     矛盾する述語は主要な述語項構造の述語の反意語である.
     例: “ 学力が向上する ” は “ 学力が低下する ” の矛盾として抽出される.

“ レーシック手術 ”, “ 合成洗剤 ” など, 25 個のトピックを対象に提案手法の評価実験を行った. 図 2.4 に主要な述語項構造とその矛盾のペアの例を示す. 得られた述語項構造の正確さを評価するため, 与えられたトピックとの関連に基づいて次の三つのクラスにそれらを分類した.

A). 適切である

B). 適切であるが, 他の述語項構造へマージされるべきである

C). 適切でない

適切な述語項構造は, 以下の条件をすべて満たすものとする.

- トピックと関係がある
- 機能的で無意味な表現ではない
- 意味が同じであるオリジナルの文が存在する

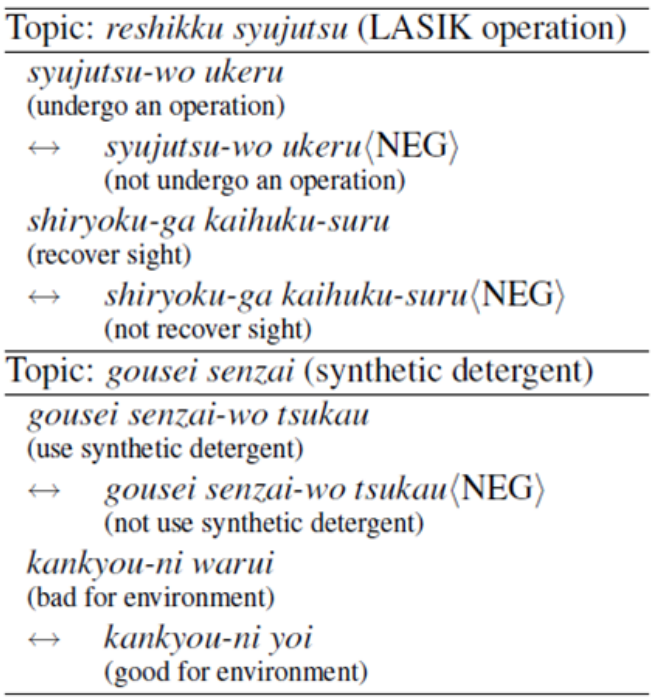

図 2.4: 主要な述語項構造とその矛盾のペアの例 [4]

抽出した述語項構造を分類した結果の例を図 2.5 に示す.

Kawahara らの手法の評価を表 2.6 に示す. major p-a は主要な述語項構造の抽出評価, contradictions は主要な述語項構造と矛盾するものの抽出評価である. この結果から, 分類クラスの A, B を正解としたとき, 正解率は主要な述語項構造が 82.5%, 矛盾が 79.3%であり, 高い値を示していることがわかる.

Kobayashi らは, 構造化されていないブログ記事からの顧客意見の抽出を試みた [5]. Kobayashi らの手法では, 以下の四種類の情報 (意見ユニット) を定義する.

- opinion holder

  評価をする人物

- subject

  特定のクラスの固有名 (製品や会社)

Topic: *yutori kyouiku* (cram-free education)	
yutori kyouiku-wo minaosu (reexamine cram-free education)	A
gakuryoku-ga teika-suru (scholastic ability deteriorates)	A
↔ *gakuryoku-ga koujou-suru* (scholastic ability ameliorates)	A
yutori kyouiku-ga hajimaru (cram-free education starts)	A
chikara-wo hagukumu (cultivate ability)	A
yutori-ga aru (have time)	A
kyouiku-wo nyuusu-kara yomu (read education from news)	A
gakuryoku teika-wo maneku (cause deterioration of scholastic ability)	B
yutori kyouiku-to iu (as cram-free education)	C
↔ *yutori kyouiku-to iu*⟨NEG⟩ (not as cram-free education)	C

図 2.5: 分類した評価結果の例 [4]

例: 自動車ドメインの車のモデル名

- aspect

  部品, 部材, 関連するオブジェクト, 評価される Subject の属性など

- evaluation

  Opinion holder の精神的/感情的な態度を表現するフレーズ

  例: 良い, 悪い, 強力, スタイリッシュなど

次に, これらのユニット間の関係である Asp-Eval 関係, Asp-of 関係を抽出する. Asp-Eval 関係は aspect と evaluation の関係であり, Asp-of 関係は, aspect が複数存在する場合のそれらの関係である. 例えば, “ 車のタイヤの部品 ” という文について, “ 車 ” が subject であり, “ タイヤ ”, “ 部品 ” が aspect であるため, “ タイヤの部品 ” が Asp-of 関係である. 例文とその意見ユニットの例を図 2.6 に示す. 図の左の例文に対し, 二種類の意見ユニットが抽出されている. このように, 1 つの文に意見ユニットが複数存在する場合もある.

表 2.6: Kawahara らの手法の評価 [4]

	major p-a	contradictions
relevant(A, B)	160/194 (82.5%)	46/58 (79.3%)
relevant(A)	118/194 (60.8%)	39/58 (67.2%)
should be merged(B)	42/194 (21.6%)	7/58 (12.1%)
not relevant(C)	34/194 (17.5%)	12/58 (20.7%)

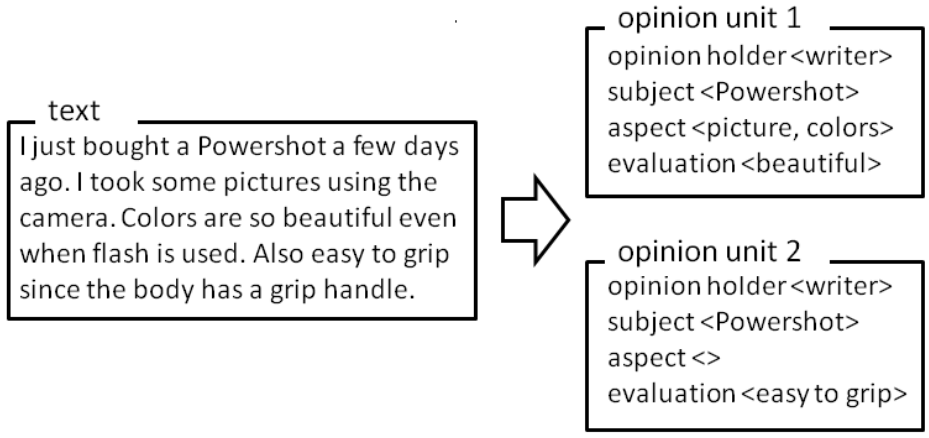

図 2.6: 例文とその意見ユニット [5]

実験データとして四種類のドメイン (レストラン (Rest)、自動車 (Auto)、携帯電話 (Phone)、ビデオゲーム (Game)) のブログ記事を収集し, アノテーターが意見ユニットを注釈した結果を表 2.7 に示す. I は意見ユニットの関係の数, II は意見ユニットの関係における aspect の数である. なお, other は aspect 数が 3 以上のものを表し, Non-writer op. holder は opinion holder が存在しないものを表す.

Asp-Eval 関係, Asp-of 関係の抽出には文脈パターンを用いる. 例えば, “接客が訓練されていて気持ち良い”という文は, 構文パターン ⟨Aspect⟩-ga VP-te ⟨Evaluation⟩ にマッチする. また, 文脈パターンの統計的な手掛かりも用いる. 例えば, aspect-aspect が出現すると, 同一文に aspect-evaluation も出現する傾向がある. 文書の集合に出現するこのようなパターンを教師あり機械学習し, Asp-Eval 関係や Asp-of 関係を抽出する.

単一の文の場合と複数の文の場合で, 関係抽出の方法が異なる. これらのモデルの違いを以下に示す.

表 2.7: 意見ユニットの注釈結果 [5]

		Rest	Auto	Phone	Game
	articles	1,356	564	481	361
	sentences	21,666	14,005	11,638	6,448
	# of opinion units	4,267	1,519	1,518	775
I	Asp-Eval	3,692	943	965	521
I	Asp-Asp	1,426	280	296	221
I	Subj-Asp	2,632	877	850	451
II	Subj-Eval	575	576	553	243
II	Subj-Asp-Eval	2,314	736	768	351
II	Subj-Asp-Asp-Eval	1065	175	172	127
II	other	313	32	25	54
	Non-writer op. holder	95	17	22	2

1). 単一文の関係抽出

- evaluation(または aspect) が与えられると, 上記の手法を使用し, 文中で最も可能性の高い aspect 候補を選択する.
- スコアが負の場合, 関係は複数の文にまたがるとみなし, 2) のステップに移る.

2). 複数文の場合の関係抽出

- evaluation(または aspect) が出現する文の前の文で最も可能性の高い aspect 候補を選択する.

Kobayashi らの手法と比較するため, Tateishi らの手法 [6] をベースラインシステムとし, これらを比較する実験を行った. また, ブリッジング参照解析 [7] を用いた共起統計モデルを Kobayashi らの手法と併用し, 精度の向上を図った. その結果を表 2.8, 表 2.9 に示す. 単語 A と単語 B が Asp-of 関係であり, 単語 B と単語 C も Asp-of 関係であるならば, 単語 A と単語 C も Asp-of 関係であるとみなす. そのため, 表 2.9 では Asp-of 関係は単一文と複数文を区別しないで評価している.

Asp-Eval 関係の抽出については, ベースラインより精度, 再現率ともに約 10%改善された. Asp-of 関係の抽出においても, 精度が 10%, 再現率が 20%以上改善された. しかし, 共起統計モデルとの併用での評価値の向上はわずかであった.

2.2 先行研究と本研究の違い

前節で紹介した百瀬らの研究, Kato らの研究, Giuffrida らの研究は, 情報発信者名や著者名を抽出対象としている. 一方, 本研究では作成者の名前だけでなく, ウェブサイトの

表 2.8: Asp-Eval 関係抽出の評価 [5]

Asp-Eval			
		単一文	複数文
ベースライン	P	0.56 (432/774)	—
	R	0.53 (432/809)	—
提案手法	P	0.70 (504/723)	0.13 (46/360)
	R	0.62 (504/809)	0.17 (46/274)
提案手法 + 共起統計モデル	P	0.72 (502/694)	0.14 (53/389)
	R	0.62 (502/809)	0.19 (53/274)

表 2.9: Asp-of 関係抽出の評価 [5]

Asp-of		
	precision	recall
ベースライン	0.27 (175/682)	0.17 (175/1048)
提案手法	0.44 (458/1047)	0.44 (458/1048)
提案手法 + 共起統計モデル	0.45 (474/1047)	0.45 (474/1048)

説明文 (サイト情報) や, 作成者に関する年齢, 性別, 職業, 自己紹介文など (作成者情報) をウェブページから取得する点に特徴がある. 先行研究のように単に情報発信者名を抽出するよりも, サイト情報や作成者情報を抽出し, ユーザに掲示することで, ウェブページの信頼性がより判断しやすくなると考えられる. Kawahara ら, Kobayashi らも, ウェブからの情報抽出を行っているが, ウェブサイトの信頼性を判断できる情報の抽出とは異なる研究である. また, 本研究では教師あり機械学習の手法を用いるほか, 負例のフィルタリングなど, 抽出精度を向上させるための様々な手法を試みる.

第3章　提案手法

3.1　概要

ブログページからのサイト情報や作成者情報の抽出は, HTML ファイルにおける Document Object Model (DOM) の個々のノードに対し, そのノードがサイト情報や作成者情報を含むか否かを判定することで実現する．Document Object Model とは, HTML の要素にアクセスするためのインターフェースである．DOM は, 通常 HTML ページ内における要素 (タグ) の関係を木構造で表現する．DOM ノードとは, その木構造におけるノードを指し, 1 つの HTML タグに対応する．また, HTML タグで囲まれたテキストを「DOM ノードが含むテキスト」と呼ぶ. 図 3.1 に簡単な HTML ページとそれに対応する DOM の木構造を示す. ⟨div⟩ のノードは ⟨h1⟩ と ⟨h2⟩ を子の要素として持つ. ⟨h1⟩ の DOM ノードは「テキスト 1」を含み, ⟨h2⟩ の DOM ノードは「テキスト 2」を含む. ここでの目標は, DOM ノードに対し, それが含むテキストがサイト情報や作成者情報であるかを判定することである.

サイト情報や作成者情報がタグ付けされたブログページの集合を用意し，上記の判定を行う分類器を教師あり機械学習によって獲得する．機械学習アルゴリズムは Support Vector Machine (SVM)[8] を用いた．

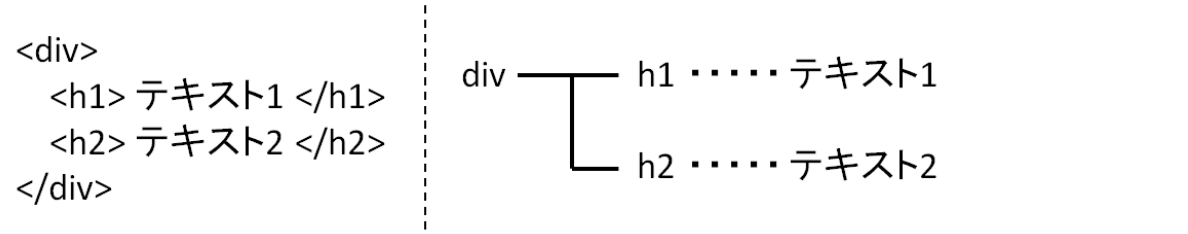

図 3.1: HTML ページと DOM の例

3.2 分類クラス

DOM ノードの分類クラスを以下のように定義する.

site サイト情報を含む DOM ノード

person 作成者情報を含む DOM ノード

site-link サイト情報が別ページに記述されているとき, それへのリンクを含む DOM ノード

person-link 作成者情報が別ページに記述されているとき, それへのリンクを含む DOM ノード

site-part テキストの一部のみがサイト情報に該当する DOM ノード

person-part テキストの一部のみが作成者情報に該当する DOM ノード

site-image サイト情報を含むが, テキストではなく画像によって表示している DOM ノード

person-image 作成者情報を含むが, テキストではなく画像によって表示している DOM ノード

other サイト情報や作成者情報を含まないノード

例として, site および person に分類される DOM ノードの領域を図 3.2 に示す. 「病気をしてから人生変わりました！ …… 」というテキストは, ブログの内容を紹介しているとみなせるため, サイト情報である. 一方, 「性別 女性 …… 」というテキストは, ブログの書き手の自己紹介やプロフィールを含むので, 作成者情報である. これらは 1 つの DOM ノードに含まれるテキストである.

person-link に分類される DOM ノードの領域の例を図 3.3 に示す. このブログでは, 「プロフィール」というテキストを含む DOM ノードに, 作成者情報が記述されているページへのリンクが含まれている. また, この図における「続きを見る」のように, 現在のページに作成者情報の一部が存在し, 残りの作成者情報が別ページに書かれているとき, そのページへのリンクも person-link としている.

person-part に分類される DOM ノードの領域の例を図 3.4 に示す. 「Author:mirura」が作成者情報に該当するテキストである. しかし, このテキストのみを含む DOM ノードは存在しない. そのため, テキストの一部に「Author:mirura」を含む DOM ノード (枠線で囲われた領域) を person-part とする. なお, このブログは飼い猫を紹介しているため, ブログのプロフィール欄に猫の情報が書かれている. 本研究では, ブログの真の書き手ではないため, このような情報は作成者情報とみなさない.

図 3.2: site および person を含むブログページの例

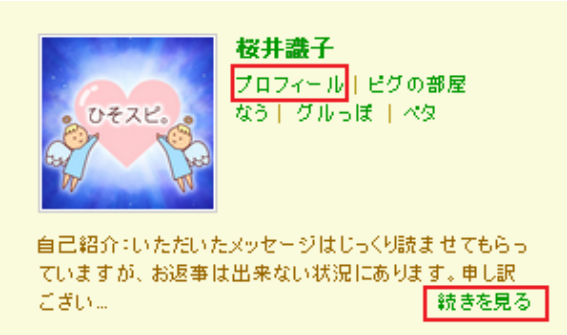

図 3.3: person-link を含むブログページの例

図 3.4: person-part を含むブログページの例

3.3 素性

機械学習に用いる素性は node+infor という形式で表現する．node は, 素性を取り出す DOM ノードを表わす．本研究では, node は判定対象の DOM ノード (N_t), N_t の親ノード (N_p), N_t の 1 つ前に出現する兄弟ノード (N_s), N_t の親の 1 つ前に出現する兄弟ノード (N_{ps}) のいずれかとする．すなわち, 判定対象のノードだけでなくその周辺のノードから得られる情報も素性として利用する．本研究の抽出対象の node の位置関係を図 3.5 に示す．一方, infor はサイト情報や作成者情報の存在の有無を判定する手がかりとなる情報を表わす．SVM の学習に用いる素性ベクトルの重みは, node に infor に該当する情報が存在すれば 1, それ以外は 0 とする．

以下, 本研究における infor の一覧を示す．

- DOM ノードのタグ名

  HTML タグは情報抽出の有力な手がかりとなると考えられる．例えば, site-link や

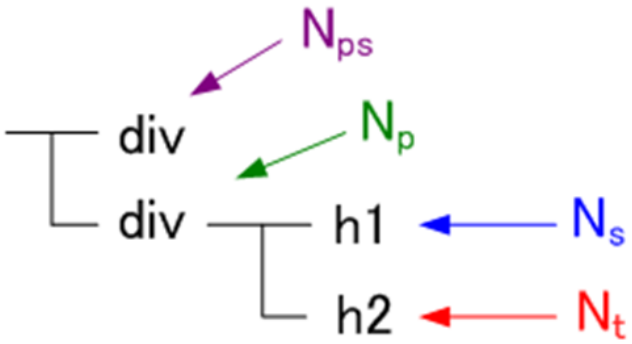

図 3.5: node の位置関係

person-link とタグ付けされたノードの HTML タグは必ず ⟨a⟩ であり, site とタグ付けされたノードの HTML タグは ⟨h1⟩, ⟨h2⟩ であることが多いという傾向がある.

- id, class の属性値

  id=”title” や class=”profile” のように, id や class の属性値にはサイト情報や作成者情報を示唆するキーワードが含まれていることがあるため, 素性とする. 属性値にスペース, ハイフン, アンダーバーが含まれている場合, これらで属性値を分割し, 分割された文字列を素性とする. 例えば, id=”title-top” という属性値からは ‘title’, ‘top’ の 2 つの素性を得る.

- テキスト長

  DOM ノードが支配するテキストの長さを l とし, l が $[1, 20]$, $[11, 30]$, ..., $[181, 200]$ の範囲にあるとき, もしくは $l = 0$, $l > 200$ のときに重みを 1 とする素性を導入した. サイト情報や作成者情報のテキストは短いものが多く, 一方ブログ本文のテキストは長いと考えられるため, テキスト長は両者を区別するために有効である. 一方, サイト情報や作成者情報の周囲のテキスト長は本研究の情報抽出にあまり有効ではないと考えられるため, この素性は node が N_t の場合のみ使用する.

- 自立語

  DOM ノードが支配するテキストに含まれる自立語を素性とする. これは, サイト情報には「ブログ」, 作成者情報には「年齢」「性別」などのキーワードが頻出するといった傾向を学習するためである. ただし, テキストが長い場合には, 素性数が多くなり, 過学習を引き起こすことが懸念される. そのため, N_t のノードからは先頭か

ら 20 番目まで, それ以外のノードからは先頭から 3 番目までに出現する単語のうち, 自立語のみを素性とする. 自立語を抽出する手法を以下に示す.

1. テキストを ChaSen[1] を用いて形態素解析する.
2. 先頭から 3 番目 (N_t のときは 20 番目) の単語品詞が表 3.1 に示した品詞のいずれかである場合, その単語を自立語として抽出する.

表 3.1: 自立語として抽出する品詞

名詞	名詞 - 接尾 - 一般
名詞 - 一般	名詞 - 接尾 - 人名
名詞 - 固有名詞	名詞 - 接尾 - 地域
名詞 - 固有名詞 - 一般	名詞 - 接尾 - サ変接続
名詞 - 固有名詞 - 人名	名詞 - 接尾 - 助動詞語幹
名詞 - 固有名詞 - 人名 - 一般	名詞 - 接尾 - 形容動詞語幹
名詞 - 固有名詞 - 人名 - 姓	名詞 - 接尾 - 副詞可能
名詞 - 固有名詞 - 人名 - 名	名詞 - 接尾 - 助数詞
名詞 - 固有名詞 - 組織	名詞 - ナイ形容詞語幹
名詞 - 固有名詞 - 地域	動詞
名詞 - 固有名詞 - 地域 - 一般	動詞 - 自立
名詞 - 固有名詞 - 地域 - 国	形容詞
名詞 - 代名詞	形容詞 - 自立
名詞 - 代名詞 - 一般	形容詞 - 接尾
名詞 - 代名詞 - 縮約	副詞
名詞 - 副詞可能	副詞 - 一般
名詞 - 左辺接続	副詞 - 助詞類接続
名詞 - 形容動詞語幹	未知語
名詞 - 接尾	記号 - アルファベット

- タイトル

  N_s もしくは N_{ps} が支配するテキストがそのページの ⟨title⟩ タグの内容 (ブログページのタイトル) と一致しているとき重みを 1 とする素性. この素性は, 4.1 節で述べる開発データを精査した結果, ブログタイトルと同一のテキストの近くにサイト情報が出現しやすいということが観察されたために設計した. 例えば, 図 3.6 は図 3.2 に示したブログの DOM の一部であるが, ⟨h2⟩ タグが判定対象の DOM ノードのとき, その 1 つ前の兄弟ノード (⟨h1⟩ タグ) が支配するテキスト「しゃかしゃか 3 人娘との毎日」はこのページの ⟨title⟩ タグと一致しており, タイトル素性の重みが 1 となる. この素性は node が N_t の場合のみ使用する.

[1] http://chasen-legacy.osdn.jp/

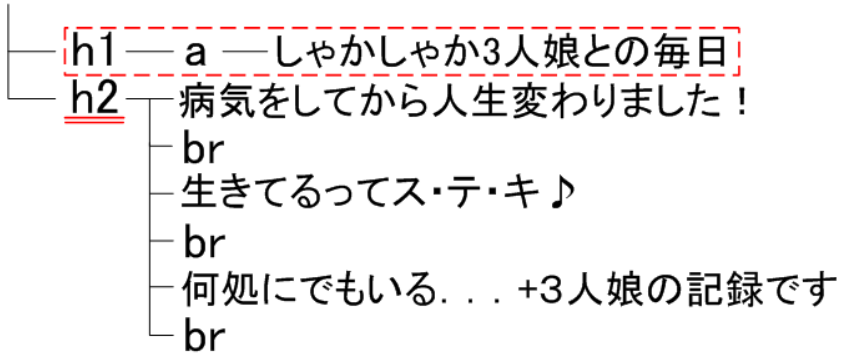

図 3.6: タイトル素性

- サイト情報を示唆するキーワード

  上述の素性が 1 であっても, その DOM ノードに含まれるテキストは常にサイト情報であるわけではない. そこで, タイトル素性の重みが 1 であり, かつテキストの文末が「です」「ます」「ブログ」「日記」「(動詞) + ブログ」「(動詞) + 日記」であるとき重みを 1 とする素性を導入した. この素性を導入したのは, ブログタイトルの周辺に存在するサイト情報は文末に上記のキーワードやパターンが出現することが多いという観察に基づいている. この素性は node が N_t の場合のみ使用する.

- サイト情報へのリンクを示唆するキーワード

  ノードが支配するテキストが「このブログについて」「～ブログとは」「ABOUT」というキーワードを含むときに重みを 1 とする素性. サイト情報へのリンク先が記述されているテキストは, 上記の表現を含んでいる傾向があるため, この素性を導入した.

- サイトの説明文に頻出する n-gram

  サイト情報はブログタイトルと同一のテキストの近くに出現しやすいことから, 本研究ではタイトル素性を導入した. しかし, タイトル素性が出現するノードが含むテキストは, 必ずしもサイト情報であるとは限らない. サイト情報を正確に識別するために, サイト情報を示唆するキーワードを素性としたが, 文末だけを手掛かりとしているため, 不十分である. 文末表現だけでなく, テキストの内容を見て, サイトの説明文であるかを判定するべきである. そこで, ブログのサイト説明文の集合を用意し, そこで頻出する n-gram を取得することで, テキストがサイト情報であるか否かを判別するための素性を得た. タイトル素性の重みが 1 となり, かつ DOM ノードのテキストにこれらの n-gram が含まれていれば, その素性の重みを 1 とする. サイト説明文のコーパスとして, 「人気ブログランキング」というサイト [2] における

[2] http://blog.with2.net/

ランキング上位 2 万件のブログの説明文を取得した. 本研究では n = 3 とし, n-gram の出現数の上位 100 個を素性とした. 素性とした n-gram の一部を表 3.2 に示す. また, 素性とした 100 個の n-gram を付録 A の表 A.1, 表 A.2 に示す.

表 3.2: サイトの説明文に頻出する n-gram の例

出現数	n-gram
558	紹介, し, て
557	ブログ, です, 。
281	綴っ, て, い
115	情報, を, 発信
100	更新, 中, ！

上述の素性は, N_t, N_p, N_s, N_{ps} の全てから抽出されるものと, N_t のみから抽出されるものがある. 素性ごとに抽出対象とする DOM ノードを表 3.3 にまとめる.

表 3.3: 素性と抽出対象となる DOM ノード

	N_t	N_p	N_s	N_{ps}
DOM ノードのタグ名	○	○	○	○
id, class の属性値	○	○	○	○
テキスト長	○	×	×	×
自立語	○	○	○	○
タイトル	○	×	×	×
サイト情報を示唆するキーワード	○	×	×	×
サイト情報へのリンクを示唆するキーワード	○	○	○	○
サイト情報の説明文に頻出する n-gram	○	×	×	×

3.4 負例のフィルタリング

本手法の問題設定では, 正例 (サイト情報や作成者情報を含む DOM ノード) よりも負例 (情報を含まない DOM ノード) の方が圧倒的に多い. 実際, 4.1 節の表 4.1 に示すように, 実験に使用したデータにおける負例の占める割合は 99%である. 訓練データにおける分類クラスの数に極端な偏りがあることは, SVM による判定の正解率の低下の原因になるため, 望ましくない.

そこで, 本研究では負例であると考えられる DOM ノードを訓練およびテストデータから除外することにより, 正例数と負例数のバランスを是正する. この処理を「負例のフィルタリング」と呼ぶ.

3.4.1 コンテンツ領域のフィルタリング

ウェブページは, そのページの主な内容を記述するコンテンツ領域と, サイト内リンク, 目次, 広告などを表示する非コンテンツ領域に分けることができる. サイト情報や作成者情報は非コンテンツ領域に配置されると考えられる. そこで, ウェブページのコンテンツ領域を自動的に検出し, その領域内の DOM ノードは全て負例とみなして削除することで, 負例のフィルタリングを行う. 本研究では, コンテンツ領域の検出アルゴリズムは Kato らの手法 [2] を用いた. このアルゴリズムを図 3.7 に示す. その概要を以下に示す.

1. ウェブページのテキストを全て含む DOM ノード (⟨body⟩ の DOM ノード) をメインノードとする [3].
2. メインノードの子ノードを探索する.
3. ノードが含むテキストの長さについて, 親ノードに対する比が閾値 t_m より大きい子ノードが存在する場合, その子ノードをメインノードとし, 2. に戻る. 存在しない場合, 現在のメインノードをコンテンツ領域を含む DOM ノードとして返す.

なお, 本研究では $t_m = 0.5$ としている. 図 3.7 のアルゴリズムは, DOM ノードが支配するテキストの長さを手掛かりにコンテンツ領域を検出する. しかし, ブログのコンテンツ領域にテキストの記述が少なく, 大部分が画像で満たされているものも存在する. この場合, コンテンツ領域を誤検出してしまう可能性がある. そこで, 本研究では, 画像のサイズをテキスト長に換算してコンテンツ領域を検出する. 画像のテキスト長は, $\langle$ 高さ $\times$ 幅 $\times$ α $\rangle$ とする. 本研究では, $\alpha = 0.1, 0.2, 0.5$ のうち, 4.1 節で述べる開発データにおけるサイト情報・作成者情報抽出の結果が一番良くなる値を選んだ. その結果, $\alpha = 0.1$ のとき, 提案手法の評価値が最も高かった. そのため, 本研究ではコンテンツ領域のフィルタリングを適用する際, α の値を 0.1 に設定する. 画像の高さと幅は, ⟨img⟩ タグの height 属性, width 属性から取得する. また, ノードに画像のサイズが記載されていない場合, その画像のテキスト長は一律に 100 とする.

検出したコンテンツ領域の中に, ブログタイトル (⟨title⟩ タグに含まれるテキスト) および「プロフィール」「profile」というテキストを含むノードが存在する場合, コンテンツ領域にサイト情報や作成者情報が含まれている可能性が高い. この場合は, 非コンテンツ領域が誤ってコンテンツ領域と検出されていると考えられる. そのため, 上記のノードが存在する場合, 検出した領域を含むノードを削除しないことにした.

3.4.2 テキスト長が 0 のノードのフィルタリング

ウェブページの HTML の DOM ノードには, テキストを含まないノード (テキスト長が 0 のノード) も存在する. テキスト長が 0 の DOM ノードは, 明らかにサイト情報, 作成者

[3] メインノードとは, コンテンツ領域を含む DOM ノードを表す.

情報を含んでいない. そのため, テキスト長が 0 のノードを負例とみなし, 削除する. このフィルタリングは, コンテンツ領域のフィルタリングを実行した後で行う.

3.4.3 2 通りの負例のフィルタリング手法

3.4.1 項で説明したコンテンツ領域のフィルタリングを適用する際, 画像の大きさをテキスト長に換算する手法を用いる. この手法が負例のフィルタリング手法として効果的であるかを検証するため, 本研究では以下の 2 種類のフィルタリング手法を定める.

フィルタリング T コンテンツ領域のフィルタリングにおいて, 画像の大きさをテキスト長に換算せず, 負例のフィルタリングを実行する手法

フィルタリング I コンテンツ領域のフィルタリングにおいて, 画像の大きさをテキスト長に換算し, 負例のフィルタリングを実行する手法

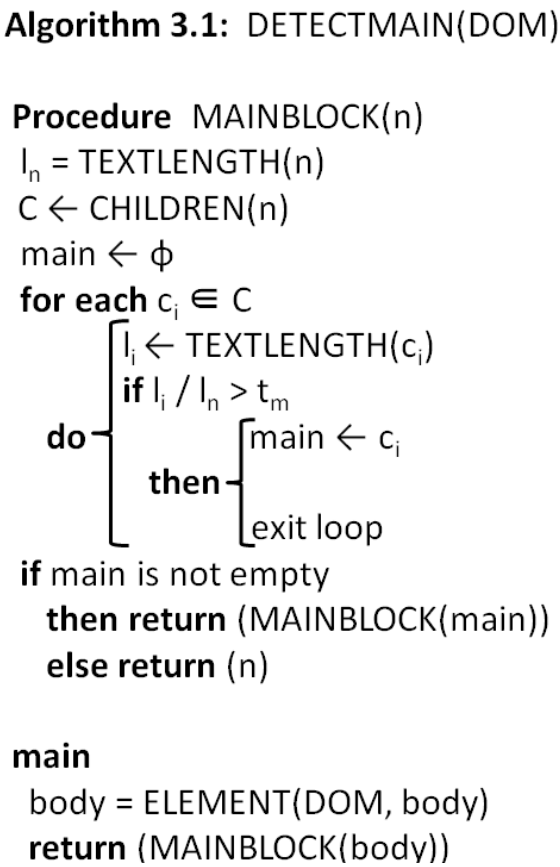

図 3.7: コンテンツ領域検出アルゴリズム (Kato et al. (2008) p.39 Figure 4)

第4章　評価

本章では, サイト情報, 作成者情報を抽出した評価結果および考察について述べる. 4.1 節では, 実験データとするブログページや, タグ付けした分類クラス分布について述べる. 4.2 節では, 提案手法と比較するために構築したベースラインシステムについて説明する. 4.3 節では, 提案手法ならびにベースラインシステムの評価基準について説明する. 4.4 節では, 提案手法およびベースラインシステムでサイト情報, 作成者情報を抽出した結果を報告し, その考察を述べる. 4.5 節では, サイト情報, 作成者情報を抽出できなかった要因を述べる.

4.1　実験データ

まず, 実験データとするブログページを収集した. ウェブには Yahoo!,goo,FC2 などのブログサービスが存在するが, 様々なブログサービスのブログを収集するために, 「人気ブログランキング」からランキング上位 500 件のブログのトップページを取得した. 次に, これらのブログページに対し, サイト情報と作成者情報を人手でタグ付けした. 表 4.1 に分類クラス毎にタグ付けした DOM ノード数を示す.

今回の実験では, site-part と person-part は数が少なかったため, それぞれ site もしくは person と同じとみなした. また, site-image や person-image は数も少なく, また画像で表示された情報を抽出することは難しいことから, 今回の実験では抽出の対象外とし, other と同じであるとみなした. 取得した 500 件のブログを 10 分割 (それぞれ D_1〜D_{10} とする) し, 交差検定を実行した. また, D_{10} は提案手法の設計や素性の考案などのための開発データとして用いた. 一方, 訓練データとは別のブログを 50 件新たにダウンロードし, これをテストデータとした. テストデータにも人手でサイト情報と作成者情報を付与した. テストデータにおける分類クラスの出現頻度を表 4.2 に示す. 訓練データから SVM を学習し, テストデータでその性能を評価する実験も行う. 以下, テストデータを D_{test} と記す.

4.2　ベースライン

提案手法との比較のために, 以下のルールにしたがって DOM ノードを分類するベースラインシステムを構築した.

1. N_s または N_{ps} のテキストがブログのタイトルと一致するとき, site と判定する.

表 4.1: 訓練データの分類クラス別の DOM ノード数

site	252	person	243
site-link	14	person-link	183
site-part*	17	person-part*	35
site-image*	8	person-image*	2
		other	668386

表 4.2: テストデータの分類クラス別の DOM ノード数

site	20	person	28
site-link	0	person-link	21
site-part*	1	person-part*	3
site-image*	0	person-image*	0
		other	55424

2. N_t のテキストが「について」「とは」「about」という文字列を含み, かつタグが $\langle$a$\rangle$ であるとき, site-link と判定する.

3. N_s または N_{ps} のテキストが「プロフィール」「profile」であるとき, person と判定する.

4. N_t のテキストが「プロフィール」「profile」であり, かつタグが $\langle$a$\rangle$ であるとき, person-link と判定する.

5. それ以外は other と判定する.

4.3 評価基準

提案手法ならびにベースラインは, サイト情報と作成者情報の抽出を, HTML ファイルから分類クラスに該当する DOM ノードを検索するタスクとみなしたときの精度, 再現率, F 値で評価する. 精度 (P), 再現率 (R), F 値 (F) の定義式を以下に示す.

$$P = \frac{\text{正例の } DOM \text{ ノードを正しく判定した数}}{\text{正例と判定した } DOM \text{ ノードの数}} \tag{4.1}$$

$$R = \frac{\text{正例の } DOM \text{ ノードを正しく判定した数}}{\text{評価データにおける正例の } DOM \text{ ノードの数}} \tag{4.2}$$

$$F = \frac{2 \times P \times R}{P + R} \tag{4.3}$$

P, R, F は other 以外の分類クラス, すなわち site, site-link, person, person-link のそれぞれについて算出し, 評価する. 訓練データにおける 10 分割交差検定では, 分割された 10 個の部分データのそれぞれをテストデータとしたときの結果, 及び 10 回の試行の平均を示す. 一方, テストデータに対する P, R, F も示す.

4.4 実験結果と考察

4.4.1 提案手法の評価

実験の結果を表 4.3〜表 4.6 に示す. 表 4.3 は, 訓練データの 10 分割交差検定の 10 回の試行におけるベースラインシステムの評価結果, 表 4.4 は, 同じく提案手法の評価結果である. 一方, 表 4.5 及び表 4.6 は, 訓練データ及びテストデータにおいてベースラインと提案手法を比較している. 表 4.5 は交差検定の結果 (10 回の試行の平均) である. また, 提案手法における負例のフィルタリング手法は, 3.4.3 項で説明したフィルタリング I を適用した.

まず, ベースラインシステムについて考察する. person-link の F 値は, 全てのデータにおいて高い値を示している. しかし, site, site-link, person は, 全体的に再現率は高いが, 精度は低い. このことから, 4.2 節で説明したルールで多くの正例を抽出することができるが, 負例にもルールの条件を満たしているものが多いことがわかる. また, D_3, D_6, D_7, D_9 に対しては, site-link であると判定した DOM ノードは存在せず, site-link の抽出に完全に失敗している. 次に, 表 4.3 における D_1〜D_{10} の F 値を比較する. site に関して, 最も低い値は 0.235(D_4), 最も高い値は 0.541(D_9) であり, 最大で 0.306 の差が生じた. 同様に, site 以外のクラスにおけるデータ間の最大の差を求めると, site-link は 0.253, person は 0.3, person-link は 0.21 であった. また, 各クラスの平均値を算出し, その値との差が ±0.1 以上生じたデータを調査した. その結果, site は D_2, D_4, D_8, D_9, D_{10}, site-link は D_{10}, person は D_1, person-link は D_9 であった. このことから, site はデータ間の最大の差も大きく, 平均値との差が ±0.1 以上生じたデータも多いため, 評価データによってサイト情報抽出の性能が大きく異なることがわかる.

次に, 提案手法の評価について考察する. site-link に関して, D_2 以外は抽出に完全に失敗している. これは, 訓練データにおいて site-link とタグ付けされた事例が少ないためであると考えられる. site, person, person-link の F 値を比較すると, 全体的に, person-link が高く, 次点で person, 最も低いのが site であった. このことから, サイト情報と作成者情報のうち, 提案手法では作成者情報の方が正確に抽出できることがわかる. ベースラインシステムと同様に, 表 4.4 において各クラスのデータ間の最大の差を算出すると, site が 0.192, person が 0.367, person-link が 0.309 であった. また, 各クラスの平均値との差が ±0.1 以上生じたデータは, person は D_1, D_4, D_5, D_6, D_{10}, person-link は D_7, D_9 であり, site に関しては該当するデータはなかった. このことから, person, person-link に関しては, データ間で評価値にばらつきがあることがわかる.

表 4.3: ベースラインシステムの評価 (訓練データにおける 10 分割交差検定)

	精度			
	site	site-link	person	person-link
D_1	0.246	0.067	0.027	1.000
D_2	0.153	0.077	0.146	0.714
D_3	0.300	—	0.161	0.909
D_4	0.148	0.063	0.143	0.857
D_5	0.311	0.125	0.217	0.933
D_6	0.194	—	0.132	0.909
D_7	0.214	—	0.225	0.950
D_8	0.392	0.059	0.219	0.733
D_9	0.411	—	0.230	1.000
D_{10}	0.388	0.250	0.157	0.733
	再現率			
	site	site-link	person	person-link
D_1	0.600	0.500	0.462	0.692
D_2	0.565	0.333	0.583	0.714
D_3	0.643	—	0.600	0.625
D_4	0.571	1.000	0.480	0.545
D_5	0.704	1.000	0.670	0.700
D_6	0.633	—	0.474	0.556
D_7	0.682	—	0.714	0.704
D_8	0.667	1.000	0.694	0.579
D_9	0.793	—	0.742	0.750
D_{10}	0.765	0.667	0.706	0.667
	F 値			
	site	site-link	person	person-link
D_1	0.349	0.118	0.051	0.818
D_2	0.241	0.125	0.233	0.714
D_3	0.409	—	0.254	0.741
D_4	0.235	0.118	0.220	0.667
D_5	0.432	0.222	0.331	0.800
D_6	0.297	—	0.207	0.690
D_7	0.326	—	0.342	0.809
D_8	0.494	0.111	0.333	0.647
D_9	0.541	—	0.351	0.857
D_{10}	0.515	0.364	0.257	0.667

表 4.4: 提案手法の評価 (訓練データにおける 10 分割交差検定)

	精度			
	site	site-link	person	person-link
D_1	0.667	—	0.611	0.917
D_2	0.556	1.000	0.750	0.792
D_3	0.667	—	0.824	0.929
D_4	0.478	—	0.692	0.857
D_5	0.684	—	0.917	0.850
D_6	0.720	—	0.882	0.867
D_7	0.579	—	0.895	0.963
D_8	0.750	—	0.629	0.762
D_9	0.810	—	0.692	0.952
D_{10}	0.792	—	1.000	0.750
	再現率			
	site	site-link	person	person-link
D_1	0.480	—	0.423	0.846
D_2	0.435	0.333	0.500	0.905
D_3	0.571	—	0.560	0.813
D_4	0.524	—	0.360	0.545
D_5	0.481	—	0.759	0.850
D_6	0.600	—	0.789	0.722
D_7	0.500	—	0.607	0.963
D_8	0.500	—	0.611	0.842
D_9	0.586	—	0.581	1.000
D_{10}	0.559	—	0.765	0.833
	F 値			
	site	site-link	person	person-link
D_1	0.558	—	0.500	0.880
D_2	0.488	0.500	0.600	0.844
D_3	0.615	—	0.667	0.867
D_4	0.500	—	0.474	0.667
D_5	0.565	—	0.830	0.850
D_6	0.655	—	0.833	0.788
D_7	0.537	—	0.723	0.963
D_8	0.600	—	0.620	0.800
D_9	0.680	—	0.632	0.976
D_{10}	0.655	—	0.867	0.789

表 4.5: 提案手法とベースラインの実験結果 (訓練データにおける 10 分割交差検定)

		精度	再現率	F 値
ベースライン	site	0.276	0.662	0.384
	site-link	0.107	0.750	0.176
	person	0.116	0.613	0.258
	person-link	0.874	0.653	0.741
提案手法	site	0.670	0.524	0.585
	site-link	1.000	0.333	0.500
	person	0.789	0.596	0.675
	person-link	0.864	0.832	0.842

表 4.6: 提案手法とベースラインの実験結果 (テストデータ)

		精度	再現率	F 値
ベースライン	site	0.320	0.762	0.451
	site-link	—	—	—
	person	0.208	0.645	0.315
	person-link	0.722	0.619	0.667
提案手法	site	0.667	0.667	0.667
	site-link	—	—	—
	person	0.750	0.677	0.712
	person-link	0.840	1.000	0.913

次に, 表 4.5 の結果を基に, 提案手法とベースラインを比較する. 再現率について, site, site-link, person は提案手法よりベースラインの方が高くなった. しかし, 精度では提案手法がベースラインを大きく上回った. 一方, person-link に関しては, 精度はベースライン, 再現率は提案手法が高くなった. 提案手法とベースラインの F 値を比較すると, 全てのクラスで提案手法がベースラインを上回った. このことから, 訓練データにおける 10 分割交差検定の結果からは, ベースラインより提案手法の方が性能が高いことがわかる.

次に, 表 4.6 の結果を基に, 提案手法とベースラインを比較する. site の再現率については, 提案手法よりベースラインの方が高くなった. しかし, それ以外では提案手法はベースラインを上回った. F 値を比較すると, 提案手法はベースラインと比べて, site では 0.216, person では 0.397, person-link では 0.346 ほど高い. このことから, テストデータに対する結果からも提案手法の有効性が確認された.

4.4.2 素性の評価

表 4.7: 素性の評価 (精度, テストデータ D_{test})

	精度			
	site	site-link	person	person-link
F_{-tag}	0.667	—	0.808	0.895
	(−0.000)		(+0.058)	(+0.055)
$F_{-id,class}$	0.571	—	0.714	0.808
	(−0.096)		(−0.036)	(−0.032)
$F_{-length}$	0.579	—	0.778	0.808
	(−0.088)		(+0.028)	(−0.032)
F_{-bow}	0.615	—	0.955	1.000
	(−0.052)		(+0.205)	(+0.160)
F_{-title}	0.684	—	0.750	0.840
	(+0.017)		(−0.000)	(−0.000)
$F_{-sitekey}$	0.667	—	0.750	0.840
	(−0.000)		(−0.000)	(−0.000)
$F_{-linkkey}$	0.667	—	0.750	0.840
	(−0.000)		(−0.000)	(−0.000)
$F_{-n\text{-}gram}$	0.770	—	0.778	0.840
	(+0.103)		(+0.028)	(−0.000)
F_{all}	0.667	—	0.750	0.840

表 4.8: 素性の評価 (再現率, テストデータ D_{test})

	再現率			
	site	site-link	person	person-link
F_{-tag}	0.571	—	0.677	0.810
	(−0.096)		(−0.000)	(−0.190)
$F_{-id,class}$	0.571	—	0.645	1.000
	(−0.096)		(−0.022)	(−0.000)
$F_{-length}$	0.524	—	0.677	1.000
	(−0.143)		(+0.010)	(−0.000)
F_{-bow}	0.762	—	0.677	0.476
	(+0.095)		(+0.010)	(−0.524)
F_{-title}	0.619	—	0.677	1.000
	(−0.048)		(+0.010)	(−0.000)
$F_{-sitekey}$	0.667	—	0.677	1.000
	(−0.000)		(+0.010)	(−0.000)
$F_{-linkkey}$	0.667	—	0.677	1.000
	(−0.000)		(+0.010)	(−0.000)
$F_{-n\text{-}gram}$	0.667	—	0.677	1.000
	(−0.000)		(+0.010)	(−0.000)
F_{all}	0.667	—	0.667	1.000

表 4.9: 素性の評価 (F 値, テストデータ D_{test})

	F 値			
	site	site-link	person	person-link
F_{-tag}	0.615	—	0.737	0.895
	(−0.052)		(+0.025)	(−0.018)
$F_{-id,class}$	0.571	—	0.678	0.894
	(−0.096)		(−0.034)	(−0.019)
$F_{-length}$	0.550	—	0.724	0.894
	(−0.117)		(+0.012)	(−0.019)
F_{-bow}	0.681	—	0.792	0.645
	(+0.014)		(+0.080)	(−0.268)
F_{-title}	0.650	—	0.712	0.913
	(−0.017)		(−0.000)	(−0.000)
$F_{-sitekey}$	0.667	—	0.712	0.913
	(−0.000)		(−0.000)	(−0.000)
$F_{-linkkey}$	0.667	—	0.712	0.913
	(−0.000)		(−0.000)	(−0.000)
$F_{-n\text{-}gram}$	0.683	—	0.724	0.913
	(+0.016)		(+0.012)	(−0.000)
F_{all}	0.667	—	0.712	0.913

表 4.10: 素性の評価 (精度, 開発データ D_{10})

	精度			
	site	site-link	person	person-link
$F_{-\text{tag}}$	0.900	—	0.889	0.846
	(+0.108)		(−0.111)	(+0.096)
$F_{-\text{id,class}}$	0.762	—	0.703	0.750
	(−0.030)		(−0.297)	(−0.000)
$F_{-\text{length}}$	0.818	—	1.000	0.750
	(+0.026)		(−0.000)	(−0.000)
$F_{-\text{bow}}$	0.821	—	0.864	1.000
	(+0.029)		(−0.136)	(+0.250)
$F_{-\text{title}}$	0.833	—	1.000	0.750
	(+0.041)		(−0.000)	(−0.000)
$F_{-\text{sitekey}}$	0.833	—	1.000	0.750
	(+0.041)		(−0.000)	(−0.000)
$F_{-\text{linkkey}}$	0.783	—	1.000	0.750
	(−0.009)		(−0.000)	(−0.000)
$F_{-\text{n-gram}}$	0.818	—	1.000	0.750
	(+0.026)		(−0.000)	(−0.000)
F_{all}	0.792	—	1.000	0.750

表 4.11: 素性の評価 (再現率, 開発データ D_{10})

	再現率			
	site	site-link	person	person-link
$F_{-\text{tag}}$	0.529	—	0.706	0.611
	(−0.030)		(−0.059)	(−0.222)
$F_{-\text{id,class}}$	0.471	—	0.765	0.833
	(−0.088)		(−0.000)	(−0.000)
$F_{-\text{length}}$	0.529	—	0.735	0.833
	(−0.030)		(−0.030)	(−0.000)
$F_{-\text{bow}}$	0.676	—	0.559	0.500
	(+0.117)		(−0.206)	(−0.333)
$F_{-\text{title}}$	0.588	—	0.765	0.833
	(+0.029)		(−0.000)	(−0.000)
$F_{-\text{sitekey}}$	0.588	—	0.765	0.833
	(+0.029)		(−0.000)	(−0.000)
$F_{-\text{linkkey}}$	0.529	—	0.765	0.833
	(−0.030)		(−0.000)	(−0.000)
$F_{-\text{n-gram}}$	0.529	—	0.765	0.833
	(−0.030)		(−0.000)	(−0.000)
F_{all}	0.559	—	0.765	0.833

表 4.12: 素性の評価 (F 値, 開発データ D_{10})

	F 値			
	site	site-link	person	person-link
$F_{-\text{tag}}$	0.667	—	0.787	0.710
	(+0.012)		(−0.080)	(−0.079)
$F_{-\text{id,class}}$	0.582	—	0.732	0.789
	(−0.073)		(−0.135)	(−0.000)
$F_{-\text{length}}$	0.643	—	0.847	0.789
	(−0.012)		(−0.020)	(−0.000)
$F_{-\text{bow}}$	0.742	—	0.679	0.667
	(+0.087)		(−0.188)	(−0.122)
$F_{-\text{title}}$	0.690	—	0.867	0.789
	(+0.035)		(−0.000)	(−0.000)
$F_{-\text{sitekey}}$	0.690	—	0.867	0.789
	(+0.035)		(−0.000)	(−0.000)
$F_{-\text{linkkey}}$	0.632	—	0.867	0.789
	(−0.023)		(−0.000)	(−0.000)
$F_{-\text{n-gram}}$	0.643	—	0.867	0.789
	(−0.012)		(−0.000)	(−0.000)
F_{all}	0.655	—	0.867	0.789

次に, 本研究で提案した素性の有効性を評価する. ここでは, 全ての素性を用いて学習した SVM と, 1 つの素性を除外して学習した SVM の評価値を比較する. もし, 素性を除くことで精度, 再現率, F 値が大きく低下するなら, その素性はサイト情報や作成者情報の抽出に有効に働くと言える. F_{-tag} は DOM ノードのタグ名, $F_{-id,class}$ は id, class の属性値, $F_{-length}$ はテキスト長, F_{-bow} は自立語, F_{-title} はタイトル素性, $F_{-sitekey}$ はサイト情報を示唆するキーワード, $F_{-linkkey}$ はサイト情報へのリンクを示唆するキーワード, $F_{-n\text{-}gram}$ はサイトの説明文に頻出する n-gram を除いた素性集合を表す. 一方, 全ての素性の集合を F_{all} と表す. なお, この実験では, フィルタリング I によって負例を削除する処理を行った.

F_{all} ならびに 1 つの素性を除いた素性集合を用いたときのテストデータにおける精度, 再現率, F 値を表 4.7, 表 4.8, 表 4.9 に示す. 表中の () は F_{all} との差を表す. $F_{-n\text{-}gram}$ と F_{all} の F 値を比較すると, site, person は $F_{-n\text{-}gram}$ の方が高くなっており, person-link は同じ値である. $F_{-n\text{-}gram}$ が F_{all} を下回っているクラスが存在しないため, サイトの説明文に頻出する n-gram は有効な素性ではないことがわかる. 全てのクラスで F_{all} を下回っている素性集合は $F_{-id,class}$ のみである. このことから, 最も有効な素性は id, class の属性値であると言える. それぞれのクラスについて, 最も値が低い素性集合は, site が $F_{-length}$, person が $F_{-id,class}$, person-link が F_{-bow} である. このことから, site はテキスト長, person は id, class の属性値, person-link は自立語の素性がそれぞれの抽出に有効であることがわかる.

次に, 本研究で開発データとした D_{10} でも同様の実験を行い, 素性を評価した. その結果を表 4.10, 表 4.11, 表 4.12 に示す. F_{-title} と F_{all} を比較すると, site は F_{-title} の方が高くなっており, person, person-link は同じ値である. また, $F_{-sitekey}$ と F_{all} を比較しても, site は $F_{-sitekey}$ の方が高くなっており, person, person-link は同じ値である. このように, F_{all} を下回っているクラスが存在しない素性集合は, F_{-title} と $F_{-sitekey}$ であるため, タイトル素性, サイト情報を示唆するキーワードは有効な素性ではないことがわかる. F_{all} を上回っているクラスが存在しない素性集合は, $F_{-id,class}$ と $F_{-length}$ である. この 2 つの素性集合を比較すると, person-link の F 値は同じであり, site, person の F 値はどちらも $F_{-id,class}$ の方が低い. このことから, 最も有効な素性は id, class の属性値であると言える. それぞれのクラスについて, 最も値が低い素性集合は, site が $F_{-id,class}$, person が F_{-bow}, person-link も F_{-bow} である. このことから, site は id, class の属性値, person および person-link は自立語の素性がそれぞれの抽出に有効であることがわかる.

サイトの説明文に頻出する n-gram の素性は D_{10} では有効だが, D_{test} では有効ではなかった. 逆に, タイトル素性は D_{test} では有効だが, D_{10} では有効ではなかった. このように, テストデータと開発データで有効な素性に違いが見られた. そのため, タイトル素性やサイトの説明文に頻出する n-gram の素性が有効であるかを明確に確認することはできなかった. しかし, id, class の属性値の素性は, 両方のデータで最も有効に働いたため, この素性はサイト情報, 作成者情報の抽出に特に有効であることが確認された.

4.4.3 フィルタリングの評価

3.4 節で述べた負例のフィルタリング手法を評価する. 表 4.1 に示すように, 訓練データにおける DOM ノードの総数は 668386 個であるが, 3.4.3 項で説明したフィルタリング T を実行した結果, 192161 個になった. DOM ノードの総数を大きく削減することに成功したが, いくつかの正例も誤って削除されている. 削除された正例の数は, site が 6 個, site-link が 0 個, site-part が 4 個, site-image が 6 個, person が 7 個, person-link が 6 個, person-part が 4 個, person-image が 2 個であった. 削除されたノードの総数がおよそ 71%なのに対し, 削除される正例はおよそ 5%であった.

3.4.3 項で説明したフィルタリング I を実行した結果, 訓練データにおける DOM ノードは 196954 個になった. このフィルタリングにより誤って削除された正例の数は, site が 2 個, site-part が 2 個, person が 2 個, person-link が 2 個, person-part が 1 個, それ以外は 0 個である. 削除されたノードの総数がおよそ 70%なのに対し, 削除される正例はわずか 1%程度に抑えることができた.

負例のフィルタリングを実行しないとき, フィルタリング T を実行したとき, フィルタリング I を実行したときの精度, 再現率, F 値を評価した. テストデータにおける実験結果を表 4.13 に示す. フィルタリング T を実行したときと, フィルタリングを実行しないときを比較すると, person-link の値に変化はなかったが, site, person の F 値はフィルタリング T を実行したときの方が高くなった. person はフィルタリング T によって精度のみが向上し, site に関しては精度, 再現率ともに向上した. このことから, フィルタリング T が有効に働いていることがわかる.

フィルタリング T を実行したときと, フィルタリング I を実行したときを比較すると, person-link の値に変化はなかったが, site, person の F 値はフィルタリング T を実行したときの方が高くなった. 精度と再現率を比較すると, site の精度, 再現率及び person の精度において, フィルタリング T を実行したときの方が高くなっている. また, フィルタリング I は, フィルタリングを実行しないときよりも site, person の F 値が低くなった. この原因として, フィルタリング I によって正例が削除されてしまったことが考えられる. D_{test} に関して, フィルタリング I により, site の DOM ノードは 20 個のうち 5 個 (25%) が, person の DOM ノードは 28 個のうち 3 個 (約 11%) が誤って削除された. これに対し, フィルタリング T で誤って削除された正例の数は, site の DOM ノードは 20 個のうち 1 個（5%）であり, person の DOM ノードは削除されなかった. この結果から, フィルタリング I よりフィルタリング T の方が, 正例が削除されていないことがわかる. このことから, 3.4.1 項で説明したコンテンツ領域のフィルタリングについて, 画像の大きさをテキスト長に換算する手法は効果的ではないことがわかる.

次に, 開発データ D_{10} に関しても同様の実験を行った. その結果を表 4.14 に示す. テストデータのときとは異なり, 開発データではフィルタリング T よりフィルタリング I の方が person の F 値が向上していることがわかった. 3.4.1 項で説明したように, 画像の大きさをテキスト長に換算する際の比率 α は 0.1 に定めているが, これは開発データの F 値が最大となるように設定したため, person の F 値においてフィルタリング I がフィルタリング

表 4.13: 負例のフィルタリングの評価 (テストデータ D_{test})

	精度			
	site	site-link	person	person-link
フィルタリングなし	0.682	—	0.778	0.840
フィルタリング T	0.762	—	0.808	0.840
フィルタリング I	0.667	—	0.750	0.840
	再現率			
	site	site-link	person	person-link
フィルタリングなし	0.714	—	0.677	1.000
フィルタリング T	0.762	—	0.677	1.000
フィルタリング I	0.667	—	0.677	1.000
	F 値			
	site	site-link	person	person-link
フィルタリングなし	0.698	—	0.724	0.913
フィルタリング T	0.762	—	0.737	0.913
フィルタリング I	0.667	—	0.712	0.913

T を上回ったと考えられる. しかし, フィルタリング T, フィルタリング I ともに, フィルタリングを実行しないときよりも person の F 値が高いため, 負例のフィルタリングが作成者情報の抽出に有効であることがわかる. 一方, site の評価値はフィルタリング T, フィルタリング I ともにフィルタリングを実行しないときよりも低下している. これも, 負例のフィルタリングにより正例が削除されたことが原因であると考えられる. D_{10} に関して, フィルタリング I により, site の DOM ノードは 31 個のうち 4 個 (約 13%) が, person の DOM ノードは 32 個のうち 2 個 (約 6%) が誤って削除された. これに対し, フィルタリング T で削除された正例は, site の DOM ノードは 31 個のうち 1 個 (約 3%) が削除され, person の DOM ノードは削除されなかった. site の正例が person よりも多く削除されていることが, フィルタリングによって person の F 値は向上するが site の F 値は低下する原因と考えられる.

4.5 エラー分析

提案手法のエラー分析を行った. 表 4.3 に示した通り, ベースラインシステムの site の再現率が大きくなっているため, 多くのブログにおいて, サイト情報がブログタイトルの近辺に存在することがわかった. そのため, 提案手法ではタイトル素性を導入したが, ブログタイトルの近辺のテキストが必ずしもサイト情報であるとは限らないため, 正しく抽出できない事例が存在した. このようなブログの例を図 4.1 に示す. 線で囲まれたテキストは, ブログタイトルの近辺に記載されているため, タイトル素性の重みが 1 となり, 提案手法では site に分類された. しかし, このテキストからはサイトの内容を読みとることがで

表 4.14: 負例のフィルタリングの評価 (開発データ D_{10})

	精度			
	site	site-link	person	person-link
フィルタリングなし	0.800	—	0.926	0.750
フィルタリング T	0.792	—	0.963	0.750
フィルタリング I	0.792	—	1.000	0.750
	再現率			
	site	site-link	person	person-link
フィルタリングなし	0.588	—	0.735	0.833
フィルタリング T	0.559	—	0.765	0.833
フィルタリング I	0.559	—	0.765	0.833
	F 値			
	site	site-link	person	person-link
フィルタリングなし	0.678	—	0.820	0.789
フィルタリング T	0.655	—	0.852	0.789
フィルタリング I	0.655	—	0.867	0.789

きず, サイトの説明文とはいえないため, このテキストを支配する DOM ノードの分類クラスは other である. タイトル素性が反映されたテキストがサイト情報であるか否かを識別する素性として, サイト情報を示唆するキーワード, サイトの説明文に頻出する n-gram の二種類を導入したが, それでもサイト情報か否かを識別できない事例が存在した.

図 4.1: サイト情報抽出の失敗例

作成者情報を正しく抽出できなかった事例として, 提案手法が作成者以外のプロフィールを作成者情報に分類してしまうことがあった. このようなブログの例を図 4.2 に示す. この例では, ペットである猫のプロフィールを作成者情報として誤抽出してしまっている. このように, ペットなどの動物が人物のように紹介されている事例が存在したため, 作成者情報を正しく抽出できなかったブログが存在した. また, ペット以外にも, 家族や知人がプロフィール欄に記載されているブログも存在した.

負例のフィルタリングでは, 負例だけでなく正例も誤って削除されたことで, 評価値が低下したと考えられる事例があった. そこで, 正例が誤って削除されてしまう原因を調査した. 原因の一つは, コンテンツ領域の検知の失敗である. 3.4.1 項に記載したアルゴリズムでは, DOM ノードのテキスト長を手掛かりにコンテンツ領域を検出する. しかし, 図 4.3

図 4.2: 作成者情報抽出の失敗例

のように, コンテンツ領域の大部分が画像で満たされており, テキストがほとんど記載されていないブログでは, コンテンツ領域を検出できていないものが存在した. 本研究では, このようなブログのコンテンツ領域を検出するため, 画像のサイズをテキスト長に換算する手法を実行している. しかし, 画像のサイズが記載されていない DOM ノードもあったため, このような検出ミスが発生した.

また, 図 4.4 のように, 非コンテンツ領域に多くの画像が添付されている場合, この領域をコンテンツ領域と検知してしまう場合があった. DOM ノードにこれらの画像のサイズが記載されている場合, 画像の大きさをテキスト長に換算し, DOM ノードのテキスト長を大きく見積もってしまったため, コンテンツ領域と誤って判定された. 本研究で用いたコンテンツ領域検出アルゴリズムは, テキスト長に基づく単純な手法である. テキストの内容やレイアウト情報などを利用するようなより洗練された手法を用いて, コンテンツ領域検出の正解率が向上すれば, 負例のフィルタリングも有効に働くと考えられる.

図 4.3: コンテンツ領域検知の失敗例 1

図 4.4: コンテンツ領域検知の失敗例 2

第5章　結論

5.1　本研究のまとめ

本研究では，サイト情報, 作成者情報を自動抽出する手法を提案した．これらの情報の抽出には, HTML ファイルにおける DOM の個々のノードが上記の情報を含むか否かを機械学習により判定する手法を用いた．機械学習アルゴリズムは SVM を使用し, 8 種類の素性を用いた．また, 負例のフィルタリングを実行し, 正例と負例のバランスを是正することで, SVM の評価値の向上を図った．訓練データ及びテストデータを用意し, 提案手法の有効性を評価する実験を行った．DOM の分類クラスは 8 種類提案したが, 実験では出現頻度の小さい分類クラスを除いて 4 種類に絞り込んだ．実験の結果, 訓練データにおける 10 分割交差検定による評価, テストデータでの評価ともに提案手法がベースラインシステムを上回った．さらに, 提案した素性の評価も行った．その結果, id, class の属性値の素性はテストデータ, 開発データともに F 値が向上したため, サイト情報・作成者情報抽出のための素性として有効であることがわかった．しかし, 一部の素性については, テストデータと開発データで実験結果の違いが見られたため, その有効性を明確に確認できなかった．さらに, テストデータと開発データについて, 負例のフィルタリングの評価を行った．テストデータについては, 負例のフィルタリングが有効に働いていることを確認できた．開発データについては, 負例のフィルタリングが作成者情報抽出の評価値の向上に貢献していることがわかったが, サイト情報抽出の評価値は低下した．また, 抽出に失敗した要因の考察も行った．その結果, 負例を正例と判定してしまうケースや, 負例のフィルタリングにより正例が削除されてしまうケースが存在することがわかった．

5.2　今後の課題

本研究では, メインコンテンツのフィルタリングにより, 正例が削除されてしまう事例が存在した．これは, コンテンツ領域が誤検出されてしまったためである．そのため, コンテンツ領域を検出するアルゴリズムの精度を高めなければならない．本研究では, 画像をテキスト長に換算する手法を実行したため, コンテンツ領域の誤検出を削減することができたと考えられる．しかし, 画像のサイズが記載されていない DOM ノードも存在し, これらのノードのテキスト長は一律に 100 と定めたことによって, コンテンツ領域が誤検出されたブログも存在した．本研究では, このような画像のテキスト長を 100 としたが, より良い換算の手法を考案する必要がある．また, 負例のフィルタリングにより多くの負例を

削除したものの, まだ負例の数が正例に比べて圧倒的に多い. このことが, 提案手法による抽出精度が低い要因と考えられる. そのため, 3.4 節に記載した手法以外で, 負例を削除できる手法について考えたい.

また, 3.3 節に記載した素性で, サイト情報を示唆するキーワードや, サイト情報へのリンクを示唆するキーワードは, 4.4.2 項の実験では効果的な素性であることが証明されなかった. 提案した素性が効果的であるか否かをより多くのデータを用いてさらに調査し, 素性として採用するかを検討しなければならない. さらに, 新たな素性を追加することも検討したい.

サイト情報の抽出について, 本研究ではタイトル素性の重みが 1 の DOM ノードに含まれるテキストがサイト情報であるか否かを識別する素性を二つ導入した. しかし, サイト情報を示唆するキーワードは有効性は実証されず, サイトの説明文に頻出する n-gram も, テストデータによっては効果が現れないものも存在した. ベースラインシステムの評価結果より, ブログタイトルの近辺にサイト情報が記載されていることが多いのは明らかなので, タイトル近辺のテキストがサイト情報であるかを識別できる素性を検討する必要がある.

本研究ではブログ記事を対象にサイト情報, 作成者情報の抽出を試みた. 提案手法を拡張し, 一般のウェブページからサイト情報や作成者情報を抽出する技術を確立することも今後の重要な課題である.

謝辞

本研究の進行, 本論文の作成にあたり, 丁寧なご指導を頂いた白井清昭准教授に感謝致します. 本研究において, 適切な意見や助言を頂いた池田心准教授, 長谷川忍准教授に感謝致します. また, 本研究の趣旨を理解して頂き, 議論などを通じて意見を頂いた白井研究室に所属する学生の皆様に感謝致します.

参考文献

[1] 百瀬亮, 宮崎林太郎, 渋木英潔, 森辰則. Web ページからの情報発信者の抽出におけるレイアウト情報の利用. 言語処理学会第 16 回年次大会, p.94-p.97, 2010.

[2] Yoshikiyo Kato, Daisuke Kawahara, Kentaro Inui, Sadao Kurohashi and Tomohide Shibata. Extracting the Author of Web Pages. Proceedings of the 2nd ACM workshop on Information Credibility on the WICOW '08, p.35-p.42, 2008.

[3] Giovanni Giuffrida, Eddie C. Shek, and Jihoon Yang. Knowledge-Based Metadata Extraction from PostScript Files. Proceedings of the Fifth ACM Conference on Digital Libraries(DL '00), p.77-p.84, 2000.

[4] Daisuke Kawahara, Sadao Kurohashi, and Kentaro Inui. Grasping Major Statements and their Contradictions Toward Information Credibility Analysis of Web Contents. IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, p.393-p.397, 2008.

[5] Nozomi Kobayashi, Kentaro Inui, and Yuji Matsumoto. Extracting Aspect-Evaluation and Aspect-of Relations in Opinion Mining. Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, p.1065-p.1074, 2007.

[6] K. Tateishi, T. Fukushima, N. Kobayashi, T. Takahashi, A. Fujita, K. Inui, and Y. Matsumoto. Web Opinion Extraction and Summarization Based on Viewpoints of Products, In IPSJ SIGNL Note 163, p.1-p.8, 2004.

[7] Razvan Bunescu. Associative Anaphora Resolution: a Web Based Approach. In Proceedings of the EACL Workshop on the Computational Treatment of Anaphora, p.47-p.52, 2003.

[8] Chih-Chung Chang and Chih-Jen Lin. LIBSVM : a Library for Support Vector Machines. ACM Transactions on Intelligent Systems and Technology, Vol.2, No.3, Article 27, 2011.

付 録A　サイトの説明文に頻出する上位100件のn-gram

表 A.1: サイトの説明文に頻出するn-gram(上位1〜50件)

出現数	3-gram	出現数	3-gram
2992	て, い, ます	248	を, 書いて, い
2138	い, ます, 。	242	いき, ます, 。
1524	し, て, い	216	届け, し, ます
868	し, ます, 。	205	日々, の, 出来事
761	て, ます, 。	194	し, ます, !
588	紹介, し, て	191	て, おり, ます
557	ブログ, です, 。	169	い, ます, !
539	を, 紹介, し	167	&, amp, ;
509	・, ・, ・	161	で, い, ます
455	を, 中心, に	158	を, 目指し, て
435	て, いき, ます	156	ご, 紹介, し
426	し, て, ます	154	ませ, ん, か
403	紹介, し, ます	152	綴り, ます, 。
353	まし, た, 。	149	し, て, いる
315	の, ブログ, です	148	の, 日々, の
308	を, ご, 紹介	148	て, ます, !
281	綴っ, て, い	145	し, まし, た
281	日記, です, 。	143	の, 日常, を
266	書い, て, い	139	を, 綴り, ます
264	を, し, て	138	の, 情報, を
262	お, 届け, し	137	中, です, 。
261	書い, て, ます	133	の, 日記, です
261	を, お, 届け	130	あり, ます, 。
260	を, 綴っ, て	130	おり, ます, 。
260	し, て, いき	128	の, こと, 、

表 A.2: サイトの説明文に頻出する n-gram(上位 51〜100 件)

出現数	3-gram	出現数	3-gram
126	の, ため, の	98	に, なっ, て
125	の, ブログ, 。	97	し, て, おり
122	気, に, なる	95	情報, を, お
121	、, 日々, の	95	の, こと, を
119	やっ, て, ます	94	を, 発信, し
118	を, 楽しん, で	93	を, 公開, し
118	に, し, て	92	も, あり, ます
116	ん, か, ?	90	作っ, て, い
115	情報, を, 発信	89	を, 目指し, ます
114	ご, 紹介, 。	88	綴っ, て, ます
114	お伝え, し, ます	88	に, 来, て
112	を, 紹介, 。	88	お, 得, な
111	更新, し, て	86	発信, し, て
110	公開, し, て	86	始め, まし, た
110	ブログ, で, す	85	です, が, 、
107	の, 出来事, を	82	を, 紹介, !
103	を, お伝え, し	81	掲載, し, て
103	に, なっ, た	80	など, を, 紹介
102	と, 思い, ます	79	の, 生活, を
100	楽しん, で, い	79	まし, た, !
100	更新, 中, !	79	て, ください, 。
99	毎日, 更新, 中	74	ませ, ん, 。
99	に, 書い, て	74	た, こと, を
99	を, 作っ, て	73	記録, です, 。
99	と, し, た	73	の, 写真, を