修士論文

ページ送りで掲載されたウェブコンテンツの自動抽出

花村　直親

主指導教員　白井　清昭

北陸先端科学技術大学院大学
先端科学技術研究科
（情報科学）

令和２年６月

Abstract

Pagination on the Web is a process to divide textual contents into several pages and show each page on a discrete Web page. It is a useful way to publish long documents. Users first see a moderate amount of a text in the first Web page, then they can choose whether they follow the second page to see all of document. However, paginated Web sites are problematic for Web mining, which aims at analyzing a lot of Web pages and acquiring useful knowledge, since documents are divided into several pages. To precisely analyze documents on the Web to discover new knowledge, separated pieces of texts in the paginated Web sites should be restored to the original single document. In the past studies on Web mining, pagination has not been paid much attention. AutoPagerize is the plug-in of a Web browser that can automatically concatenate paginated contents and show it as a single document. However, since the concatenation of contents is relied on the hand-crafted rules in the Wiki-like database "Wedata", it is applicable for only 8,000 paginated Web sites in the Wedata. For the practical Web mining, it is required to automatically restore paginated contents in not limited but all Web sites.

The goal of this thesis is to propose a method to automatically extract contents in any paginated Web sites as a single document. It enables us to process paginated Web sites more flexibly for many purposes, such as information extraction, opinion mining, and so on. While AutoPagerize relies on the manually created rules, this study applies supervised machine learning to obtain models that automatically extract the contents from any paginated Web sites.

Our proposed method consists of three modules: the module to extract the link to the next page, to extract the main content, and to concatenate the extracted main contents. Here the "main content" means the most important content such as texts and images in a Web page, other than less informative texts such as a navigation link and advertisement. For a given paginated Web site, the first module finds the link to the next page, while the second module extracts the main contents. By applying these modules repeatedly, all main contents in the paginated Web pages can be extracted. The third module concatenates them as a single document. Since the process of the third module is obvious, this thesis only focuses on the first and second modules.

In the first module, the link to the next page is extracted as follows. All internal links, i.e. URLs to other Web pages in the same domain, are extracted from a given Web page. Then, a classifier that judges whether each link refers to the next page is trained by supervised machine learning. The features for machine learning are: (1) existence of the keyword "next" in an anchor text, (2) existence of the keyword "page" in an anchor text, (3) whether an anchor text consists of one character,

(4) frequency of the link in a Web page, (5) length of an anchor text, (6) relative length of an anchor text, (7) length of a URL, (8) relative length of a URL, and (9) LinkSimilarity. The last feature LinkSimilarity evaluates the similarity between a target link and its neighbor links. Since the training data is extremely imbalanced, i.e. the number of positive samples (the link to the next page) is much less than negative samples (the link to the other page), Synthetic Minority Over-sampling (SMOTE) is applied to make the training data modestly balanced. Finally, using the above features, the classifier is trained from the balanced training data. Three machine learning algorithms are applied: Decision Tree, Random Forest, and Gradient Boosting Decision Tree (GBDT).

In the second module, the main content is extracted as follows. First, DOM (Document Object Model) tree of a given Web page is obtained, then all nodes in the DOM tree, which correspond to HTML tags, are extracted. Hereafter, a DOM node is simply called "tag". Then, a classifier that judges whether each tag contains the main content of the Web page is trained by supervised machine learning. The features for machine learning are: (1) length of the tag, (2) depth of the tag in the DOM tree, (3) position of the tag in the HTML file, (4) relative position of the tag in the HTML file, (5) the tag is a block element or not, (6) the tag obviously suggests non-main contents or not, (7) length of texts in sibling tags in the DOM tree, (8) proportion of texts in sibling tags, (9) amount of punctuation in sibling tags, (10) text density of sibling tags, (11) number of sibling tags, (12) number of child tags, (13) proportion of the number of child tags, and (14) distance to the link to the next page in the DOM tree. To extract the last feature, the link to the next page is identified by the aforementioned first module. In addition, the imbalanced training data is converted to the totally balanced data consisting of equal number of the positive samples (tags that include the main content) and negative samples (tags that do not include the main content) by the over-sampling method SMOTE and the under-sampling that randomly removes negative samples. Finally, the classifier is trained by Decision Tree, Random Forest, and GBDT.

Several experiments are conducted to evaluate our proposed method. A collection of Web pages annotated with the links to the next pages and the main contents is constructed from Wedata, then it is divided into the training and test data. Our systems are compared with the baselines that extract the link to the next page or the main content by simple heuristic rules.

Precision, recall, and F-measure of our best model for extraction of the link to the next page are 0.818, 0.692, and 0.750, respectively. It outperforms the baseline of which F-measure is 0.607. Furthermore, the F-measure is improved by 0.027 points by the LinkSimilarity feature that is specially designed by considering characteristics of pagination. Precision, recall, and F-measure for the extraction

of the main content are 0.588, 0.555, and 0.571, respectively. It also outperforms the baseline of which F-measure is 0.003. In addition, the F-measure is improved 0.07 points by the feature of the distance to the link to next page. It indicates that the proximity to the link to the next page is an effective feature to extract the main content in paginated Web pages. Since our models significantly outperform the baselines, the effectiveness of our proposed method is confirmed.

概要

ウェブサイト上で長い記事を掲載するときにはページ送りがよく使われる。ウェブにおけるページ送りとは、長い記事をページ番号を付けていくつかのページに分割して掲載することを指す。1 ページにおさまりきらない記事を分割することで、初めのウェブページの読み込み時間を短縮し、ユーザーは最初に表示されたページの内容を見た後、次ページへ遷移して続きを読むかを判断できる。ページ送りはユーザが閲覧する際は便利だが、ウェブから情報を自動的に獲得する際には、複数のページに分割された記事から元の記事全体を復元する必要がある。ページ送りが使われているウェブサイトに対して元の記事を復元する試みとして AutoPagerize がある。AutoPagerize は、8000 件程度のウェブサイトに対してあらかじめ人手で作成された連結規則が登録されているデータベース Wedata に基づいて機能するため、登録されていないウェブサイトについては元の情報を復元できないという問題がある。

本研究は、大量のウェブページから知識を獲得するウェブマイニングのための基礎技術として、ページ送りによって複数のページに分割された記事を自動的に 1 つに連結することを目的とする。AutoPagerize が人手で連結規則を作成するのに対し、本研究は教師あり機械学習によって次ページへのリンクと主コンテンツを自動検出するモデルを学習し、任意のウェブサイトに対応する点に特徴がある。

本研究の提案手法は、「次ページリンク検出タスク」、「主コンテンツ検出タスク」、「連結タスク」を処理する 3 つのモジュールから構成される。「次ページリンク検出」モジュールは、ページ送りのあるウェブページ内から次のページへのリンクを検出する。ウェブページの HTML ソースファイルから同一ドメインへのリンクを抽出し、機械学習されたモデルを適用して、それぞれのリンクが次のページへのリンクに相当するかを判定する。「主コンテンツ検出」モジュールは、ウェブページの HTML ソースファイルと検出された次ページリンクを入力とし、機械学習されたモデルを適用して、個々のタグが主コンテンツに該当するかを判定する。これら 2 つのモジュールは繰り返し適用される。検出した次ページリンクを辿ることで次ページの HTML ソースファイルを取得し、これを新たな入力として次ページリンクと主コンテンツを再起的に検出する。最終的に獲得された複数の主コンテンツを「連結」モジュールで連結する。最後のモジュールは単純な処理であるため、本研究では最初の 2 つのモジュールの開発、特に次ページリンクと主コンテンツを判定する分類器の機械学習に注力する。

次ページリンクを検出する分類器を学習する際には、素性として、(1)「次」もしくは「NEXT」がタグに含まれるか、(2)「ページ」もしくは「PAGE」がタグに含まれるか、(3) リンクテキストが 1 文字であるか、(4) ウェブサイトにおけるリンクの出現回数、(5) リンクテキスト長、(6) リンクテキスト長のウェブページ全体のテキスト長に対する割合、(7) リンクの URL の長さ、(8) リンクの URL の長さのウェブページ全体に対する割合、(9) 近傍のリンクとの類似性 (LinkSimilarity) を用いた。訓練データは、正例である次ページリンクが、負例であるそれ以外の

リンクに対して圧倒的に少ないため、Synthetic Minority Over-sampling(SMOTE) を用いて不均衡データを是正した後、分類器を学習する。学習アルゴリズムとして、決定木、ランダムフォレスト、Gradient Boosting Decision Tree(GBDT) を用いる。

主コンテンツを検出する分類器を学習する際には、素性として、(1) タグの長さ、(2)DOM ツリーにおけるタグの深さ、(3)HTML ファイルにおけるタグの位置、(4)HTML ファイルにおけるタグの相対位置、(5) ブロックレベル要素に該当するか、(6)HTML タグの種類が明らかに主コンテンツにならないものであるか、(7) 兄弟タグ内にあるテキストの長さ、(8) 兄弟タグ内にあるテキストの割合、(9) 兄弟タグ内の句読点の割合、(10) 兄弟タグのテキスト密度、(11) 兄弟タグ数、(12) 子タグ数、(13) ウェブページ全体のタグ数における子タグ数の割合、(14) 次ページリンクタグからの距離を用いた。次ページリンクタグからの距離の素性は、前述のモジュールで検出された次ページリンクタグと判定対象のタグとの DOM ツリー上の距離を値とする。次ページリンク検出タスクと同様に、訓練データでは、正例である主コンテンツのタグが、負例である主コンテンツ以外のタグと比べて圧倒的に数が少ない。そのため、SMOTE を用いて正例を増加させた後、負例をランダムに減少させて、完全に均衡した訓練データを作成した後、分類器を学習する。学習には決定木、ランダムフォレスト、GBDT を用いる。

提案手法の評価実験について述べる。データセットとして Wedata に登録されたウェブサイトを利用する。簡易なルールに基づくベースライン手法と提案手法の性能を比較する。評価基準として精度、再現率、F 値を用いる。次ページリンクの検出モデルについては、3 つの機械学習アルゴリズムのうちランダムフォレストが最も性能がよく、精度は 0.818、再現率は 0.692、F 値は 0.750 であった。ページ送りの特徴を特に考慮した LinkSimilarity 素性によって F 値が 0.027 ポイント向上した。主コンテンツの検出モデルについては、精度は 0.588、再現率は 0.555、F 値は 0.571 であった。また、ページ送りの特徴を特に考慮した「次ページリンクからの距離」の素性を導入することで F 値が 0.07 ポイント向上した。これら 2 つの提案手法の結果は、それぞれ、ベースライン手法よりも顕著に高く、機械学習によってページ送りされたウェブサイトから主コンテンツを検出する提案手法のアプローチが有効であることが確認された。

目 次

図 目 次

表 目 次

第1章　はじめに

1.1　背景

ウェブサイト上で長い記事を掲載するときにはページ送りがよく使われる。ページ送りとは、長い記事にページ番号を付けていくつかのページに分割して掲載することを指す。ページ送りを用いたウェブページの例を図 1.1 に示す。1 ページにおさまりきらない記事を分割することで、初めの読み込み時間を短縮し、ユーザーは最初に表示されたページの内容を見た後、次ページへ遷移して続きを読むかを判断できる。

ページ送りはユーザが閲覧する際は便利だが、ウェブから情報を自動的に獲得する際には問題となる。一般に、ウェブから大量の情報や知識を獲得・収集するウェブマイニングを行う際には、個々のウェブページのソースファイルをダウンロードし、これに対して自動処理が行われることが多い。すなわち、ウェブページを単位として処理が行われる。しかし、ページ送りを使ったウェブサイトでは、本来は 1 つのページに掲載されるべき 1 つの記事が複数のページに分割されて掲載されているため、ウェブページを単位とした処理では、元の記事全体から必要な情報を抽出したり、元の記事全体を解析することができない。例えば、オピニオンマイニングは評価対象に対する書き手の意見を解析し集約する技術であるが、一つの記事が複数ページに分かれていると、分割された記事、すなわち元の記事の一部に対して書き手の意見が肯定的か否定的かを判定することになり、対象に対する書き手の意見を正確に把握できない可能性がある。

一方、ページ送りで掲載された記事を一つにまとめる試みがある。AutoPagerize[1] は、ページ送りされた記事を自動的に連結して表示するウェブブラウザのプラグインである。ページ送りは便利と感じるユーザもいれば、逆に不便と思うユーザもいる。AutoPagerize は、ページ送りを好まないユーザのために、分割された記事を 1 つに連結してブラウザ上に表示させることができる。しかし、AutoPagerize では、複数ページのテキストを連結する規則をあらかじめ登録されたウェブサイト毎に人手で作成している。すなわち、あらかじめ登録された 8000 件程度のウェブサイトに対してしか、ページ送りで分割されたウェブページを連結できない。分割された記事を 1 つに連結するルールのデータベースはオープンソースになっており、誰でも新しいウェブサイトに対して連結ルールを登録することができるが、

[1] http://autopagerize.net/

(引用元 URL
https://headlines.yahoo.co.jp/hl?a=20200507-00000020-zdn_mkt-bus_all)

図 1.1: ページ送りの例

ルールはXPathで記述する必要があるため、XPathに関する知識を持たない一般のユーザがルールを登録するのはそれほど容易ではない。以上から、AutoPagerizeをウェブマイニングに適用することは難しい。ページ送りされたウェブサイトを含む大量のウェブサイトから知識を獲得するためには、ページ送りされたウェブページを完全に自動的に連結する技術が必要である。

1.2 目的

本研究は、大量のウェブページから知識を獲得するウェブマイニングのための基礎技術として、ページ送りによって複数のページに分割された記事を自動的に1つに連結することを目的とする。

具体的には、この目的を達成するために、以下の2つの要素技術を探究する。

- ページ送りされたウェブサイトから次のページへのリンクを検出する技術
- ページ送りされたウェブサイトからテキストや画像などを含む主コンテンツを検出する技術

ここでの「主コンテンツ」とは、ウェブページにおける主たる内容を表すテキストや画像と定義する。ウェブページには、書き手によって発信された情報の他に、広告、ナビゲーションリンク (トップページへのリンクなど)、情報発信者の情報 (名称、住所など)、コピーライト表示などを含むが、主コンテンツはそれらを除いた重要な情報のみを指す。

上記の2つの要素技術により、ページ送りを使用しているウェブサイトに対し、次ページへのリンクを辿り、各ページから抽出された主コンテンツを連結することで、元の記事を1つのページとして復元することができる。

次ページへのリンクの検出および主コンテンツの抽出は、人手で作成されたラベル付きデータを訓練データとし、教師あり機械学習で、次ページへのリンクもしくは主コンテンツに該当するか否かを判定するモデルをそれぞれ構築する。任意のページ送りされたウェブサイトに対して、学習された2つのモデルを適用することで、分割された記事を元の1つの記事に復元することが可能になる。

1.3 本論文の構成

本論文の構成は以下の通りである。第2章ではウェブサイトから主要なコンテンツを検出する既存の研究を外観し、本研究の立場を明らかにする。第3章では、本研究の提案手法について述べる。次ページリンクの検出手法と主コンテンツの検出手法について、機械学習アルゴリズム、機械学習のための素性、不均衡データに対する対応方法を詳述する。第4章では、提案手法の評価実験について述べ

る。実験の手続きと結果を報告し、提案手法の有効性について議論する。最後に、第5章では、本論文の成果を総括し、今後の研究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、ページ送りされているウェブサイトに対して、ページ分割された記事を連結する研究を紹介する。2.2 節では、ウェブページから主コンテンツを抽出する先行研究を紹介する。最後に、2.3 節では、本研究と先行研究の違いについて論じる。

2.1　ページ送りを対象とした研究

沢田らはページ送りによって複数のページに分けて掲載された記事を一つに連結するツール AutoPagerize を提案した [1]。AutoPagerize は、処理の対象とするウェブサイト毎の連結規則が収集されたデータベース Wedata[2] に基づいて機能する。

Wedata に記載される情報を表 2.1 に示す。url, nextLink, pageElement, exampleUrl, insertBefore の 5 つの属性で一つのウェブサイトに対する連結規則を表す。url は規則を適用するウェブサイトを指定する。nextLink はページ送りされている次のページへのリンクを特定する。pageElement は連結するべき記事本文を特定する。exampleUrl はこのルールを適用するウェブページの URL の例を表す。insertBefore は、連結したテキストをウェブブラウザに表示する際、次ページに掲載された記事を挿入するべき場所を指定する。Wedata は、インターネット上の誰でも編集可能な集合知データベースとして構築・運用されている。

AutoPagerize は、Wedata に登録された規則に基づいて、ページ送りされた主コンテンツを自動連結する。ウェブブラウザで表示しているウェブページの URL が属性 url の正規表現にマッチするとき、nextLink によって次のページへの URL を特定し、それを取得する。さらに、取得した次ページから、pageElement によって指定される主コンテンツを抽出する。そして、それを現在のページの主コンテンツに連結して表示する。以上の操作で、ページ送りによって複数のウェブページに分割して掲載された記事を、リンクを辿ることなく閲覧することができる。

AutoPagerize は、Wedata に登録されている既知のウェブサイトに対して自動連結を行うツールとして非常に役立つが、連結規則が登録されていないウェブサイトに対して適用することができないという問題がある。1.1 節で述べたように、ウェブマイニングのように大量のウェブページから知識や情報を自動獲得する際には、任意のウェブサイトに対して、ページ送りで表示された記事を連結することが求

表 2.1: wedata に格納されている情報

属性	必須	内容	例
url	○	ルールを適用する対象となるウェブサイトの URL の正規表現	https?://deskgram.org/*
nextLink	○	次のページの URL を示す要素を指定する XPath 式	//div[@id=”loadmoreimg”]/a
pageElement	○	ページ本体 (連結するテキストを含む要素) を指定する XPath 式	//div[@id=”posts-container”]
exampleUrl		対象ウェブサイトの URL の一例	http://deskgram.org/instagram
insertBefore		次のページのテキストを挿入する箇所を指定する XPath 式	//div[@id=”loadmoreimg”]

められる。また、ウェブサイトのレイアウトが変更されたときは、連結規則もそれに合わせて修正する必要がある。すなわち、Wedata のデータベースは人手でメンテナンスする必要があり、このメンテナンスにコストがかかるという問題がある。

2.2 主コンテンツの検出に関する研究

本研究はページ送りによって複数のウェブページに分けて掲載された記事を自動的に連結することを目的とするが、その際、ウェブページの中から連結するべきテキストや画像、すなわちウェブページの主コンテンツを自動的に検出する処理を行う。これまでにも、ウェブサイトから主コンテンツを検出する研究は数多く行われている。

Uzun らは、ウェブ上の新聞記事サイトから有用な文書コンテンツ（本文）を自動的に抽出する手法を提案した [3]。本文を含む HTML タグをアノテーションしたデータを訓練データとし、id,class などの属性値、タグ内テキストの単語数、リンクの情報などを素性とした教師あり機械学習により本文を抽出した。機械学習アルゴリズムとして、複数のアルゴリズムをいくつか実験的に比較した結果、subtree raising による剪定を行う決定木アルゴリズムを採用した。機械学習の素性を抽出する際には、div タグが入れ子になっている場合など、他の要素を囲んでいるだけのタグについてはテキストに関する素性が抽出できないため、ページ全体の DOM

ツリーにおいて最も深いタグから素性を抽出する「After Extraction」という手法を提案した。提案手法による主コンテンツを含むタグの分類性能を実験により評価したところ、精度は 0.950、再現率は 0.950、F 値は 0.950 であった。さらに、主コンテンツの検出に有効だった素性として、「全タグの単語数に対する a タグ内の単語数の割合」(0.56)、「After Extraction 実施後の全タグの単語数に対する a タグ内の単語数の割合」(0.38)、「td, div, h1 – h6, p, font, a, span, em, ul, li のいずれかのタグに該当するか」(0.37) などをあげている。括弧内の数値は素性の利得で、決定木アルゴリズムにおいて素性が分類に貢献した指標である。「After Extraction」による最も深いタグからの素性抽出が有効であることを示した。

Zeng らは、EC サイトの検索結果のページから主要なコンテンツを抽出する LTDE(Layout Tree based Data Extraction) と呼ばれる手法を提案した [4]。EC サイトの多くは自社の製品データベースに対する検索機能を持ち、検索結果のページは複数の製品の情報が並べて表示されることが多い。LTDE は、検索結果のページにおける繰り返し構造を手がかりに製品の情報を抽出する手法である。LTDE では、まずウェブサイト内で画面表示サイズが指定されている HTML タグをブロックとみなし、ブロック内部のレイアウトを元にレイアウトを表す木構造を作成・抽出する。抽出された複数の木構造間の類似度を Tree Edit Distance を用いて測り、類似構造をクラスタリングする。その後、各クラスタにおけるレイアウト情報やブロック数による重みづけなどの情報を手がかりに、主となる商品ブロックを含むクラスタを検出する。評価実験では、主コンテンツに該当するレコード検出の精度、再現率、F 値を計測した。その結果、Javascript を使わない静的ページに対する主コンテンツ検出については、精度 0.993、再現率 0.977、F 値 0.985 となり、Javascript によって動的生成されたページに対しては、精度 0.989、再現率 0.961、F 値 0.975 といった結果が得られた。また、Javascript によって動的生成されたページを対象としたとき、ウェブページの HTML ソースファイルの構造のみに基づく先行研究の手法よりも、LTDE による主コンテンツの抽出性能が高いことを示した。

Song らは、ウェブページ内の HTML タグが主コンテンツを含む可能性の高さを示す指標を、テキスト密度と視覚的な重要度の組み合わせによって算出する手法を提案し、最も指標が高いタグによって囲まれたテキストを主コンテンツとして検出する手法を提案した [5]。この手法では、HTML ソースファイルにおける個々の HTML タグに対し、自身を含んだタグが内包するタグの総数に対し、タグが内包する文字数の割合をテキスト密度として定義し、テキスト密度の高いタグが重要なコンテンツと考える。さらにリンクが少なくテキストが多いタグや、水平画面表示サイズが大きいタグをより重要だと重み付けすることで、主コンテンツ検出の性能が向上することを実験的に示した。評価実験では、テキストが主たるコンテンツとなっているサイトを対象とし、提案手法の精度は 0.962、再現率は 0.961、F 値は 0.962 であったと報告している。

Wu らは、まず機械学習によって主コンテンツを表す DOM ノードの候補を検出

し、次にそれらをグループにまとめるという二段階の手続きで、ウェブページ内から主コンテンツを検出する手法を提案した [6]。この手法では、まずウェブページ内の DOM ノードについて主コンテンツに該当する可能性を教師あり機械学習によってスコアリングする。次に、あらかじめ定められた閾値以上のスコアの DOM ノードを主コンテンツ候補の DOM ノードとする。検出した主コンテンツ候補の DOM ノード群を、水平方向と垂直方向の位置レイアウトが一定以上の重なりを持っていることや、親タグが同一であることなどの条件により、複数のグループに分ける。最後に、作成されたグループのうち、機械学習のスコアの平均値と、そのグループに属するタグが占める領域の積算値が最も高いグループを主コンテンツとして検出する。評価実験では、多様なウェブページを対象に、検出した主コンテンツの領域と正解の領域の重なりを F 値で測り、評価指標とした。SVM を単純に適用して主コンテンツを検出するベースライン手法の F 値が 0.7 程度であったのに対し、提案手法の F 値は 0.8 を越えた。機械学習による主コンテンツの DOM ノードの検出と検出された DOM ノードのグループ化の組み合わせが有効であることを実験的に示した。

吉田らは、ニュースページから記事に相当する主コンテンツを検出する教師なし機械学習手法を提案した [7]。この手法では、検出対象は W3C[8] に定義されているブロックレベル要素のタグに限定する。ブロック内のタグ、テキスト、title 属性のテキスト、alt 属性のテキストをベクトルとして表現し、同じウェブサイト内の別ページに出てくるブロック同士のコサイン類似度を計算する。コサイン類似度が 0.9 を超えるブロックは同一内容とみなす。ニュースサイトにおいて、複数ページに同一の内容が出てくるブロックは広告やタイトルバーなど主コンテンツではない要素だと判断して除外する。3 種類のウェブサイトをテストデータに用いた評価実験では、記事 (主コンテンツ) に相当するブロック検出の精度は 0.980、再現率は 0.911、F 値は 0.945 となった。

2.3 本研究の特徴

本研究は、ページ送りされているウェブサイトに対して、次ページへのリンクと主コンテンツを検出し、複数ページに分かれて掲載されている記事を自動的に 1 つの記事に連結する点に特徴がある。ページ送りが使われているウェブサイトに対して元の記事を復元するという目的は沢田らの研究 [1] と同じであるが、沢田らの研究は集合知により各ウェブサイトに対して連結規則を人手で作成してあらかじめ収集するのに対し、本研究では教師あり機械学習によって次ページへのリンクと主コンテンツを自動検出するモデルを学習し、任意のウェブサイトに対応する点が異なる。

主コンテンツを検出するという目的については、先行研究がページ送りを含むウェブページを処理の対象としていないのに対し、本研究ではページ送りされているウェブページを対象としている点に違いがある。Uzun らの研究 [3] や吉田ら

の研究 [7] では新聞のウェブサイトを、Zeng らの LTDE[4] は EC サイトにおける製品の検索結果のページを主な対象としており、ページ送りを対象としたものではない。一方、ページ送りは新聞や EC サイトに限らず幅広いウェブサイトで使われており、テキスト、画像、動画など様々な要素を含み、ウェブページのレイアウトも多様性に富む。本研究で対象とするウェブページは、ページ送りされていることを前提としてはいるが、様々なウェブサイトを対象とするという点では主コンテンツの検出が難しく、そのような難しい問題に取り組んでいるという点に特徴がある。

一方、ページ送りされているウェブサイトから主コンテンツを抽出する際には、次ページへのリンクが手がかりになると考えられる。次ページへのリンクは主コンテンツの近くに (多くの場合は主コンテンツの下に) 配置されることが多いことから、主コンテンツを表す DOM ノードと次ページへのリンクを表す DOM ノードの関連性を機械学習の素性として利用することが考えられる。次ページへのリンクを素性として利用することで、ページ送りされていないウェブページから主コンテンツを抽出する場合と比べて、より正確に主コンテンツを検出できる可能性がある。本研究では、主コンテンツ検出タスクにおいて次ページへのリンクを利用する効果を実験的に検証する。

第3章　提案手法

本章では、ページ送りされた任意のウェブサイトに対して、複数のウェブページに分割して掲載された記事をひとつにまとめる手法を提案する。3.1 節では、提案手法の概要を述べる。3.2 節では次ページリンクの検出手法について、ウェブページからのリンクの抽出方法、機械学習の素性、不均衡データへの対応、機械学習アルゴリズムを述べる。3.3 節では主コンテンツの検出手法について、タスクの定義、機械学習の素性、不均衡データへの対応、機械学習アルゴリズムを述べる。

3.1　概要

提案手法の処理の流れを図 3.1 に示す。ページ送りされたウェブサイトの 1 ページ目を入力とし、元の記事を分割して掲載している個々のウェブページから抽出した主コンテンツを連結して、最終的にひとつの記事として出力する。

提案手法は以下の 3 つのタスクを処理するモジュールから構成される。

タスク 1 次ページリンク検出タスク

タスク 2 主コンテンツ検出タスク

タスク 3 単純連結タスク

タスク 1「次ページリンク検出タスク」は、与えられたウェブページ内のリンクから次ページへのリンクを検出するタスクである。通常ウェブページ内には多数のリンクが存在するが、ここで検出したい次ページリンクは外部のウェブサイトへのリンクではなく、同一ドメインへのリンク (同じウェブサイトの別ページへのリンク) であることは明らかである。そのため、与えられたウェブページの HTML ソースファイルから同一ドメインのリンクのみを抽出する。これらのリンクに対し、機械学習モデルを適用して、それぞれのリンクが次のページへのリンクに該当するかを判定する。また、後続のタスク 2 において、主コンテンツ検出モデルの素性として利用するために、次ページリンクに相当するリンクタグ（a タグ）をラベリングする。

タスク 2「主コンテンツ検出タスク」は、ウェブページ内の主コンテンツを検出するタスクである。ここでは、タスク 1「次ページリンク検出タスク」の出力結果、すなわち次ページのリンクがタグ付けされたウェブページの HTML ソースファイ

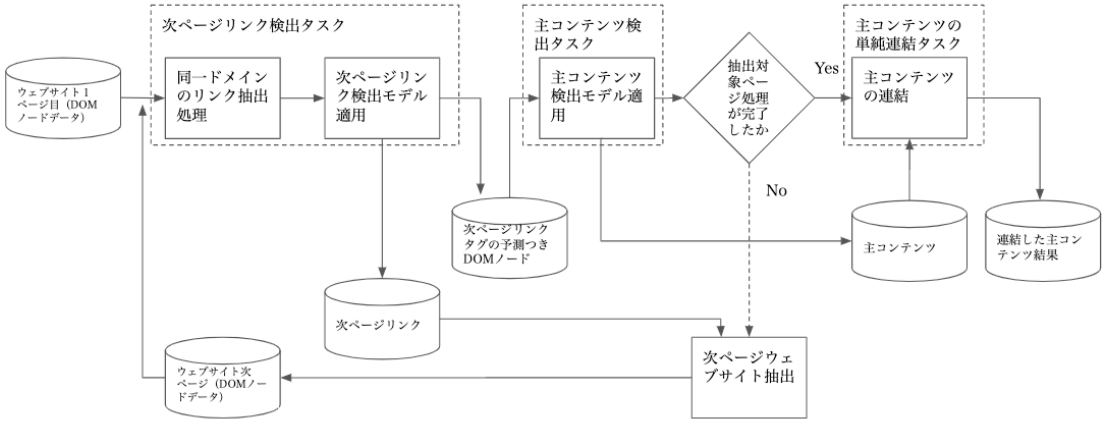


図 3.1: 提案手法の概要

ルを入力とする。このソースファイルに含まれる個々の DOM ノードに対し、それが主コンテンツに該当するかを教師あり機械学習によって学習されたモデルを用いて判定する。最終的に主コンテンツに相当するテキスト、画像を出力する。

タスク 1「次ページリンク検出タスク」とタスク 2「主コンテンツ検出タスク」は繰り返し行われる。タスク 1 で検出した次ページリンクを辿って、次ページの HTML ソースファイルを取得する。これを新たな入力として、再起的にタスク 1 とタスク 2 を繰り返し実行し、ページ送りによって分割された複数のウェブページのそれぞれから主コンテンツを抽出する。

タスク 3「単純連結タスク」では、タスク 1「次ページリンク検出タスク」とタスク 2「主コンテンツ検出タスク」を繰り返し実行して得られた複数の主コンテンツを連結する。

タスク 1 とタスク 2 において、次ページリンクならびに主コンテンツを検出するためのモデルは、教師あり機械学習によって獲得する。以下、それぞれの詳細を説明する。

図 3.2 はタスク 1 における次ページリンク検出モデルの学習の流れを示している。ページ送りを含むウェブページに対して次ページリンクがタグ付けされたデータを入力とし、同一ドメインへのリンクのみを抽出したデータを訓練データとして用いる。次ページリンクを検出するために有効な素性を設計し、訓練データからこれらの素性を抽出し、検出モデルを機械学習する。また、タスク 1 では、ページ内のそれぞれのリンクが次ページリンクに該当するかを判定するが、実際のデータでは次ページリンクに該当するリンクは該当しないリンクよりも著しく数が少ないため、訓練データにおける次ページリンクの有無のラベルに大きな偏りがある。そのため、不均衡データの是正をした上で次ページリンク検出モデルを学習する。

一方、図 3.3 は、タスク 2 における主コンテンツ検出モデルの学習の流れを示

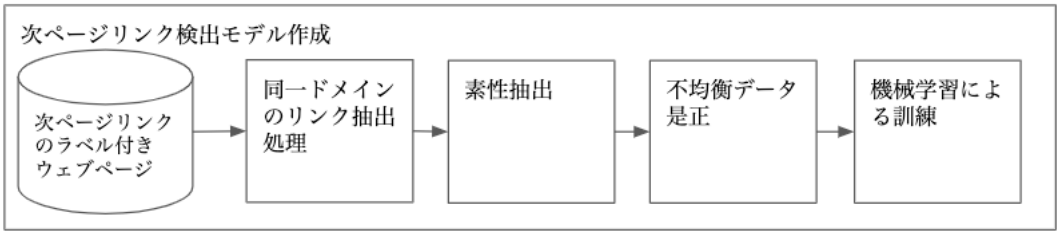


図 3.2: 次ページリンク検出モデルの学習の流れ

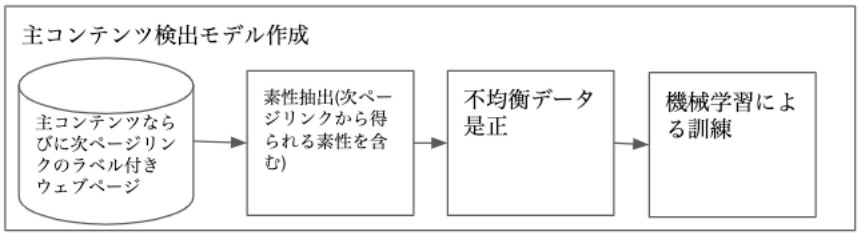


図 3.3: 主コンテンツ検出モデルの学習の流れ

している。訓練データとして、ページ送りを含むウェブページに対し、主コンテンツに相当する DOM ノードならびに次ページへのリンクがタグ付けされたデータを訓練データとする。主コンテンツを検出するのに有効な素性を設計し、訓練データからこれらの素性を抽出し、検出モデルを機械学習する。このとき、素性には次ページへのリンクも含まれるため、訓練データとして主コンテンツだけではなく次ページリンクもタグ付けされたデータが必要である。また、タスク 2 では、HTML ファイルの各 DOM ノードが主コンテンツであるか否かを判定するが、タスク 1 と同様に、主コンテンツに該当する DOM ノードは該当しない DOM ノードよりも著しく数が少ない不均衡データを訓練データとする。そのため、不均衡データを是正する処理を行ってから検出モデルを学習する。

次節以降では、タスク 1 とタスク 2 について、特に検出モデルの機械学習手法を中心に詳細を述べる。なお、タスク 3 についてはタスク 1 とタスク 2 の結果を単純に連結するタスクであるため、本論文ではこれ以上の詳細な説明を省略する。

3.2 次ページリンクの検出

本節では、ページ送りを含むウェブページから次ページリンクを検出する手法について述べる。

3.2.1 リンクの抽出

まず、ウェブページに含まれるaタグを検出し、別のページへのリンク先のURLを全て取得する。一般に、1つのウェブページにおいて、同じページへのリンク(aタグ)は複数存在することがあるが、ここではリンク先URLが重複している場合にはこれらをマージする。すなわち、重複をマージした上で、あるウェブページからリンクされている別ページのURLのリストを作成する。

既に述べたように、次ページリンクのリンク先は同一ドメインへのウェブページに限られる。そのため、作成したリンク先ページのURLのリストから、URLのドメインが入力ウェブページのURLのドメインと一致しないものを削除する。すなわち、入力ウェブページからリンクされている同一ドメインへのリンク先URLのリストを得る。このリンク先URLのリストを対象に次ページへのリンクを検出する。

以下、リンク先のURLを単に「リンク」と呼ぶ。以降の処理では、上記の手続きで作成したリスト内におけるそれぞれのリンクに対し、それが次ページリンクに該当するかを判定する。

3.2.2 素性抽出

通常の機械学習のアプローチと同様に、次リンク検出モデルの機械学習に用いる素性を抽出し、判定対象のリンクを素性ベクトルで表現する。本研究では、ページ送りされたウェブサイトにおける次ページリンクの特徴を分析し、次ページリンクか否かの判定に有効と考えられる素性を設計した。

本研究で提案する次ページリンク検出モデルの素性の一覧を表3.1に示す。これらの素性は、リンク自体の素性、すなわちリンクを含むaタグから抽出される素性(表3.1における(1)〜(8))と、他のリンクの関係性を考慮した素性(表3.1における(9))に分けられる。前者の素性は3.2.2.1で、後者の素性は3.2.2.2で、それぞれ詳述する。

3.2.2.1 リンク自体の素性

(1)「次ラベル」の素性は、リンクを定義するaタグに次ページへのリンクを示唆するキーワードが含まれているか否かを表す素性である。次ページリンクを含むaタグは、次ページであることがわかるキーワードによって表されていることが多い。例えば、図3.4は、ページ送り(網掛け部に示す)を用いているウェブサイトの例であるが、次ページへのリンクはユーザーにわかりやすいように「NEXT」という単語で表現されている。このように「NEXT」や「次」といったキーワードは、次ページリンクの判定に有効な手がかりと考えられる。

表 3.1: 次ページリンク検出に利用する素性

素性	説明
(1) 次ラベル	リンクを定義する a タグに「次」もしくは「NEXT」が含まれるか
(2) ページラベル	リンクを定義する a タグに「ページ」もしくは「PAGE」が含まれるか
(3) 1 文字ラベル	リンクテキストが 1 文字であるか
(4) リンク出現回数	ウェブサイトにおけるリンクの出現回数
(5) テキスト長	リンクテキストの長さ
(6) テキスト長の割合	リンクテキスト長のウェブページ全体のテキスト長に対する割合
(7) リンク長	リンク URL の長さ
(8) リンク長の割合	リンクの URL の長さのウェブページ全体に対する割合
(9)LinkSimilarity	そのリンクの近傍にある別のリンクとの類似性

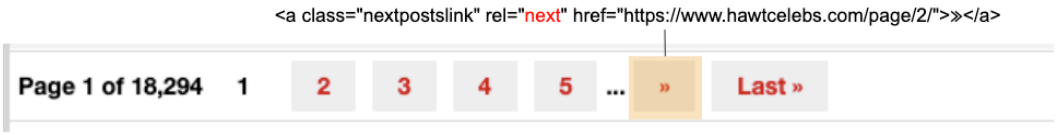
図 3.5 は別のページ送りの例である。このウェブページにおける次ページへのリンクは「≫」と表示されているタグであるが、図 3.4 のように「次」や「NEXT」といった次ページのリンクであることを示唆するキーワードはブラウザ上には現れていない。しかしながら、この次ページリンクを表す a タグの rel 属性は「next」であり、これもまた次ページへのリンクであることを示唆する。この例のように、ウェブブラウザ上に表示されるリンクテキストの他に、HTML タグの属性にも次ページへのリンクであることを示唆するキーワードが含まれることがある。

以上を踏まえ、(1)「次ラベル」の素性は、リンクを定義する a タグに「次」ならびに「NEXT」が含まれるか否かを表すものと定義する。この素性はバイナリ素性であり、「次」「NEXT」が含まれるときは素性の値は 1、含まれないときは 0 とする。また、「NEXT」が含まれるかをチェックする際には大文字と小文字を区別しない。すなわち、a タグを全て大文字に変換してから「次」「NEXT」といった文字列が含まれるかをチェックする。繰り返しになるが、この素性は、リンクテキストだけでなく a タグの属性にも「次」「NEXT」というキーワードが含まれているかを表すことに注意していただきたい。

(2)「ページラベル」の素性は、リンクを定義する A タグに次ページへのリンクを示唆するキーワードが含まれているかを否かを表す素性である。ページ送りを含むウェブページを調査したところ、(1)「次ラベル」の素性と同じように、次ページリンクのリンクテキストもしくはそれを定義する a タグの属性に、「ページ」や「PAGE」などといったキーワードがよく出現することがわかった。これらは、「次」や「NEXT」と同様に、そのリンクが次ページへのリンクであることを示唆すると考えられる。そこで、(2)「ページラベル」の素性は、リンクを定義する a タグ

（引用元 URL https://webbibouroku.com/）

図 3.4: ページ送りを含むウェブページの例



（引用元 URL https://www.hawtcelebs.com/）

図 3.5: HTML タグの属性が next を含む次ページリンクの例

に「ページ」ならびに「PAGE」が含まれるか否かを表すものと定義する。この素性もキーワードを含む (このときの値は 1) か含まないか (このときの値は 0) を表すバイナリ素性である。また、「PAGE」の有無をチェックする際には大文字と小文字を区別しない。

(3)「1 文字ラベル」の素性は、リンクテキストが 1 文字であるか否かを表す素性である。図 3.5 のページ送り部の例では、次ページリンクのリンクテキストは「»」という 1 文字で表されている。この例のように、次ページリンクを表すテキストは、矢印や数字のように 1 文字で表されていることが多かった。言い換えれば、リンクテキストが 1 文字のとき、そのリンクは次ページリンクに該当する可能性が高い。

上記を踏まえ、(3)「1 文字ラベル」の素性は、リンクテキストが 1 文字であるか否かを表すものと定義する。この素性は、リンクテキストが 1 文字であるときは 1、そうでないときは 0 を値とするバイナリ素性である。

(4)「リンク出現回数」の素性は、リンク (URL) のウェブページ全体における出現回数である。次ページの URL は他のリンク先ページの URL と比べて出現回数が多い傾向が見られる。図 3.6 に示すページ送りの例では、ページ送りのリンク先として「2」「3」「>」の 3 つがある。これらのうち、次ページリンクに該当するのは、リンクテキストが「2」と「>」である URL(https://dime.jp/genre/585883/2/) である。一方、「3」のリンクはページ分割された 3 番目のページへのリンクであ

（引用元 URL https://dime.jp/genre/585883/）

図 3.6: 複数回出現する次ページリンクの例

るので、次ページリンクに該当しない。この例のように、次ページリンクは、数字の「2」と「>」のような記号や矢印の両方で表示されることが多い。すなわち、そのウェブページで複数回出現するリンクは次ページリンクである可能性が高い。上記を踏まえ、リンクの入力ウェブページにおける出現回数を素性とする。

(5)「テキスト長」と (6)「テキスト長の割合」の素性は、リンクテキストの長さを考慮した素性である。ページ送り部は一般的に長いリンクテキストは使われていないため、これらの素性を設計した。これまでの例に示したように、ページ送りの次ページへのリンクは、数字、矢印、記号など短いテキストで表されることが多い。一方、ページ送り以外の一般の別ページへのリンクは、リンクテキストとしてリンク先ページの簡単な説明が書かれるなど、リンクテキストが長くなる傾向がある。したがって、リンクテキストの長さはリンクが次ページリンクであるか否かの判定の手がかりになると考えられる。

上記を踏まえ、(5)「テキスト長」の素性は、リンクテキストの長さ (文字数) と定義する。一方、(6)「テキスト長の割合」の素性は式 (3.1) のように定義する。

$$\frac{\text{リンクテキストの長さ}}{\text{ウェブページ全体のテキストの長さ}} \tag{3.1}$$

(7)「リンク長」と (8)「リンク長の割合」は、リンクの URL の長さを考慮した素性である。一般に、1 つのウェブサイトは複数のウェブページから構成され、また複数のウェブページは階層構造を持っている。ウェブサイトの階層構造において、深い位置にあるウェブページは、その URL の長さ (文字列の数) が長くなる。言い換えれば、リンクの URL の長さはリンク先ウェブページの階層構造の深さを表す。ベージ送りされた各ウェブページの階層構造における深さの情報が次ページリンクの判定の手がかりになることを考慮し、これら 2 つの素性を導入する。

(7)「リンク長」の素性は、リンク先ウェブページの URL の文字列数と定義する。リンクが相対リンクの場合、入力ウェブページの URL のドメインを補完して、絶対リンクに直してから URL 文字列数を測る。一方、(8)「リンク長の割合」は式 (3.2) のように定義する。

（引用元 URL https://nancoco.net/nyuuin_report1/）

図 3.7: リンクが規則的なページ送り

$$\frac{\text{リンクの URL の長さ}}{\text{ウェブページにおける全てのリンクの URL の長さの和}} \tag{3.2}$$

これらの素性は、定義は異なるが、いずれもリンク先ウェブページの URL の長さを次ページリンクの判定に利用するために導入する。

3.2.2.2 LinkSimilarity 素性

表 3.1 に示した (9)LinkSimilarity 素性は、判定対象のリンクがその周辺にあるリンクとどれだけ類似しているかを表す素性である。以下、LinkSimilarity 素性の定義と、これを素性として導入した理由を説明する。

ページ送りを含むウェブページでは、ページ分割されたウェブページへのリンクが並べて表示されることが多い。図 3.5 の例では「2」「3」「4」「5」「⟩⟩」といったように、図 3.7 の例では「入院 2...」「入院 3...」といったように、分割されたウェブページへのリンクが並べて表示されている。以下、このようなページ送りされたウェブページへのリンクが集中して現われる箇所を「ページ送り部」と呼ぶ。本節で検出の対象としている次ページリンクはページ送り部の中に存在しているため、もしリンクがページ送り部の内部にあるかどうかを示す素性を使用することができれば、次ページリンク検出の性能を高めることが期待できる。

ページ送り部の特徴として、複数のリンクが密集して表示されていることが挙げられる。また、ページ送り部に出現するリンクは互いに類似性を持つことが多い。例えば、図 3.7 のウェブページでは、ページ送り部に出現するリンクのリンクテキストは、記事のタイトルが書かれていて規則性がないものの、リンク先の URL は全て「https://nancoco.net/nyuuin_report?(?は数字)」であり、規則性が認められる。

本研究における LinkSimilarity 素性は、ページ送り部が持つ規則性に着目し、判定対象のリンクとその周辺にあるリンクがページ送り部の規則性を有するかどうかを表す素性である。具体的には、判定対象のリンクとその周辺のリンクがどれだけ類似しているかを表す。この素性の値 (判定対象のリンクと周辺のリンクの類似度) が大きいほど、判定対象のリンクはページ送り部の内部にある可能性が高く、次ページリンクに該当する可能性も高くなる。

LinkSimilarity 素性は以下の手続きで抽出する。まず、処理の対象とするウェブページから a タグ (別ページへのリンク) に相当する DOM ノード ($atag_i$ と記す) を抽出し、これを HTML ファイルにおける出現の順序に並べたリストを「A タグリスト」と呼ぶ。以下、A タグリストを $\{atag_1, atag_2, \cdots, atag_m\}$ と表す。m はリストにおける a タグの総数である。ただし、3.2.1 項で述べたように、同一ドメインのウェブページ以外へのリンクを表す a タグは除外する。次に、個々の $atag_i$ に対する LinkSimilarity 素性の値を式 (3.3) のように定義する。

$$LinkSimilarity(atag_i) = \max_{i-2 \leq j \leq i+2 \wedge j \neq i \wedge j \geq 1 \wedge j \leq m} sim_{link}(atag_i, atag_j) \quad (3.3)$$

ここで、sim_{link} は 2 つの a タグ間の類似度を表す。すなわち、LinkSimilarity 素性の値は、A タグリストにおける $atag_i$ の前後 2 つに位置する別の a タグ ($atag_j$) のそれぞれについて類似度を計算し、その最大値と定義する。ただし、 $atag_i$ が A タグリストにおける 1 番目もしくは 2 番目、あるいは A タグリストの末尾もしくは末尾の 1 つ前に位置するとき、前後 2 つの範囲に位置する a タグの個数が 4 より小さくなることに注意していただきたい。

次に、$atag$ 間の類似度を計算する方法について述べる。本研究では、$atag$ のリンク先 URL 同士の類似度を Jaccard 係数を用いて測る手法を採用する。Jaccard 係数は 2 つの集合での類似度を測る指標で、式 (3.4) のように定義される。2 つの集合 A,B の和集合の要素数を分母に、積集合の要素数を分子とする。2 つの集合 A,B の要素が似ているほど Jaccard 係数は 1 に近い値をとる。

$$Jaccard(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (3.4)$$

$atag$ が表すリンクの URL を、それに含まれる文字の集合で表現し、2 つの URL の文字集合の Jaccard 係数を求め、これを $atag$ 間の類似度とする。すなわち、a タグ $atag_i$ が表すリンク先 URL に含まれる文字集合を UC_i とするとき、$atag$ 間の類似度 sim_{link} を式 (3.5) のように定義する。

$$sim_{link}(atag_i, atag_j) = Jaccard(UC_i, UC_j) \quad (3.5)$$

例えば、図 3.7 における最初の a タグのリンクの「https://nancoco.net/nyuuin_report2」と 2 番目のリンクの「https://nancoco.net/nyuuin_report3」について Jaccard 係数を計算すると両者の URL の文字集合の和集合は {h,t,p,s,:,/,n,a,c,o,e,y,u,i,_,r,2,3}, 積集合は {h,t,p,s,:,/,n,a,c,o,e,y,u,i,_,r,2,3} となる。それぞれの要素数は 19 と 17 であるため、Jaccard 係数は $17/19 = 0.89$ となる。これは比較的高い値であり、これらの a タグがページ送り部を構成していることを示唆する。なお、図 3.7 に出現している全ての a タグの URL 同士は 1 文字違いであり、これらのリンク間の類似度は全て同じ値となる。

3.2.2.3　複数のタグからの素性抽出

判定対象のリンクから素性を抽出する際には、同じリンクがひとつのウェブページにおける複数の a タグに出現する場合を考慮する必要がある。例えば、同じリンクがタグ A、タグ B に出現し、素性 (5)「テキスト長」の値がタグ A においては 4 でタグ B では 8 の場合、素性値として 4 を採用するか 8 を採用するかという問題がある。同じリンクが複数の a タグに出現し、それぞれの a タグから異なる素性値が抽出されたとき、リンクに対する最終的な素性の値を決める方法を表 3.2 に示す。

素性 (1) から (3) はバイナリ素性であり、いずれかのタグが条件を満たすときは 1、それ以外は 0 とする。素性 (4) は、リンクの出現回数を値とし、もともと個々のタグから素性を抽出するものではない。素性 (5)〜(8) については、それぞれのタグから取得した素性値の中央値とする。最後に、素性 (9) については、それぞれのタグから取得した素性値の最大値とする。

表 3.2: 同じリンクが複数の a タグに出現するときの素性値の決定方法

素性	採用値
(1) 次ラベル	いずれかのタグが条件を満たすとき 1、それ以外は 0
(2) ページラベル	いずれかのタグが条件を満たすとき 1、それ以外は 0
(3) 1 文字ラベル	いずれかのタグが条件を満たすとき 1、それ以外は 0
(4) リンク出現回数	—
(5) テキスト長の割合	中央値
(6) テキスト割合	中央値
(7) リンク長	中央値
(8) リンク長の割合	中央値
(9)LinkSimilarity	最大値

3.2.3　不均衡データへの対応

ページ送りを行うウェブサイトには通常多くのリンクが存在する。ページ送り以外のリンクとして、他ページへのリンク、ページ内で表示される各記事へのリンクなどがある。3.2.1 項で述べたように、ドメイン外のリンクを削除したり、複数の a タグで出現する同一リンクをマージする処理を行ったりしても、ひとつのウェブページから数百以上のリンクが抽出されることも多い。一方、次ページリンクは一つのウェブページに一つだけ存在する。

次ページリンク検出タスクでは、次ページリンクに該当するリンクが正例、該当しないリンクが負例となるが、訓練データにおける 1 つのウェブページにおいては、正例が一つに対し負例が数百個存在する。このため、訓練データは正例と負例の数に著しい偏りがある不均衡データとなる。一般に、不均衡データから分類器を学習すると、数の少ない正例の特徴を学習できず、常に負例と判定するような分類性能の低い分類器が学習される問題があることが知られている。そのため、分類器を学習する前に不均衡データの是正を行う。

本研究では、オーバーサンプリング手法である Synthetic Minority Over-sampling(SMOTE)[9] を用いる。一般に、オーバーサンプリングとは、正例のデータを人工的に増やして不均衡データを是正する手法である。SMOTE の概要は次の通りである。素性ベクトルのベクトル空間上において、1 つの正例に対し、その近傍にある他の正例を探索し、正例とその近傍データの直線上にある点をランダムに選択し、それを新たな正例として訓練データに加える。これを繰り返すことで正例の数を増やし、正例と負例の数の不均衡を是正する。本研究では、SMOTE により、正例数が負例数の 1/10 となるまで正例の数を増やす。SMOTE を用いるときは正例と負例の数が等しくなるまで正例を増やすこともあるが、実際のページ送りを含むウェブページでは、負例の数は正例の数より多いのは明らかであるため、是正後の訓練データの正例と負例の比を 1:10 と設定する。

3.2.4 機械学習アルゴリズム

本研究では次ページリンク検出モデルを学習するための機械学習アルゴリズムとして、決定木、ランダムフォレスト、Gradient Boosting Decision Tree(GBDT) の 3 つを用いる。

決定木は正例と負例の分類への貢献度の高い説明変数 (素性) を選別して、木構造の分類モデルを学習する機械学習アルゴリズムである。学習したモデルが説明変数をノードとする木構造として表現されるため、人間が容易に解釈できるという特徴がある。

ランダムフォレストは決定木にアンサンブル学習を組み合わせた手法である。複数の決定木を構築し、個々の決定木の予測結果の多数決をとることで最終的な分類結果を決定する。分類器として学習されるのは複数の決定木となる。1 つの決定木による分類よりも性能が良くなりやすいことが知られている。また、決定木と同様に、各々の説明変数の重要度が理解しやすい。

GBDT もランダムフォレストと同様に決定木にアンサンブル学習を組み合わせた手法であるが、構築する一連の分類器が反復的に学習される。ランダムフォレストの反復学習において、あるステップの分類器によって誤分類された訓練データに対して、これが正しく分類される可能性が高くなるように決定木間の重みが更新される。パラメータを適切に設定することによってランダムフォレストよりも良い分類性能が得られるが、誤分類された訓練データについてこれを正しく分

類するように重みを更新することから、過学習を起こしやすい恐れがあることも知られている。

3.3 主コンテンツの検出

本節では、ページ送りを含むウェブページから主コンテンツを検出する手法について述べる。

3.3.1 タグの抽出

ウェブページの構造は DOM ツリーで表現される。DOM ツリーとは、ウェブページの HTML ソースファイルにおける HTML のタグの入れ子構造を木構造で表現したものである。図 3.8 はページ送りを含むウェブサイトとその DOM ツリーの例である。図中の青い点は DOM ツリーのノードを表す。これは「DOM ノード」と呼ばれ、HTML ソースファイルにおける一つの HTML タグに対応する。

主コンテンツ検出タスクでは、DOM ツリーにおける個々の DOM ノードに対し、それが主コンテンツに該当するかを判定する。主コンテンツに該当する DOM ノードとは、それに対応する HTML タグが主コンテンツとなるテキストを囲んでいるノードと定義する。以下、DOM ノード、すなわち HTML タグを単に「タグ」と呼ぶ。主コンテンツ検出タスクは、タグが主コンテンツを含むか否かを判定するタスクと定義する。ひとつのウェブページに対し、主コンテンツに該当するタグが複数存在することがある。以下の例では、主コンテンツが div_1 と div_2 の 2 つのタグに分かれて配置されている。この場合、div_1 と div_2 をともに主コンテンツを含むタグとして検出の対象とする。

$\langle \mathrm{div}_1 \rangle$ (主コンテンツ 1) $\langle /\mathrm{div}_1 \rangle$
$\langle \mathrm{div}_2 \rangle$ (主コンテンツ 2) $\langle /\mathrm{div}_2 \rangle$

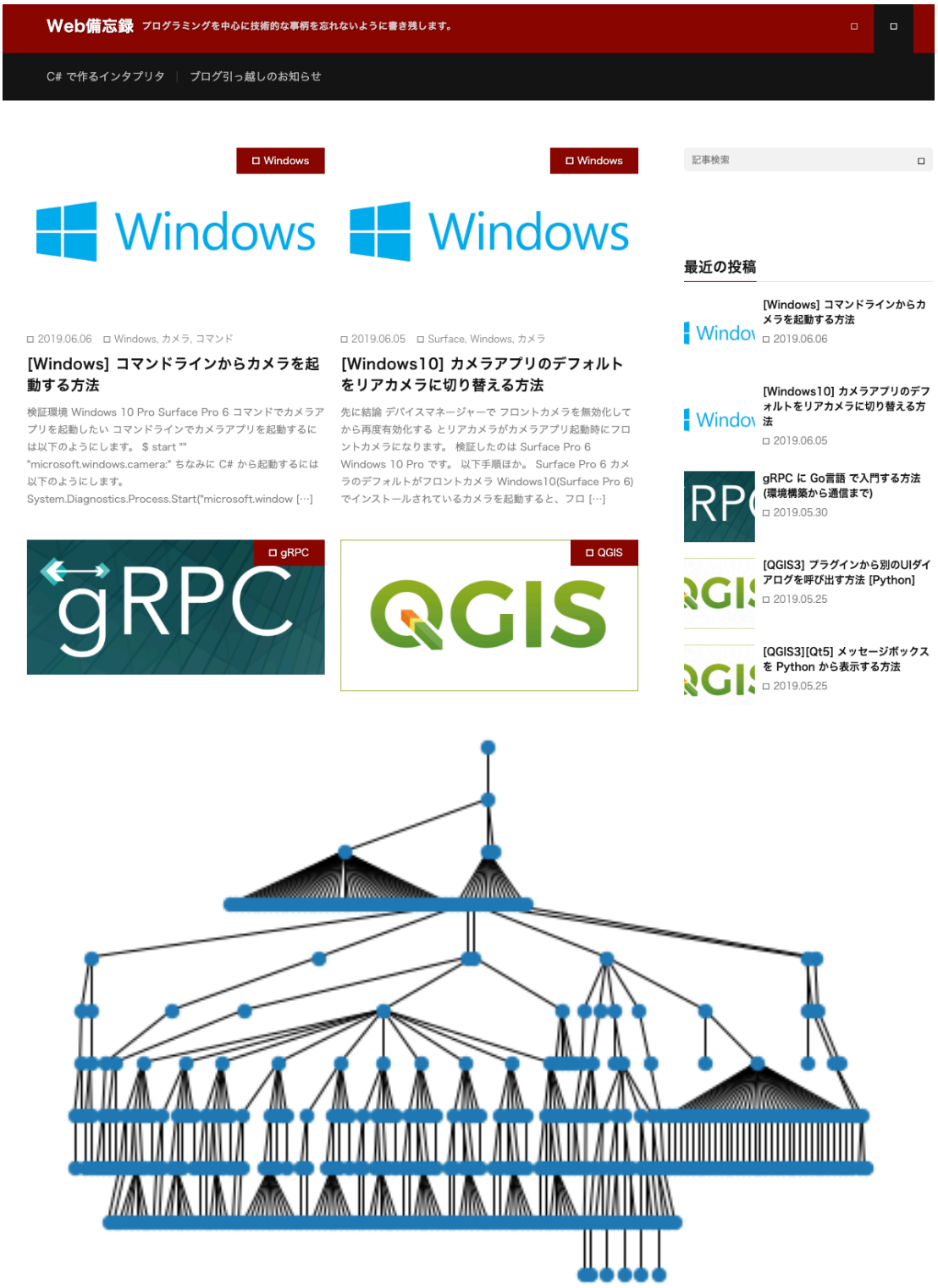
一方、以下のような入れ子構造により、同じ主コンテンツを囲む HTML タグが複数存在することがある。

$\langle \mathrm{div}_3 \rangle \langle \mathrm{div}_4 \rangle$ (主コンテンツ) $\langle /\mathrm{div}_4 \rangle \langle /\mathrm{div}_3 \rangle$

この場合は入れ子構造の一番外側、すなわち DOM ツリーにおける一番上位の HTML タグ (上の例では div_3) のみを検出するべき主コンテンツのタグとする[1]。

前処理として、入力となるウェブページの構造を解析し、その DOM ツリーを得る。そして、それに含まれる全てのタグを抽出する。次項以降で説明する後続の処理では、それぞれのタグから素性を抽出し、タグを素性ベクトルで表現した上で、そのタグが主コンテンツを含むか否かを判定する。

[1] 4 章で後述する評価実験に用いたデータセット Wedata では、一番外側以外のタグが主コンテンツとして定義されていることがある。



（引用元 URL https://webbibouroku.com/）

図 3.8: ウェブサイトとその DOM ツリーの例

3.3.2 素性

表 3.3: 主コンテンツ検出に利用する素性

素性	説明
(1) タグの長さ	タグの長さ
(2) タグの深さ	DOM ツリーにおけるタグの深さ
(3) タグの位置	HTML ファイルにおけるタグの位置
(4) タグの相対位置	HTML ファイルにおけるタグの相対的な位置
(5) ブロックレベル要素	W3C 定義のブロックレベル要素に該当するか
(6) 主コンテンツにならないタグ	HTML タグの種類が明らかに主コンテンツにならないものであるか
(7) 兄弟タグのテキスト長	兄弟タグ内にあるテキストの長さ
(8) 兄弟タグのテキスト長の割合	兄弟タグのテキスト長のウェブページ全体のテキスト長に占める割合
(9) 兄弟タグの句読点割合	兄弟タグ内にある句読点数のウェブページ全体の句読点数に占める割合
(10) 兄弟タグのテキスト密度	兄弟タグにおける内包するタグ数とテキストの比
(11) 兄弟タグ数	兄弟タグの数
(12) 子タグ数	子タグの数
(13) 子タグ数の割合	子タグ数のウェブページ全体のタグ数に占める割合
(14) 次ページリンクからの距離	次ページリンクのタグからの距離

タグから機械学習のための素性を抽出する。主コンテンツ検出モデルの学習に用いる素性の一覧を表 3.3 に示す。これらの素性は、タグ自体に関する素性 (表 3.3 における (1)〜(6))、周辺のタグ (兄弟タグ、子タグ) に関する素性 (表 3.3 における (7)〜(13))、次ページリンクを持つタグとの距離に関する素性 (表 3.3 における (14)) に分けられる。

本研究では、判定対象のタグ自体だけでなく、その周辺のタグからも素性を抽出する。ここでの周辺のタグとは兄弟タグと子タグを指す。DOM ツリーでは、通常の木構造と同様に、親子関係、兄弟関係にあるタグが存在する。子タグとは、あるタグによって直接内包されているタグであり、DOM ツリーでは直近の下位のタグに相当する。親タグとは、逆にそのタグを直接内包しているタグであり、DOM ツリーでは直近の上位のタグに相当する。兄弟タグは、親タグが同一である別の

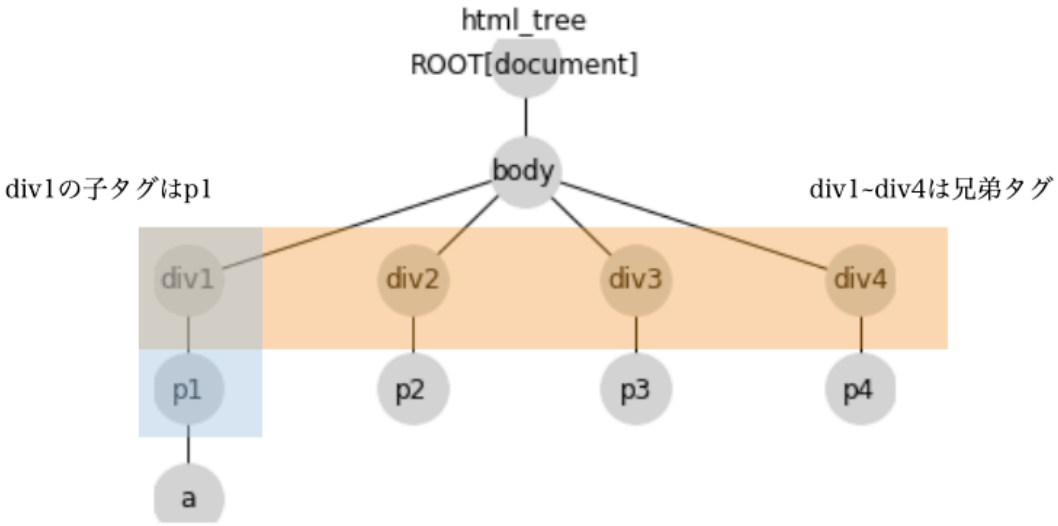


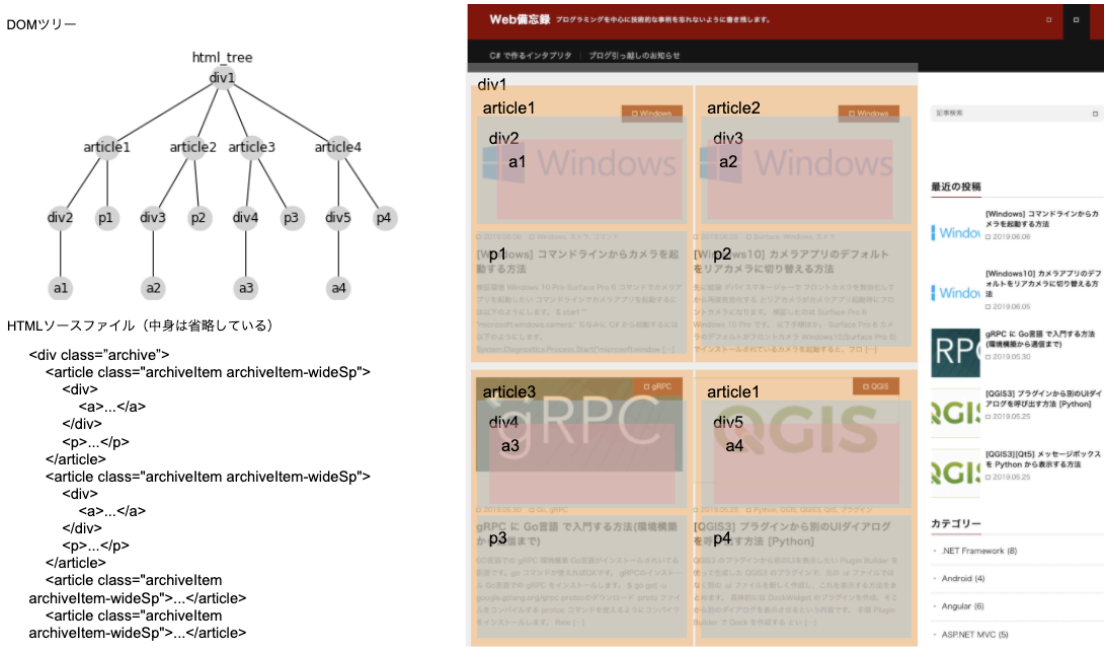
図 3.9: 兄弟タグ、子タグの例

タグである。図 3.9 に示す DOM ツリーでは、div1 タグの子タグは p1 である。また、div1〜div4 は同じ親タグ body を持つので、これらは互いに兄弟タグとなる。

主コンテンツ検出においては、タグ自体の素性に加えて、兄弟タグ、子タグの情報が手がかりになると考えられる。図 3.10 は、ページ送りを含むウェブページのスクリーンショット、その DOM ツリー、その HTML のソースファイル (抜粋) を示している。スクリーンショットと DOM ツリーでは、分かりやすくするため、タグに連番を振っている。このウェブページの場合、主コンテンツはオレンジ色で表示されている article タグである。これらのタグは、div,p タグなど、多くのタグを子タグとして内包している。主コンテンツであるタグは、情報を多く表示するために子タグを複数持つ傾向がある。したがって、子タグの数は主コンテンツ検出のための重要な手がかりとなりうる。また、article1 から article4 は全て主コンテンツを表すが、article タグ以下の構造は類似しており、それらが兄弟タグとして繰り返し出現している。このように、主コンテンツを表す一連のタグは兄弟タグとして繰り返し構造を持つ傾向がある。そのため、兄弟タグの情報もまた主コンテンツ検出の重要な手がかりとなる。

タグ自体の素性、周辺のタグの素性に加え、本研究では、次ページリンクからの距離を素性として利用する。ページ送りを含むウェブページでは、主コンテンツと次ページリンクは近くに配置されることが多いためである。

以降では、それぞれの素性の詳細を説明する。



（引用元 URL https://webbibouroku.com/）

図 3.10: 主コンテンツの例

3.3.2.1　タグ自体の素性

表 3.3 における素性 (1)-(6) は、判定対象のタグ自体に関する素性である。

(1)「タグの長さ」の素性は、タグがどれだけ多くのタグやテキストを含むかを表す素性である。HTML ソースファイルにおけるタグの長さ (文字数) を素性の値とする。具体的には、タグに対応するオープンタグ (⟨div⟩ など) とクローズタグ (⟨/div⟩ など) で囲まれた文字列の長さである。この文字列には、判定対象のタグが内包するタグも全て含まれる。主コンテンツは通常多くのテキスト、子タグ、レイアウト情報を表す HTML の属性などを含み、それを含むタグ自体が長くなると考えられるため、素性として導入する。

(2)「タグの深さ」の素性は、判定対象のタグが DOM ツリーにおいてどれだけ深い場所に位置するかを表す素性である。すなわち DOM ツリーのルートタグからの距離を素性の値とする。多くの主コンテンツは DOM ツリーにおいて浅い場所ではなく中央から深部に位置するため、この素性を導入する。

(3)「タグの位置」と (4)「タグの相対位置」の素性は、HTML ファイルにおけるタグの位置に関する素性である。解析対象のウェブページに含まれるタグを HTML ファイルに出現する順序に並べたとき、素性 (3) は、そのリストのインデックス (判定対象のタグがリストの何番目に位置するか) を素性の値とする。一方、素性 (4) は、素性 (3) のインデックスの相対値、すなわちインデックスをウェブページに含まれるタグの総数で割った値を素性値とする。主コンテンツはウェブページの先頭や末尾ではなく中央に位置することが多いため、これらの素性を導入する。

(5)「ブロックレベル要素」の素性は、判定対象のタグがブロックレベル要素であるか否かを表す素性である。WorldWide Web Consortium(W3C)[8] は、HTML を構成する要素をブロックレベル要素とインライン要素に大別している。インライン要素は、a タグや span タグのように、テキストの一部をマークアップする要素である。一方、ブロックレベル要素は、p タグや div タグのように、テキストのまとまりをマークアップする要素であり、ウェブブラウザ上では基本的に改行されて表示される。また、ブロックレベル要素は他のインライン要素を内包できる。一般に、インライン要素よりもブロックレベル要素の方がより多くのコンテンツを含むため、主コンテンツはブロックレベル要素であることが多いと考えられる。そのため、判定対象のタグがブロックレベル要素であるか否かを表す素性を導入した。この素性はバイナリ素性であり、ブロックレベル要素であるときは素性の値は 1、そうでないときは 0 とする。

(6)「主コンテンツにならないタグ」の素性は、判定対象のタグが主コンテンツになる可能性が限りなく低いことを表す素性である。HTML タグの中には、主コンテンツになりえないタグ、もしくは主コンテンツになる可能性が低いタグが存在する。表 3.4 はそのようなタグの一覧である。「ページ全体を含むタグ」は、[document](DOM ツリーのルート) と html であり、ページ全体を包含するために主コンテンツのタグではない。「テキストを含まないタグ」は、メタデータを表す meta や、Java Script のソースコードを記述する script など、ウェブブラウザで表示されるテキスト以外の情報をマークアップするタグである。「テキスト装飾に使われるタグ」は、具体的には span タグだが、テキストの一部に対してフォントの大きさや色を指定するために使わることが多く、主コンテンツ全体を包含することは稀である。(6) の素性は、判定対象のタグの種類が表 3.4 に示したタグのいずれかに該当するかを表すバイナリ素性である。該当するときに素性の値は 1、該当しないときは 0 となる。

表 3.4: 主コンテンツにならないタグ

ページ全体を含むタグ	[document], html
テキストを含まないタグ	noscript, meta, head, input, script, style
テキスト装飾に使われるタグ	span

3.3.2.2　周辺のタグに関する素性

表 3.3 における素性 (7)-(13) は、判定対象のタグの周辺に位置する兄弟タグ、子タグに関する素性である。

(7)「兄弟タグのテキスト長」と (8)「兄弟タグのテキスト長の割合」の素性は、兄弟タグが包含するテキストの長さを表す素性である。主コンテンツがいくつかのタグに分かれて掲載されているとき、これらは兄弟タグとして繰り返し構造を

持つことが多い。一方で、タグが多くのテキストを含めば、それが主コンテンツである可能性が高い。これら 2 つの素性は、兄弟タグのテキストの量を手がかりとし、兄弟タグが主コンテンツになりやすく、したがって判定対象のタグもまた主コンテンツになりやすいという傾向を学習するために導入する。素性 (7) は、兄弟タグが包含するテキストの長さ (文字数) を素性の値とする。一方素性 (8) は、素性 (7) のテキスト長のウェブページ全体のテキスト長に対する割合を素性の値とする。なお、タグ自身が含むテキストの大きさは、(1)「タグの長さ」の素性に反映されていることに注意していただきたい。

(9)「句読点の割合」の素性は、兄弟ノードがどれだけ句読点を含むかを表す素性である。ウェブページにおけるテキストのうち、主コンテンツに該当するテキストと、広告やリンクなどの主コンテンツ以外のテキストとでは、句読点の使われ方が異なる。広告やリンクでは句読点はあまり使われないのに対し、主コンテンツでは句読点が多用される傾向がある。そのため、兄弟タグの句読点の多さは主コンテンツ検出の重要な手がかりになると考えられる。上記を踏まえ、この素性を式 (3.6) のように定義する。SIB は判定対象のタグの兄弟タグの集合を表す。

$$\frac{\sum_{tag \in SIB} tag \text{ が内包するテキストに出現する句読点の数}}{\text{ウェブページ全体のテキストに出現する句読点の数}} \tag{3.6}$$

(10)「兄弟タグのテキスト密度」の素性は、テキスト密度によって規定される素性である。テキスト密度 (Text Density) は、Song[5] らの研究において、主コンテンツ抽出の手がかりとして利用された素性である。テキスト密度の定義を式 (3.7) に示す。

$$TD_i = \frac{C_i}{T_i} \tag{3.7}$$

TD_i はタグ i のテキスト密度を表す。T_i はタグ i が包含するタグの数、C_i はタグ i が包含するテキストの長さを表す。すなわち、TD_i は、タグ i が包含するタグについて、そのタグに含まれるテキストの長さの平均値である。

(11)「兄弟タグの数」の素性は、判定対象のタグの兄弟タグの多さを表す素性であり、兄弟タグの個数を素性の値とする。主コンテンツが同一構造で繰り返し出現しているときは、兄弟タグの数も多くなると考えられるため、主コンテンツ検出に有効と考え導入した。

(12)「子タグ数」と (13)「子タグ数の割合」の素性は、子タグの多さを表す素性である。主コンテンツのタグはブロックレベル要素であることが多く、また別のタグを子タグとして持つことが多いことから、これらの素性を導入した。素性 (12) は、判定対象のタグが持つ子タグの個数を素性の値とする。一方、素性 (13) は、素性 (12) の個数のウェブページ全体におけるタグの総数に対する割合を素性の値とする。

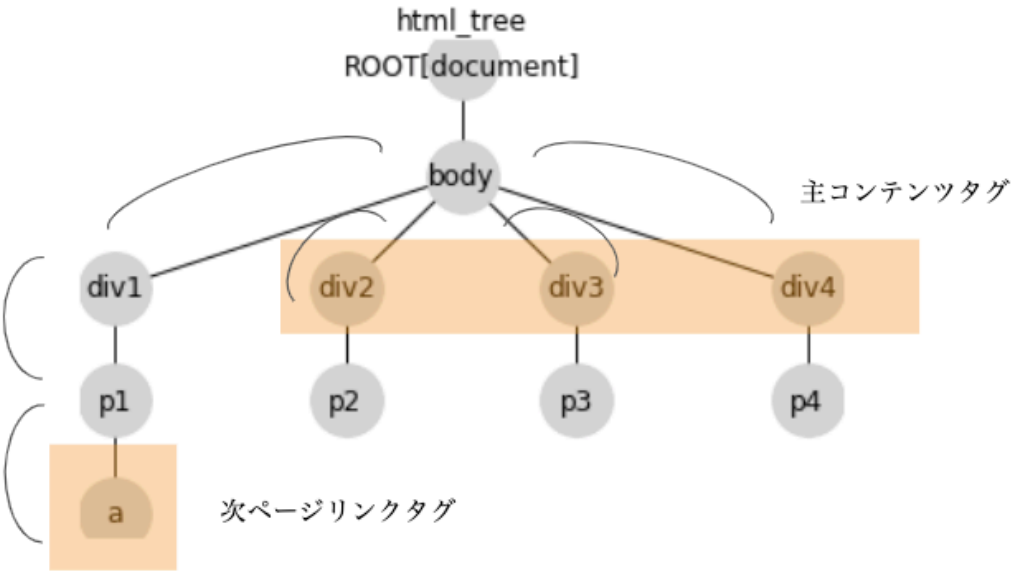


図 3.11: 主コンテンツタグと次ページリンクタグの距離の計算例

3.3.2.3 次ページリンクからの距離

表 3.3 における (14)「次ページリンクからの距離」の素性は、次ページリンクタグが判定対象のタグとどれほど近い位置にあるかを表す素性である。ページ送りを含むウェブページでは、ページ送りを行う次ページへのリンクを持つタグと、ページ送りの対象である主コンテンツに該当するタグは近くに出現することが多い。そのため、主コンテンツタグと次リンクページタグの距離は主コンテンツを検出するための素性として有効であると考えられる。

タグ間の距離は、全てのタグを HTML ソースファイルにおける出現順序に並べたリストにおけるインデックスの差によって測ることもできるが、本研究では DOM ツリーの木構造上の距離を測る。すなわち、DOM ツリーにおいて、あるタグ (ノード) から別のタグ (ノード) に到達する最短パスの長さを距離と定義する。

図 3.11 に示す DOM ツリーにおいて、主コンテンツに該当するタグは div2、div3、div4、次ページリンクであるに該当するタグは a であるとする。div2 タグと a タグの最短パスは div2 → body → div1 → p1 → a であり、そのエッジ数が 4 であるため、2 つのタグ間の距離も 4 となる。同様に、div3 タグと a タグ、div4 タグと a タグの距離も 4 となる

ウェブページにおける全てのタグを HTML ソースファイルにおける出現順序で並べ、そのリスト上で 2 つのノードの距離を測るとき、div2、div3、div4 タグと a タグの距離は異なり、div4、div3、div2 の順に距離が長くなる。しかし、div2、div3、div4 は兄弟タグで同じ性質を持つので、a タグとの距離が変わるのは望ましくない。DOM ツリー上で距離を測ると、div2、div3、div4 タグと a タグの距離は

全て同じになる。これは主コンテンツ検出の素性として使用する際には望ましい性質である。

木構造における2つのノードの距離を計算する方法について述べる。2つのノードvとwの距離はルートからvまでの距離とルートからwまでの距離の和から、vとwの共通の先祖のうち最も深いノード（Lowest Common Ancestor) とルートとの距離の2倍を引くことで計算できる。これは式 (3.8) のように表される。$dis(x, y)$は2つのタグxとyの距離、$root$はルートのタグ、LCAは2つのタグに最も近い共通の先祖のタグを表す。

$$dis(v, w) = dis(root, v) + dis(root, w) - 2 \times dis(root, LCA(v, w)) \tag{3.8}$$

「次ページリンクからの距離」の素性を抽出する際には、次ページリンクタグを特定する必要がある。分類器を学習するときには、訓練データに次ページリンクの情報がタグ付けされているため、これをそのまま用いる。一方、学習した分類器を未知のデータに適用する際には、3.2節で述べた手法で次ページリンクを自動推定し、その推定結果を利用する。いずれの場合も、情報として与えられるのは次ページへのリンク (URL) であり、次ページリンクを表わすタグそのものではない。そこで、次ページリンクのURLを持つaタグを探索し、これを次ページリンクのタグとする。該当するaタグが複数出現するときは、それらのうちHTMLソースファイル上で最初に出現したタグを次ページリンクのタグとする。このように特定された次ページリンクタグと判定対象のタグとの距離を測り、素性の値とする。

3.3.3 不均衡データへの対応

主コンテンツ検出タスクにおいても、次ページリンク検出タスクと同様に、分類器の訓練データは不均衡データである。タグが主コンテンツであるか否かを判定する分類器の訓練データでは、ウェブページに出現するタグのうち、主コンテンツに該当するタグが正例、それ以外のタグが負例となる。両者を比較すると、正例は負例よりもはるかに数が少ない。次ページリンク検出タスクでは、1つのウェブページにおける正例の数は1個であるが、主コンテンツ検出タスクでは、主コンテンツは複数のタグで表現されることがあるので、正例は必ずしも1つではない。それでも、実際のウェブページでは圧倒的に負例の数が多い。

本研究では、主コンテンツ検出については、オーバーサンプリングとアンダーサンプリングを組み合わせた手法で不均衡データを是正する。既に述べたように、オーバーサンプリングは正例の数を増やす手法である。一方、アンダーサンプリングは負例の数を減らすことで正例数と負例数の不均衡を是正する手法である。まず、3.2.3項と同様に、オーバーサンプリング手法であるSMOTE[9]を用いる。ここでは正例の数が負例の数の1/25になるまで正例を増やす。次に、訓練データの

中から負例をランダムに削除するアンダーサンプリング手法を用いる。最終的に、正例と負例を同じ数だけ含む訓練データを作成する。

3.3.4 機械学習アルゴリズム

主コンテンツ検出の分類器を学習するための機械学習アルゴリズムとして、3.2.4 項で述べた次ページリンク検出のための分類器を学習する際に用いた機械学習アルゴリズムと同じものを用いる。すなわち、決定木、ランダムフォレスト、GBDT の 3 つを用いて分類器を学習する。

第4章　評価

本章では提案手法の評価実験について述べる。具体的には、次ページリンク検出モデルと主コンテンツ検出モデルの性能を評価する。4.1 節では実験データについて述べる。4.2 節では評価基準ならびに実験で比較する手法を説明する。4.3 節では次ページリンク検出モデルの実験結果を、4.4 節では主コンテンツ検出モデルの実験結果を報告する。最後に、4.5 節では実験結果全体を考察する。

4.1　実験データ

実験データとして Wedata[2] を用いる。2.1 節で述べたように、Wedata は、ページ送りを使用しているウェブサイトに対し、分割された記事を連結するための情報を記載したデータベースである。1 つのウェブサイトにつき、ウェブサイトの URL の正規表現、次ページリンクを示す XPath 式、主コンテンツを示す XPath 式、対象ウェブサイトにおいてページ送りされているウェブページの URL の例、などの情報を持つ。今回の実験では、データベースに登録されているウェブサイトのうち、例として登録されているウェブページに、次ページリンクと主コンテンツが存在しないウェブサイトを除いて無作為に抽出した 129 ウェブサイトを利用した。これらのうち、8 割の 103 ウェブサイトを訓練データ、残りの 2 割の 26 ウェブサイトをテストデータとして用いた。例として登録されているウェブページの HTML ソースファイルを取得し、リンクの抽出ならびに DOM ツリーの構築を行った。Wedata の XPath 式に基づいて、次ページリンクに該当するリンク、主コンテンツに該当する DOM ノードを正解ラベルとして付与した。1 つのウェブページにつき、次ページリンクは 1 つ、主コンテンツに該当する DOM ノードは複数が正解ラベルとして設定される。

表 4.1 は実験データにおけるリンクの総数と次ページリンクの数を示している。訓練データ、テストデータ合計のリンク数が 19548 であるのに対し、次ページリンクの数は 129 で、全体のおよそ 150 分の 1 に過ぎず不均衡データとなっている。3.2.3 項で述べたように、提案手法では不均衡データを是正してから次ページリンク検出モデルを学習する。一方、表 4.2 は DOM ノードの数と主コンテンツに相当する DOM ノードの数を示している。訓練データ、テストデータ合計の DOM ノード数は 123590 であるのに対し、主コンテンツに相当する DOM ノードの数は 2474 であり、全体のおよそ 50 分の 1 に過ぎずこちらも不均衡データである。これらの

不均衡データは 3.3.3 項で述べたように、提案手法では不均衡データを是正してから主コンテンツ検出モデルを学習する。

表 4.1: 実験データにおけるリンクの数

実験データの種類	リンクの数	次ページリンクの数
訓練データ	14854	103
テストデータ	4694	26
合計	19548	129

表 4.2: 実験データにおける主コンテンツの数

実験データの種類	DOM ノードの数	主コンテンツに相当する DOM ノードの数
訓練データ	101054	1952
テストデータ	22536	522
合計	123590	2474

4.2 実験条件

4.2.1 評価基準

次ページリンク検出モデル、主コンテンツ検出モデルの評価基準について述べる。次ページリンクも主コンテンツ検出も、多くのデータ (ウェブページに出現する全てのリンクならびにタグ) から特定のデータ (次ページリンク、主コンテンツのタグ) を検索するタスクであることから、評価基準として精度、再現率、F 値を用いた。

次ページリンク検出タスク、主コンテンツ検出タスクの実験結果は表 4.1 の混同行列で表すことができる。ここで、次ページリンク検出タスクでは、Positive(正例) は次ページリンクに該当するリンク、Negative(負例) は該当しないリンクを表す。主コンテンツ検出タスクでは、Positive(正例) は主コンテンツに該当するタグ、Negative(負例) は該当しないタグを表す。この混同行列から、精度、再現率、F 値はそれぞれ式 (4.1)、(4.2)、(4.3) のように定義される。

$$\text{精度} = \frac{\text{真陽性}}{\text{真陽性} + \text{偽陽性}} \tag{4.1}$$

$$\text{再現率} = \frac{\text{真陽性}}{\text{真陽性} + \text{偽陰性}} \tag{4.2}$$

表 4.3: 混同行列

		予測されたクラス	
		Positive	Negative
実際のクラス	Positive	真陽性 ($TruePositive$)	偽陰性 ($FalseNegative$)
	Negative	偽陽性 ($FalsePositive$)	真陰性 ($TrueNegative$)

$$\text{F 値} \quad = \quad \frac{2 \times \text{精度} \times \text{再現率}}{\text{精度} + \text{再現率}} \tag{4.3}$$

なお、正解率 (予測されたラベルと実際のラベルが一致した割合) もよく利用される評価基準であるが、正例に対して負例が著しく多い不均衡データでは実際のクラスと予測されたクラスがともに Negative であるケースが多く、正解率が不自然に高く見積られ、評価基準として適さないため、今回の実験では利用しない。

4.2.2 比較する手法

次ページリンク検出タスクについては、以下の手法を比較する。

- ベースライン

  簡単なルールによって次ページリンクを検出する手法をベースラインとする。具体的には、以下のいずれかに該当するリンクを次ページリンクと判定する。

  - リンクテキストに「NEXT」または「次」という文字が入っている
  - リンクテキストが「2」である

  今回の実験データにおいては、ほぼ全てのウェブページがページ送りされた複数のウェブページの 1 ページ目であり、次のページは 2 ページ目に該当するため、リンクテキストが 2 であるという条件を設定している。

- 提案手法

  3.2 節で提案した手法。決定木、ランダムフォレスト、GBDT で学習した 3 つのモデルを比較する。

主コンテンツ検出タスクについては、以下の手法を比較する。

- ベースライン

  Kato らがウェブページから著者名を自動検出する研究の中で用いた主コンテンツ検出手法 [10] を基にする。Kato らの手法は、テキストの割合が DOM ツ

リー全体の半分を超えるDOMノードのうち、最も小さいテキスト割合を持つDOMノードを主コンテンツとみなす。この手法では主コンテンツとして検出されるDOMノードはひとつである。一方、本研究のデータセットでは正解ラベルのDOMノードは1つのウェブページ内に複数あり、1つのDOMノードだけを主コンテンツとして判定すると再現率が低くなることが予想される。そのため、Katoらの手法で検出したDOMノードの配下にある全てのDOMノードを主コンテンツとみなす手法をベースラインとする。これは、主コンテンツがウェブサイト内のテキストの過半数を占めると仮定した考えに基づいている。

- 提案手法 (正解の次ページリンクを使用)

  3.3節で提案した手法。決定木、ランダムフォレスト、GBDT で学習した3つのモデルを比較する。なお、「次ページリンクからの距離」の素性を抽出する際には、次ページリンクの情報が必要だが、ここでは実験データに付与された正解の次ページリンクの情報を用いる。

- 提案手法 (次ページリンクからの距離素性なし)

  提案手法で用いた素性のうち、「次ページリンクからの距離」の素性を使用しないモデル。この素性の有効性を検証するために比較する。また、決定木、ランダムフォレスト、GBDT で学習した3 つのモデルを比較する。

- 提案手法

  3.3節で提案した手法。「次ページリンクからの距離」の素性を抽出する際、訓練データでは正解の次ページリンクの情報を用いるが、テストデータでは提案手法によって自動推定された次ページリンクの情報を用いる。

4.3 次ページリンク検出手法の評価

ベースラインならびに決定木、ランダムフォレスト、GBDT により学習された分類器の精度、再現率、F値を表4.4に示す。また、提案手法のPR曲線を図4.1に示す。横軸は再現率、縦軸は精度を表す。3種類の機械学習アルゴリズムを用いた提案手法はいずれも、ベースラインと比べて、再現率は低いが精度は高く、F値も上回った。特にランダムフォレストはベースラインと大きな差が見られる。GBDTは決定木と比べてF値は同等だが、精度が高く再現率が低かった。F値が一番高いのはランダムフォレストであり、そのF値は0.750であった。

次に、機械学習に用いた個々の素性の有効性を評価する。ランダムフォレストの学習過程では素性の重要度が獲得できる。素性の重要度の計算方法を以下に示す。ランダムフォレストにおいては学習過程で決定木を生成するが、本研究では決定木学習におけるノードの分割にはジニ不純度を用いる。ジニ不純度は、集合

表 4.4: 次ページリンク検出モデルの実験結果

モデル	精度	再現率	F 値
ベースライン	0.474	0.844	0.607
決定木	0.679	0.731	0.704
ランダムフォレスト	0.818	0.692	0.750
GBDT	0.720	0.692	0.706

の中に存在するサンプルについて分類対象のクラス数が混合している程度を表す指標である。

ノード i におけるジニ不純度の定義を式 (4.4) に示す。ノード i のジニ不純度を G_i、分類クラス数を C、分類クラス $c \in \{1, \ldots, C\}$ に割り当てられている素性の出現割合を f_c と定義する。

$$G_i = \sum_{c=1}^{C} f_c(1 - f_c) \tag{4.4}$$

決定木の各ノードにおける分割前のジニ不純度から分割後のジニ不純度の減少量は、分割の有効性を示す指標となる。この指標を利得といい、利得が最も大きくなるようノードを分割することを繰り返して決定木を学習する。ノード i を 2 つのノード j,k に分割した場合の利得 I_i の定義を式 (4.5) に示す。ノード i,j,k 内のサンプル数を W_i,W_j,W_k として、ジニ不純度をサンプル数で重みづけしている。

$$I_i = W_i G_i - W_j G_j - W_k G_k \tag{4.5}$$

各素性毎に利得を決定木内で集計し、決定木内の全ての利得の合計値で割ることで、素性の利得を 0 から 1 の値に正規化する。正規化した利得が決定木における素性の重要度となる。ランダムフォレストが生成するそれぞれの決定木における素性の重要度を計算し、その平均値がランダムフォレストにおける素性の重要度になる。本実験では、このように計算されたランダムフォレストにおける素性の重要度を用いて素性の重要度を評価する。

図 4.2 は学習されたランダムフォレストにおける素性の重要度を示している。「次ラベル」の素性の重要度が 0.405 と最も高い。この素性は「次」「NEXT」といったキーワードがリンクテキストに出現するかを表すものである。「次」「NEXT」「2」といったキーワードだけで次ページリンクを検出するベースライン手法も、その F 値は 0.607 と比較的高い。次に重要度が高いのは、リンクテキストが 1 文字であるかを示す「1 文字ラベル」の素性 (0.210) である。リンクテキストが 1 文字のとき、数字や矢印などが多く、これらが次ページへのリンクであることを示唆するキーワードになっていると考えられる。その次に重要度が高い素性は、「ページ」「PAGE」といったキーワードの有無を示す「ページラベル」の素性 (0.177) であ

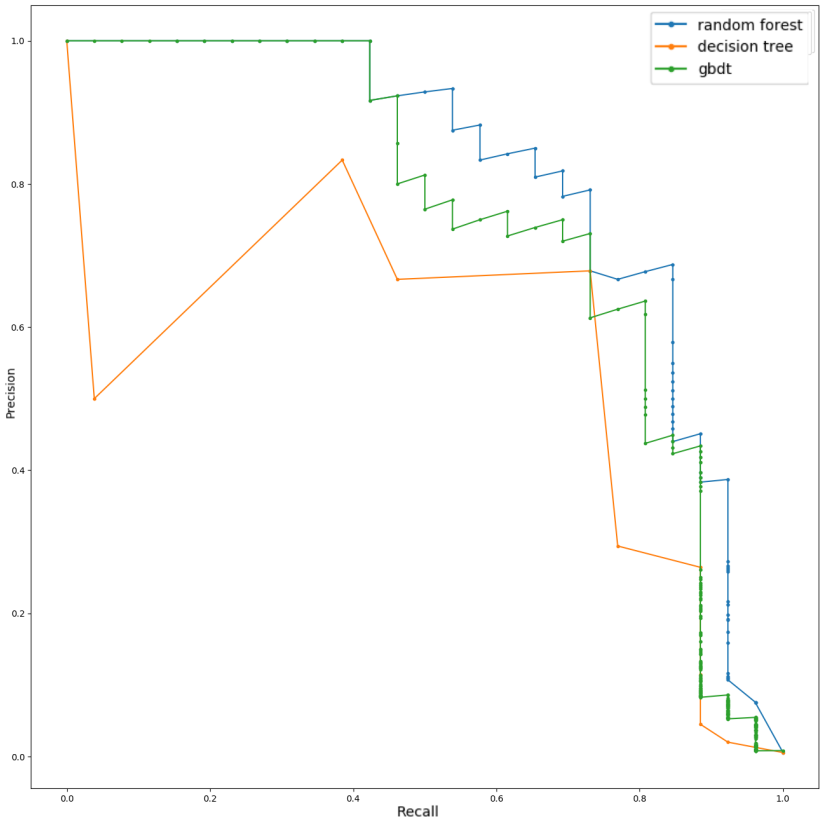


図 4.1: 提案手法による次ページリンク検出の PR 曲線

る。これら 3 つの素性の重要度が高いことから、次ページへのリンクを示唆するキーワードが次ページリンクの検出に有効であることがわかる。

これらの素性に次いで重要度が高いのは「リンク出現回数」の素性である。次ページリンクはページ送り部に複数回出現していることが多いことから、この素性が有効に働いたと考えられる。一方で、「テキスト長」、「テキスト長の割合」、「リンク長」、「リンク長の割合」の素性の重要度は相対的に低く、次ページリンクの特徴を顕著に表す素性ではないと考えられる。

次に、3.2.2.2 で提案した LinkSimilarity 素性の有効性を精査する。図 4.2 によると LinkSimilarity 素性の重要度は低い。しかし、この素性はページ送り部において参照先 URL が類似したリンクが密集していることを表すものであり、ページ送りの特徴を顕著に表すものと考えられる。そのため、LinkSimilarity 素性自体の重要性は低いが、他の素性と組み合わせることで次ページリンク検出の性能を大きく向上させる効果を持っている可能性がある。これを検証するため、LinkSimilarity 素性を使用する分類器と使用しない分類器を学習し、両者を比較した。機械学習アルゴリズムとして、表 4.4 で F 値が一番高かったランダムフォレストを用いた。結果を表 4.5 に示す。LinkSimilarity 素性を用いることで、精度、再現率、F 値が向上することがわかった。F 値の差は 0.027 ポイントとそれほど大きくはないが、

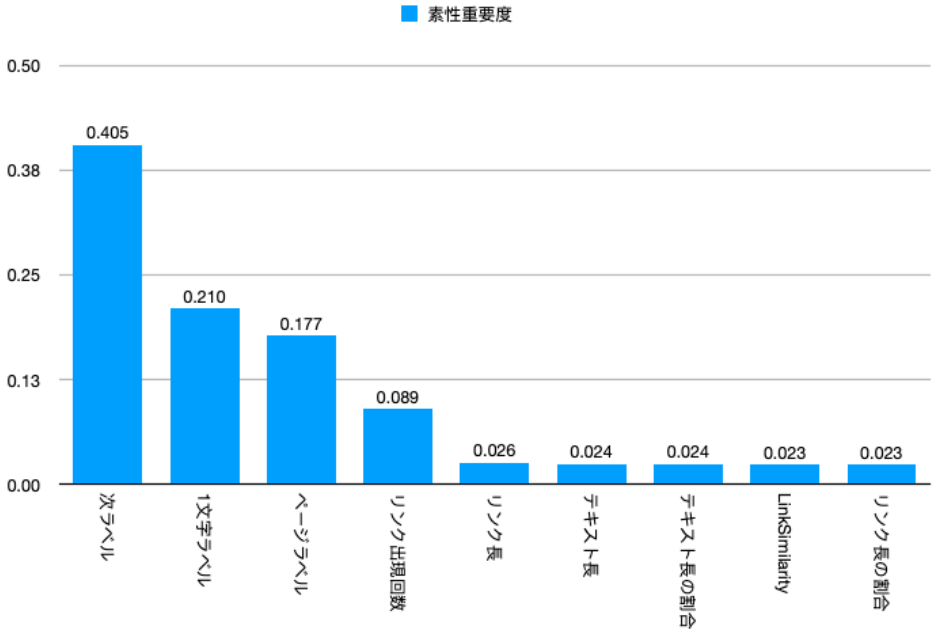


図 4.2: 次リンク検出モデルにおける素性の重要度

LinkSimilarity 素性が次ページリンク検出に有効であることが確認された。

表 4.5: LinkSimilarity 素性の有効性の評価（ランダムフォレスト）

LinkSimilarity 素性	**精度**	**再現率**	**F 値**
有り	0.818	0.692	0.750
無し	0.810	0.654	0.723

4.4 主コンテンツ検出手法の評価

まず、提案手法とベースラインによる主コンテンツ検出の性能を評価する。提案手法で「次ページリンクからの距離」の素性を抽出する際には、テストデータにおいても、正解の次ページリンクの情報を用いる。これは、提案手法の 3 つの機械学習アルゴリズムを比較する際、次ページリンクの検出の誤りによる影響を排除するためである。

表 4.6 は、決定木、ランダムフォレスト、GBDT で学習された分類器とベースラインによる精度、再現率、F 値を示している。また、図 4.3 は 3 つの機械学習アルゴリズムで学習された提案手法の PR 曲線である。提案手法はいずれもベースラインを大きく上回る。しかし、ベースラインの F 値は 0.003 と低く、主コンテンツを検出する手法としては単純すぎる。より適切なベースラインとの比較は今後の

課題である。3 つの機械学習アルゴリズムを比較すると、精度はランダムフォレストが最も高く、再現率は決定木が最も高かった。GBDT の精度と再現率はともに 3 つのアルゴリズムの中では 2 番目に結果が良く、また精度よりも再現率が高い。F 値が最も高かったのはランダムフォレストで、その値は 0.581 であった。実用的な観点から見ると十分に高いとは言えないが、一般に主コンテンツ検出は難しいタスクであり、0.581 という F 値はある程度の成果が得られていると考えられる。

表 4.6: 主コンテンツ検出モデル実験結果

モデル	精度	再現率	F 値
ベースライン	0.032	0.002	0.003
決定木	0.300	0.954	0.451
ランダムフォレスト	0.578	0.584	0.581
GBDT	0.415	0.711	0.524

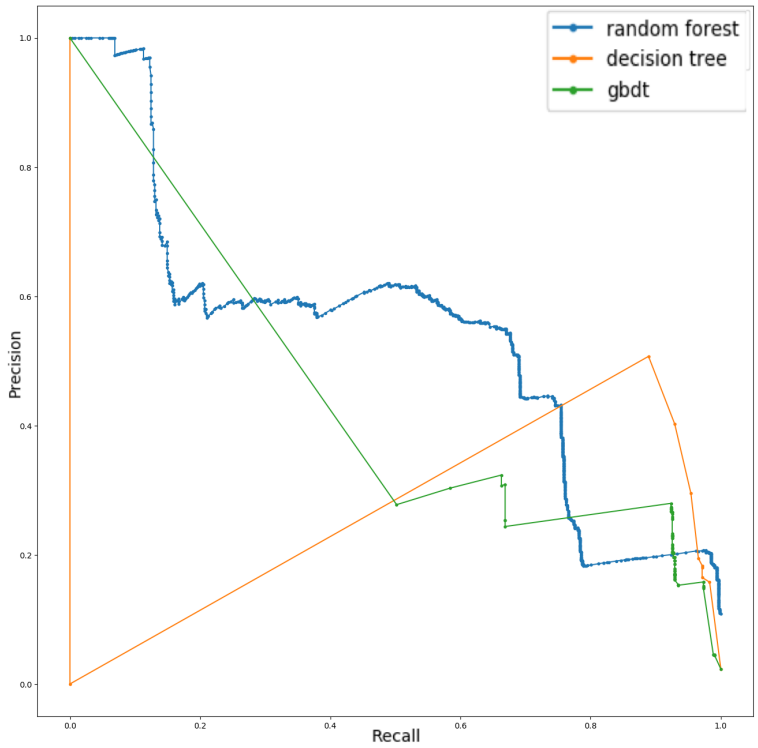


図 4.3: 提案手法による主コンテンツ検出の PR 曲線

次に、個々の素性の有効性を評価する。図 4.4 にランダムフォレストモデルの素性の重要度を示す。素性の重要度は全て 0.200 未満であり、次ページリンク検出モデルのように突出して重要度が高い素性は存在しない。重要度の高い素性としては、「兄弟タグのテキスト長」(0.171)、「兄弟タグのテキスト長の割合」(0.166)、

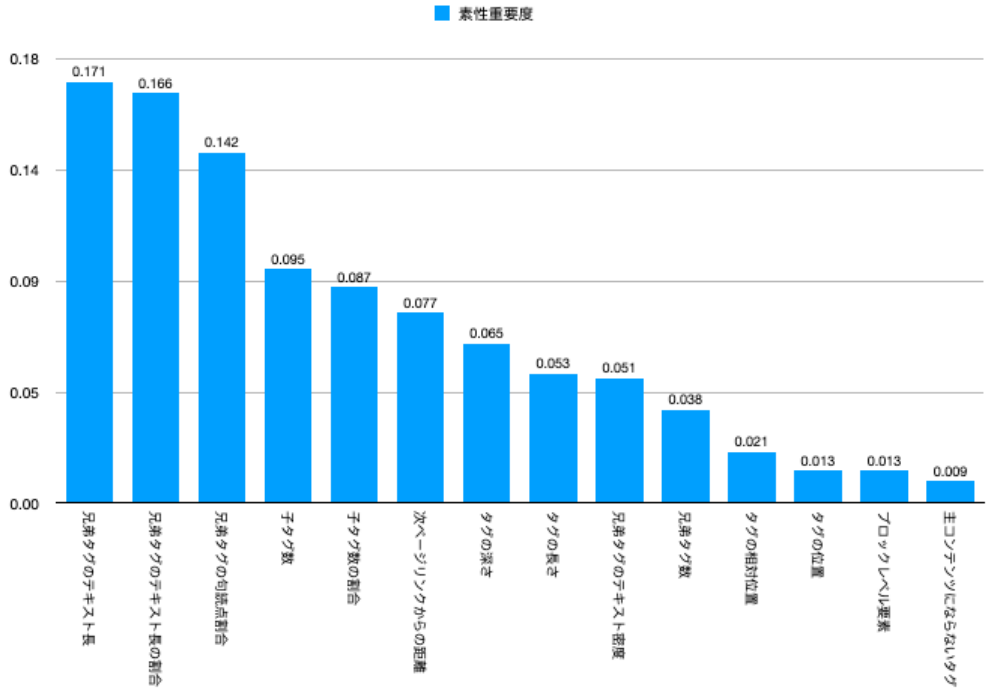
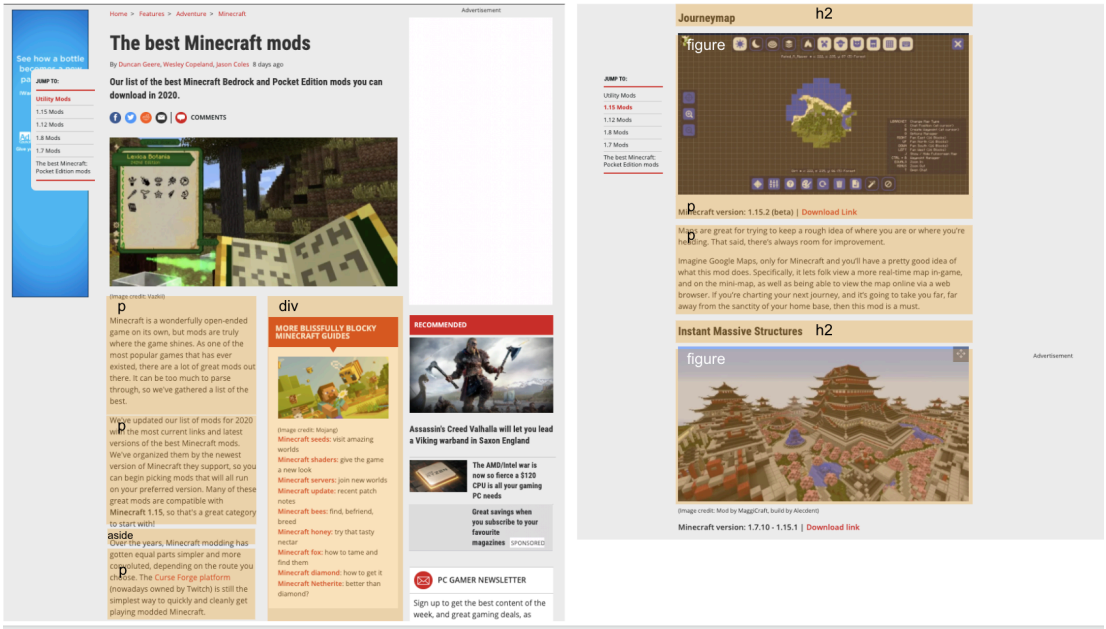


図 4.4: 主コンテンツ検出モデルの素性の重要度

「兄弟タグの句読点の割合」(0.141) といったように、兄弟タグに関する素性が上位にあり、これらが主コンテンツの検出に比較的有効であることがわかる。これはテキストの量が多いウェブサイトにおいて、主コンテンツが複数の兄弟タグによって構成されることが多いためと考えられる。例えば、図 4.5 は主コンテンツを全て抽出することに成功したウェブサイトである。図で網掛けしてある部分が主コンテンツである。テキストを含む p タグ、見出しを表す h2 タグ、画像と文字情報を含む div タグ、画像を含む figure タグ、スペースを表す aside タグなどがあり、これら全てが主コンテンツである。これらは全て兄弟タグとなっている。これらのタグについて、「兄弟タグのテキスト長の割合」の素性の値は 0.8、「兄弟タグの句読点の割合」の素性の値は 0.9 を超える。主コンテンツが持つ要素はテキスト、見出し、画像、スペースなどそれぞれ異なっており、figure タグや aside タグなどはテキスト情報を一切持たないため、タグ自体から得られる素性だけで主コンテンツとして検出することは難しいと考えられるが、兄弟タグに関する素性を導入することで主コンテンツとして正しく検出されている。

一方、主コンテンツとそうでないテキストが兄弟タグの関係にあるため、主コンテンツ以外のタグを誤って主コンテンツと判定したときもあった。その例を図 4.6 に示す。このウェブサイトでは、オレンジ色で示した主コンテンツと、薄い赤で示した閲覧者が書込むコメント欄が兄弟タグの関係にある。このコメント欄は、主コンテンツである記事以上に兄弟タグのテキストが長いため、主コンテンツと



(引用元 URL https://www.pcgamer.com/best-minecraft-mods/)

注: 1 つのウェブページを左右に分割して表示している。

図 4.5: 主コンテンツが兄弟タグとなっているウェブサイトの例

して誤って検出された。このように、兄弟タグの素性を導入することによって検出誤りが発生する事例も見られた。しかし、素性の重要度が相対的に高いことから、全体的には兄弟タグに関連する素性は主コンテンツ検出に有効であると結論付けられる。

主コンテンツ

主でないコンテンツ

(引用元 URL https://news.nicovideo.jp/watch/nw3519966)

図 4.6: 主コンテンツの誤検出の例

次ページリンクからの距離の素性の有効性の検証

ここでは、主コンテンツ検出モデルの学習に用いた素性のうち、「次ページリンクからの距離」の素性の有効性を詳細に検証する。この素性は、判定対象のタグと次ページリンクの近さを表すものである。ページ送りされているウェブページにおいては、主コンテンツと次ページリンクは近くに配置される傾向があることから、この傾向を検出モデルに反映するために導入した。この素性は、ページ送りのないウェブページから主コンテンツを検出する際には当然使用できない。つまり、ページ送りされているウェブページからの主コンテンツ検出だけに使用される特徴的な素性である。そのため、その有効性を実験的に確認する。

まず、「次ページリンクからの距離」の素性を用いるモデルと用いないモデルを比較する。結果を表 4.7 に示す。(b) は表 4.6 の再掲である。図 4.7 は「次ページリンクからの距離」の素性を用いないモデルの PR 曲線である。

表 4.7: 次ページリンクからの距離素性の効果の検証

(a)「次ページリンクからの距離」の素性なし

モデル	精度	再現率	F 値
ベースライン	0.032	0.002	0.003
決定木	0.323	0.941	0.480
ランダムフォレスト	0.474	0.536	0.504
GBDT	0.536	0.900	0.499

(b)「次ページリンクからの距離」の素性あり

モデル	精度	再現率	F 値
ベースライン	0.032	0.002	0.003
決定木	0.300	0.954	0.451
ランダムフォレスト	0.578	0.584	0.581
GBDT	0.415	0.711	0.524

「次ページリンクからの距離」の素性を用いないモデルにおいても、提案手法はベースラインを大きく上回り、またランダムフォレスト、GBDT、決定木の順に F 値が高い。図 4.7 の PR 曲線を見ても、ランダムフォレストが一番性能が良いことが確認できる。

表 4.7 の (a) と (b) を比較すると、「次ページリンクからの距離」の素性を用いることにより、F 値が大きく改善されていることがわかる。図 4.4 を見ると、この素性の重要度は 0.077 とそれほど高くはないが、ランダムフォレストと GBDT については、この素性を追加することによって精度、再現率、F 値のいずれもが向上していることから、他の素性と組み合わせることによって効果を発揮すると考えられる。

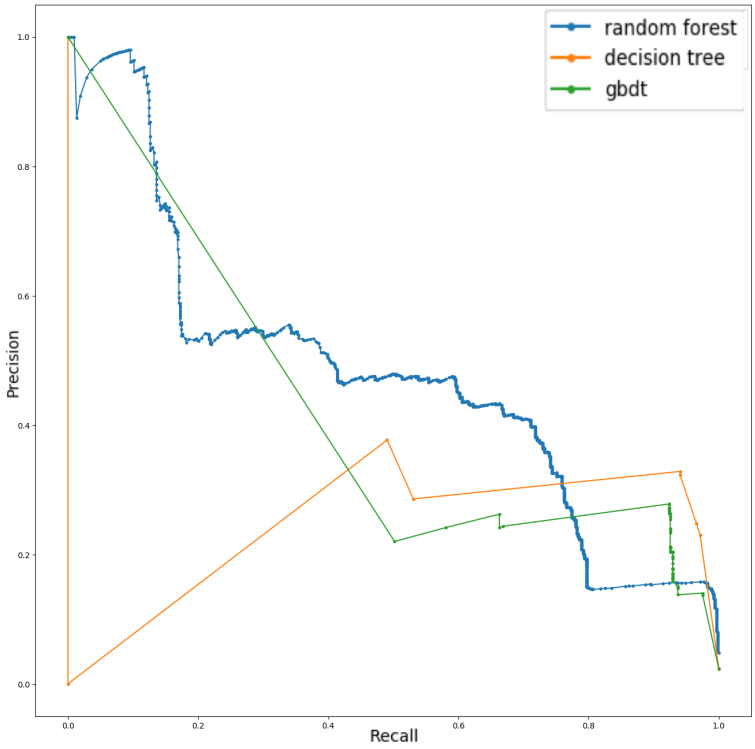


図 4.7: 提案手法による主コンテンツ検出の PR 曲線 (次ページリンクからの距離素性なし)

「次ページリンクからの距離」の素性が有効に働いた例を図 4.8 に示す。このウェブサイトでは、主コンテンツ (オレンジ色で示したブロック) の他に、テキスト量が多く主コンテンツとみなされやすいが実際には主コンテンツではないタグ (薄い赤で示したブロック) が存在する。DOM ツリーでは、主コンテンツの次ページリンクタグからの距離は 6、主でないコンテンツの次ページリンクタグからの距離は 11 となり、相対的に主コンテンツの方が次ページリンクタグの近くに現れる。したがって、「次ページリンクからの距離」の素性により主コンテンツのタグがそうでないタグよりも近い位置にあることが捉えられ、またテキストの量など他の素性を組み合わせることで、主コンテンツのみを正しく検出できたと考えられる。

次に、テストデータに対して「次ページリンクからの距離」の素性を抽出する際に、正解の次ページリンクの情報を用いる代わりに、自動推定した次ページリンクの情報を用いる実験を行う。次ページリンクを自動推定することにより、実際に提案手法を未知のデータに適用するときと同じ条件で主コンテンツ検出の性能を評価する。また、次ページリンク検出モデルと主コンテンツ検出モデルの両方を同時に評価しているとも言える。なお、分類器を学習する際、訓練データから「次ページリンクからの距離」の素性を抽出するときには正解の次ページリンクの情報を用いることに注意していただきたい。

この実験では、以下の手続きによって「次ページリンクからの距離」の素性を

図 4.8: 次ページリンクからの距離素性が有効に働いた例

抽出する。

- テストデータに対して提案手法の次ページリンク検出モデルを適用し、次ページリンクを検出する。F 値が一番高いランダムフォレストの分類器を使用する。
- 次ページリンクを含むタグのうち、HTML ファイル上に最初に出現したタグを次ページリンクタグとみなす。複数のリンクを次ページリンクとして検出した場合は、全ての内で最初に出現したタグを次ページリンクタグとみなす。
- 次ページリンクタグと判定対象のタグとの DOM ツリー上の距離を求め、「次ページリンクからの距離」の素性値とする。
- モデルによって 1 つも次ページリンクを検出できなかったテストデータについては、次ページリンクを自動検出できたテストデータから「次ページリンクからの距離」の中央値を計算しその値を素性として用いた。

表 4.8 は、「次ページリンクからの距離」の素性を使わない場合、正解の次ページリンクタグを参照して「次ページリンクからの距離」の素性を使う場合、上記の手続きによって自動推定された次ページリンクタグを参照する場合を比較した実験結果を示している。予想された通り、正解の次ページリンクの情報を使うモデルの F 値が一番高いが、次ページリンクを提案手法によって自動推定したときの F 値は 0.571 であり、その差は 0.01 ポイントと小さい。また、次ページリンクを自動推定するときでも、「次ページリンクからの距離」の素性を使わないモデルと比べて F 値が大きく改善されている。

図 4.9 は、「次ページリンクからの距離」の素性を使わない場合 (without next link)、正解の次ページリンクタグを参照して「次ページリンクからの距離」の素性を使う場合 (actual next link)、自動推定された次ページリンクタグを参照する

表 4.8: 次ページリンクタグを付与する手法の違いによる主コンテンツ検出モデルの性能比較

次ページリンクタグの付与	精度	再現率	F 値
なし	0.474	0.536	0.504
正解	0.578	0.584	0.581
自動推定	0.588	0.555	0.571

場合 (pred next link) のそれぞれの PR 曲線を示している。これを概観しても、多くの場合、正解の次ページリンクを使う場合 (actual next link) が最も性能がよく、ついで自動推定した次ページリンクを使う場合 (pred net link)、「次ページリンクからの距離」の素性を使わない場合 (without next link) の順であることがわかる。

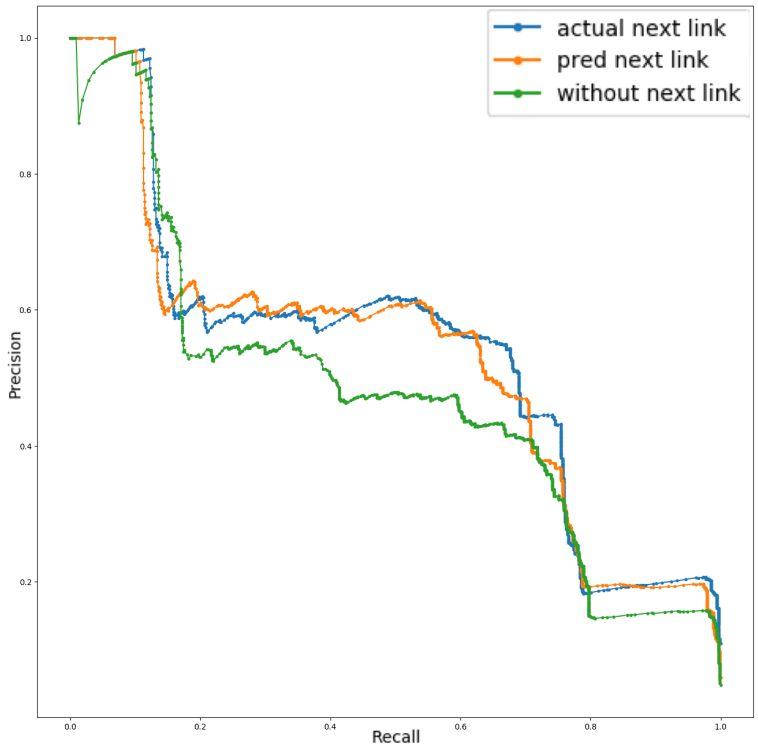


図 4.9: 次ページリンクタグを付与する手法別の PR 曲線

これらの結果から、正解の次ページリンクを用いずにこれを自動推定しても、「次ページリンクからの距離」の素性は主コンテンツ検出の性能向上に大きく貢献することが確認された。

4.5 考察

次ページリンク検出手法、主コンテンツ検出手法は、ともにベースラインを上回り、本研究で提案する素性を用いた分類器の学習が有効であることを確認した。

次ページリンク検出手法については、ランダムフォレストが最も良い性能を示した。「NEXT」「ページ」など次のページへのリンクを示唆するキーワードの素性が特に有効であった。LinkSimilarity 素性も他の素性と組み合わせることで性能向上に貢献した。

主コンテンツ検出手法については、「兄弟タグのテキスト長」、「兄弟タグのテキスト長の割合」、「兄弟タグの句読点の割合」の素性が有効であることがわかった。しかしながら、ウェブサイトによってはコメント欄など主コンテンツよりも兄弟タグのテキスト量が多いコンテンツが存在し、これを主コンテンツとして誤検出するという課題があることがわかった。

本研究の特徴である「次ページリンクからの距離」の素性について、これが主コンテンツ検出の性能向上に大きく貢献することを確認した。また、正解の次ページリンクではなく自動推定した次ページリンクの情報を用いて素性を抽出しても F 値が向上した。次ページリンクのタグと主コンテンツのタグは DOM ツリー構造上で近くに出現することが多く、この素性が主コンテンツ検出の有力な手がかりとなっていると考えられる。

今回の実験結果は提案手法の有効性を示してはいるが，次ページリンク検出の F 値は 0.750、主コンテンツ検出の F 値は 0.571 であり、改善の余地がある。本手法で用いた素性は全て HTML ソースファイルを解析した DOM ツリーから抽出されるが、主コンテンツ検出の関連研究ではブラウザ上の画面表示レイアウトの情報を素性に加えることで性能が向上することが報告されている。例えば、次ページリンクはウェブブラウザ上の表示サイズが小さく、主コンテンツは表示幅が大きく画面の中央上部に現れるなど、レイアウト上の特徴がある。このようなレイアウト情報を素性に追加することで次ページリンク検出や主コンテンツ検出の性能が向上する可能性がある。

第5章　おわりに

5.1　まとめ

本研究では、ページ送りを用いたウェブサイトから主コンテンツを抽出する手法を提案した。ページ送りで掲載された記事を一つにまとめるタスクに対し、これを教師あり機械学習による分類器の学習により実現するため、「次ページリンク検出タスク」、「主コンテンツ検出タスク」という2つのサブタスクを設計した。これら2つのタスクについて、次ページリンクならびに主コンテンツを正しく検出するための素性を考案することを中心に研究を進めた。さらに、分類器の機械学習アルゴリズムとして、決定木、ランダムフォレスト、GBDTを用い、これらを実験的に比較した。

まず、次ページリンクの検出のために有効な素性を検討した。次ページリンクであることを示す特定のキーワードの有無を表す素性やリンクテキストの長さなどタグ自体から取得できる素性を8つ考案した。これに加え、判定対象のリンクタグとその近傍にあるリンクタグの類似性も素性 (LinkSimilarity 素性) とした。評価実験の結果、提案手法は簡易な抽出ルールに基づくベースライン手法よりも高い性能を示した。3つの機械学習アルゴリズムの中ではランダムフォレストが最もF値が高く、その値は0.750であった。個々の素性の有効性を検証したところ、「次」、「NEXT」、「ページ」、「PAGE」といったテキストがタグ内に含まれていることが特に次ページリンクの検出に有効であることが確認された。また、LinkSimilarity素性を使用することでランダムフォレストによって学習された分類器のF値が向上することが確認された。

次に、主コンテンツ検出のために有効な素性を検討した。タグ自体に関する6個の素性に加えて、兄弟タグ、子タグなど周辺のタグに関する素性7個を設計した。これらに加え、次ページリンクタグと主コンテンツは近くに出現する傾向が強いという考えに基づき「次ページリンクタグからの距離」素性を追加した。実験の結果、提案手法はベースライン手法よりも高い性能を示した。次ページリンク検出タスク同様にランダムフォレストが最も性能が高く、そのF値は0.571であった。また、「次ページリンクタグからの距離」の素性を用いることでF値が0.07ポイント向上し、主コンテンツの検出のために有効な素性であることがわかった。これは、テストデータから「次ページリンクタグからの距離」の素性を抽出する際、正解の次ページリンクではなく提案手法によって自動推定された次ページリンクを用いるといった現実に近い実験条件にて確認された。

5.2 今後の課題

「次ページリンク検出タスク」、「主コンテンツ検出タスク」の両方において、提案手法の有効性を実験的に示したが、F 値自体はそれぞれ 0.750、0.571 と十分に高いわけではなく、これを改善することが今後の課題である。提案手法ではレイアウト情報は素性として一切利用していないが、次ページリンクは画面表示サイズが小さく、主コンテンツは画面の表示幅が大きく全体の中央上部に現れる傾向にあるなど、次ページリンク・主コンテンツ検出の手がかりになると考えられる。レイアウト情報の有効性を示す関連研究は多くあり、Song らは、ブラウザ上の水平画面表示サイズが大きいタグを重要だと重みづけすることで主コンテンツ検出の性能が向上することを示している [5]。レイアウト情報を素性として用いることで次ページリンクや主コンテンツの検出をより正確に行える可能性がある。

また今回利用した機械学習アルゴリズムは決定木、ランダムフォレスト、GBDT であるが、これらを学習する際のハイパーパラメータは十分に最適化されていない。一般に、機械学習されたモデルの品質はハイパーパラメータに大きく依存することが知られており、これを最適化することで次ページリンク検出、主コンテンツ検出の性能が向上する可能性がある。

関連図書

[1] 沢田洋平, 江渡浩一郎. 集合知による Web ページの構造情報の収集. 電子情報通信学会技術研究報告. AI, 人工知能と知識処理, Vol. 108, No. 208, pp. 27–32, 2008.

[2] 江渡浩一郎, 沢田洋平. 集合知データベース Wedata の構築と運用. 人工知能学会全国大会論文集 第 25 回全国大会 (2011), p. 3E3OS207. 一般社団法人 人工知能学会, 2011.

[3] Erdinç Uzun, Hayri Volkan Agun, and TarıK Yerlikaya. A hybrid approach for extracting informative content from web pages. *Information Processing & Management*, Vol. 49, No. 4, pp. 928–944, 2013.

[4] Jun Zeng, Feng Li, Brendan Flanagan, and Sachio Hirokawa. LTDE: A layout tree based approach for deep page data extraction. *IEICE TRANSACTIONS on Information and Systems*, Vol. 100, No. 5, pp. 1067–1078, 2017.

[5] Dandan Song, Fei Sun, and Lejian Liao. A hybrid approach for content extraction with text density and visual importance of DOM nodes. *Knowledge and Information Systems*, Vol. 42, No. 1, pp. 75–96, 2015.

[6] Shanchan Wu, Jerry Liu, and Jian Fan. Automatic Web content extraction by combination of learning and grouping. In *Proceedings of the 24th international conference on World Wide Web*, pp. 1264–1274, 2015.

[7] 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. *DBSJ Journal*, Vol. 8, No. 1, pp. 1–6, 2009.

[8] W3C. The global structure of an html document, 1999. `\textbf{https://www.w3.org/TR/html401/struct/global.html}`.

[9] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, Vol. 16, pp. 321–357, 2002.

[10] Yoshikiyo Kato, Daisuke Kawahara, Kentaro Inui, Sadao Kurohashi, and Tomohide Shibata. Extracting the author of web pages. In *Proceedings of the 2nd ACM workshop on Information credibility on the web*, pp. 35–42, 2008.