カスタマイズ可能なWebページサムネイル生成のための DOMノード役割推定

前田　直宏†　　山本　岳洋†

† 兵庫県立大学 大学院情報科学研究科　〒 651–2197 兵庫県神戸市西区学園西町 8-2-1
E-mail: †ad22p062@gsis.u-hyogo.ac.jp, ††t.yamamoto@sis.u-hyogo.ac.jp

あらまし　本研究では，HTML 文書から Web ページに対する評価数や作成日，タイトルなどの Web ページ自体の属性を抽出することに取り組む．そのために HTML 文書を木構造で表現した DOM を用いてテキストを含むノードを対象とし，Web ページ自体の属性を持つノードであるかを分類する．具体的には DOM の各ノードが持つテキストや XPath，class 属性値，レイアウト情報などの特徴量を用いて，ノードが Web ページ中でどのような役割を担っているかを推定し分類することで，ノードに含まれているテキストを抽出する．実験ではプログラミング情報 Q&A サイトを対象としてノードの分類を行い，質問内容や回答数，評価数などの属性を抽出し，提案手法の有効性を検証する．評価尺度として，適合率，再現率，F_1 を用いて性能を評価する．Web ページからの属性抽出に関する既存モデルと提案手法を比較することで，提案手法の有効性を検証する．

キーワード　情報検索，情報抽出，属性抽出，DOM

1 はじめに

Web ページの概要を表すサムネイルは，ユーザ所望の情報を獲得する助けになる．例えば検索結果の表示をテキストベースでけではなく，Web ページ内の画像やデザインを用いた 1 枚のサムネイルも表示することで，探している情報と関係のある情報をより正確に獲得することが可能である [3]．また，一度閲覧した Web ページの約 80% は再度閲覧することが明らかになっている [1] [2]．ブラウザに備わったブックマーク機能や閲覧履歴，記憶を辿ることで再度閲覧したページにたどり着くことができる一方で，検索結果表示欄にサムネイルも表示することで，一度訪れたサイトへの再訪問が容易に行えることが報告されている [12]．このように，ユーザが情報を獲得するとき，Web ページの概要を表したサムネイルは情報検索の効率を上昇させる．

このような Web ページの概要を表すサムネイルにおいて，表示項目の選択は重要である．ここで表示項目とは，ロゴやタイトル，写真といったサムネイルに使用する部品のことを指す．例えば，商品購買サイトを例に考える．この場合，商品の写真と販売サイト名もしくはロゴ，そして値段の三つが表示項目として考えられる．他にもプログラミング情報記事では，その記事が書かれた日付や「いいね」などの評価数が重要であると考える．プログラミング言語のような情報技術は日々開発が行われているため，これらの情報を探すユーザは最新の情報を求めていることが予想される．また多くの人から評価を得ている記事であれば，よりわかりやすく参考になる記事であることが期待できる．また Q&A サイトでは，質問内容に対する回答数がある．ユーザが探している情報は質問に対する回答であり，回答数が多いほど多様な意見や正確な回答が含まれている可能性がある．

このようにユーザが検索する内容によって把握したい情報が異なるため表示項目の選択は重要である．ユーザ側がサムネイルをカスタマイズできるサムネイルを生成し，図 1 のように検索結果欄にも表示することでより検索支援に繋がると考える．そこで本研究ではサムネイル生成のために必要となる表示項目，Web ページに対する評価数や記事の作成日，タイトルなどの属性を Web ページから抽出することに取り組む．Web ページの HTML 文書を木構造で表現した DOM を用い，既存の属性抽出モデル [15] を拡張する．具体的には，DOM の各ノードが持つテキストや XPath，class 属性，レイアウト情報などの特徴量を用いて，ノードが Web ページ中でどのような役割を担っているかを推定し分類することで，テキストの抽出を行う．モデルの性能評価実験として，プログラミング技術情報 Q&A サイトを対象としてノードの分類を行い，質問タイトルや回答数，記事作成日などの属性を抽出した．評価尺度として適合率，再現率，F_1 値，ROC 曲線の AUC を用いて提案手法の有効性を検証する．

2 関連研究

2.1 サムネイルを用いた情報検索支援

Web ページの概要を表したサムネイルについての研究が行われている．Zhou らは Web ページを検索する時に使用したクエリを強調するサムネイルを提案している [14]．Teevan らは Web ページのタイトル，ロゴ画像，注目度が高い画像を用いたサムネイルを提案している [12]．また，稲垣らは Web ページに対する顕著性マップを用いたサムネイルを提案している [16]．顕著性マップとは，人が画像を認識する際に注視しやすい領域を画像で定量的に表現したものである．Web ページを閲覧す

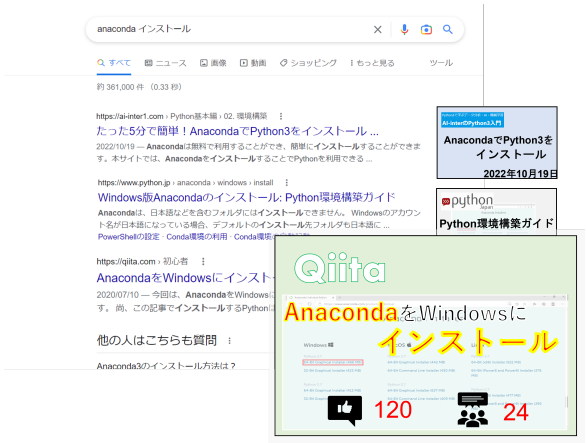

図 1　Web ページの概要を表したサムネイル.

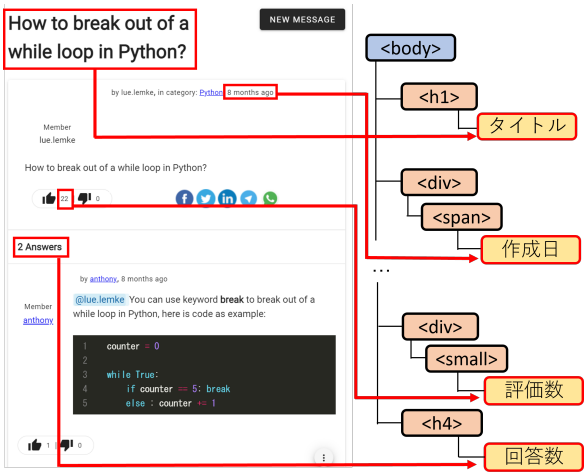

図 2　Web ページにおける DOM ノードの分類による属性抽出.

る際に注視されやすい上位 10 の領域を 1 枚の画像に並べて配置した集約画像を生成することで，ページ内容の理解補助に繋げている．これらのサムネイルで使用している画像は，Web ページ内に含まれる画像を抽出し使用している．しかし内部画像が存在しない場合，サムネイルを作成することが困難であった．そこで Jiao らはインターネット上からその Web ページと関連性が高い画像を取得し，その画像を用いたサムネイルを提案している [6]．Web 検索時以外にも Web ページの概要を表したサムネイルの利用方法が提案されている．例えばサムネイルを Web ブラウザのタブ管理に使用することが提案されている [10]．Web ページの左右余白部分に閲覧した Web ページのサムネイルを配置することで，一度閲覧した Web ページへの素早い切り替えを行うことができる．

2.2　Web ページからのテキスト情報抽出

サムネイルを生成する時にはタイトルやロゴなどの表示項目を Web ページ中から抽出する必要がある．Web ページ中からそれらの項目を抽出する手法としては，SVM（Support Vector Machine）や深層学習を用いた手法が提案されている．廖らは，SVM を用いたイベント情報の抽出を提案している [18]．イベント情報が含まれている Web ページ中の文章を対象とし，10,000 次元以上の単語ベクトル空間のような高次元特徴空間に対し SVM を使用することで実現している．Web ページに含まれるテキストのみを使用するだけでなく，Web ページを構成するマークアップ言語も使用するアプローチも提案されている．それらの手法では HTML などのマークアップ言語で記述された文書を操作するために，木構造で表現した DOM（Document Object Model）を用いる．Vogels らは DOM と隠れマルコフモデルを用いて，Web ページ中のテキストブロックがメインコンテンツなのか，ボイラープレートであるかを分類するモデルを提案している [13]．Qiang らは DOM を使用し，書籍やレストランなどの Web ページから，属性（書籍のタイトルや著者，出版日など）を抽出するモデルを提案している [5]．Anurendra らは，DOM に加え Web ページのスクリーンショットといった視覚的情報も使用し．商品購買サイトから商品の値段や商品名，商品の画像を抽出している [8]．また Gogar らは畳み込みニューラルネットワークを使用し，テキストと Web ページ全体画像を用いて商品情報を抽出している [4]．鶴田らはノードの大きさといったレイアウト情報のみを使用して Web ページ中の主要なノードを推定している [17]．Lin らは，特徴量としてレイアウト情報とテキスト情報を使用した [9]．レイアウト情報としては，Web ページ上で表示されているノードの位置を使用する．またテキスト情報としては，ノードに含まれるテキストや XPath の末端（/html/body/div/h1 の場合 “h1”）を使用したモデルを提案している．Zhou らは Lin らのモデルを拡張し，モデルの性能を高めた [15]．Lin らの提案モデルは，テキスト情報においては分類対象となるノードだけに含まれる情報を特徴量としている．一方で Zhou らのモデルは分類対象ノードだけでなく，親ノードや兄弟ノードにあるテキスト情報も使用することで，分類精度を向上させている．

本研究では Zhou らが提案したモデル（LANTERN）を拡張し，Web ページ自体の属性（Web ページのタイトルや記事作成日，記事作者名など）を抽出する．

3　Web ページからの属性抽出

3.1　問 題 定 義

本研究では HTML 文書で書かれた Web ページから，その Web ページ自体の属性を抽出する．ここで属性とは，図 2 のように Web ページのタイトルや記事作成日，評価数などを指す．ある Web ページが持つ DOM ノード集合 X について，各ノード x（$x \in X$）が持つテキストがどの属性（タイトル，記事作成日，その他など）に属するかを分類するモデルを構築する．ここで「その他」とは，ノードがどの属性にも当てはまらないことを意味する．Web ページからの属性抽出を DOM ノードの多クラス分類問題として定式化する．

3.2　属性抽出モデル LANTERN [15]

本節では Web ページから属性抽出を行うモデルの一つである LANTERN [15] について説明する．LANTERN はまず始めに，HTML 文書を木構造で表現した DOM に変換する．得

られた DOM に含まれているノードの内，HTML の開始タグと終了タグに囲まれたテキストが含まれているノードのみを分類の対象とする．ノードが持つテキストは，テキストエンコーダーに入力される．また，XPath や XPath の末端などのマークアップ言語の情報もモデルに入力する．近接ノードとの意味的類似度や，DOM 中におけるノードの相対的な位置なども特徴量として使用する．各特徴量を全結合層に入力し，その後ソフトマックス関数を適用し予測を行う．

3.2.1 近接ノード

DOM 中のあるノード x が与えられた時，friend ノード x^f と partner ノード x^p を定義する．friend ノード x^f は x の兄弟ノードであり，x1 つにつき最大で 10 存在する．partner ノード x^p は friend ノード x^f の内，最も x と距離が近いノードを指し，x1 つにつき 1 つだけ存在する．各ノード x，x^f，x^p が持つテキストをテキストエンコーダーに入力することで，d_w 次元の特徴量ベクトル $\boldsymbol{e}_x$，$\boldsymbol{e}_f$，$\boldsymbol{e}_p$（$\boldsymbol{e}_x$，$\boldsymbol{e}_f$，$\boldsymbol{e}_p \in \boldsymbol{R}^{d_w}$）を得る．これら 3 つのベクトルを結合し，新たに特徴量ベクトル $\boldsymbol{e}_s$ を得る．

$$\boldsymbol{e}_s = [\boldsymbol{e}_x; \boldsymbol{e}_f; \boldsymbol{e}_p] \quad \boldsymbol{e}_s \in \boldsymbol{R}^{3d_w} \tag{1}$$

3.2.2 テキストエンコーダー

テキストエンコーダーは CNN（Convolutional Neural Network）と Bidirectional-LSTM（Long Short-Term Memory）で構成されており，各ノードが持つテキストが入力される．CNN では，テキストを単語レベルでの埋め込みと文字レベルでの埋め込みを処理する．LSTM は連続データを扱うことができる RNN（Reccurrent Neural Network）を拡張したニューラルネットワークである．連続的な情報を扱うことができる点から自然言語処理の分野でも応用されている．Bidirectional-LSTM では LSTM で学習できないテキストの前後の文脈を学習する．

3.2.3 意味的類似性

意味的類似性とは，言語間の意味類似の程度を表す指標である．ある任意のノード x に対し，その partner ノード x^p には，ノード x の属性クラスを決定するのに大きな役割を持つテキストが存在することがある．例えば，本の著者名を持つノードの周辺には “by” という単語を持つテキストノードが存在する．この “by” は後の単語が本の作者である可能性が高いことを意味する．ノード x と最も距離が近いノード x^p と抽出する属性値間でコサイン類似度を計算し，K 次元の特徴量ベクトル $\boldsymbol{e}_{\cos}$（$\boldsymbol{e}_{\cos} \in \boldsymbol{R}^K$）を取得する．ここで K は分類する属性の数である．

3.2.4 ノードの位置特徴量

DOM 中における各ノードの相対的な位置を特徴量として使用する．例えば，Web ページのタイトルや記事作成日は Web ページ上部に書かれていることが多いため，DOM に対し深さ優先探索で各ノードの位置 pos_x を取得する．相対的位置は以下の式で与えられる．

$$\text{相対的位置} = \frac{\mathrm{pos}_x}{\max\{\mathrm{pos}_x\}} \quad x \in X \tag{2}$$

式 (2) を用いて d_{pos} 次元の特徴量ベクトル $\boldsymbol{e}_{\mathrm{pos}}$（$\boldsymbol{e}_{\mathrm{pos}} \in \boldsymbol{R}^{d_{\mathrm{pos}}}$）を取得する．

3.2.5 XPath Embedding

XPath を用いることで DOM ツリーで表されるある任意のノードを特定することができる．あるノードの XPath（/html/body/tr/td）が与えられたとき，この XPath は [<html>，<body>，<tr>，<td>] の HTML タグで構成されている．この連続するタグに対して Bi-LSTM を使用し，d_{xpath} 次元の特徴量ベクトル $\boldsymbol{e}_{\mathrm{xpath}}$（$\boldsymbol{e}_{\mathrm{xpath}} \in \boldsymbol{R}^{d_{\mathrm{xpath}}}$）を取得する．

3.2.6 Leaf Type Embedding

XPath の末端はノードの特徴を強く表す．例えば <h1> タグは Web ページのコンテンツテーマを表す HTML タグであるため，Web ページのタイトルを表すノードである可能性が高いと考えられる．この XPath の末端を用いて d_{leaf} 次元の特徴量ベクトル $\boldsymbol{e}_{\mathrm{leaf}}$（$\boldsymbol{e}_{\mathrm{leaf}} \in \boldsymbol{R}^{d_{\mathrm{leaf}}}$）を取得する．

最終的にこれらの特徴量を連結し，以下の特徴量ベクトル $\boldsymbol{e}_d$ を得る．

$$\boldsymbol{e}_d = [\boldsymbol{e}_{\mathrm{xpath}}; \boldsymbol{e}_{\mathrm{leaf}}; \boldsymbol{e}_{\mathrm{pos}}; \boldsymbol{e}_{\cos}] \quad \boldsymbol{e}_d \in \boldsymbol{R}^{d_{\mathrm{xpath}}+d_{\mathrm{leaf}}+d_{\mathrm{pos}}+K} \tag{3}$$

3.3 class 属性値に着目したモデル拡張

本研究では，Web ページ自体の属性であるタイトル名や記事作成日などを抽出する．そこで，HTML データに含まれる class 属性に着目した．class 属性とは DOM ノードの種類を示すものであり，ノードがどのような役割を担っているかが記載されていることが多い．例えば日付を表すノードの class 属性には，<class=“date”>のように日付を意味する語句（以下 class 属性値）が記載されていることがある．この class 属性値をモデルへ入力する特徴量に追加することで Web ページの構造を捉え，Web ページ自体の属性を抽出する．図 3 にモデルの概要を示す．クラス属性値をテキストエンコーダーに入力することで d_w 次元の特徴量ベクトル $\boldsymbol{e}_{\mathrm{class}}$（$\boldsymbol{e}_{\mathrm{class}} \in \boldsymbol{R}^{d_w}$）を得る．このベクトル $\boldsymbol{e}_{\mathrm{class}}$ を式 (1) の $\boldsymbol{e}_s$ と結合し，新たに特徴量ベクトル $\boldsymbol{e}_s$ を定義する．

$$\boldsymbol{e}_s = [\boldsymbol{e}_x; \boldsymbol{e}_f; \boldsymbol{e}_p; \boldsymbol{e}_{\mathrm{class}}] \quad \boldsymbol{e}_s \in \boldsymbol{R}^{4d_w} \tag{4}$$

3.4 レイアウトに着目したモデル拡張

レイアウト情報として，ノードに含まれるテキストのフォントサイズ fontsize，ノードの高さ height，横幅 width，ブラウザ上での描画面積 area を使用する．各値はレンダリングを行い，CSS プロパティから取得する．得られた値から d_{layout} 次元の特徴量ベクトル $\boldsymbol{e}_{\mathrm{layout}}$（$\boldsymbol{e}_{\mathrm{layout}} \in \boldsymbol{R}^{d_{\mathrm{layout}}}$）を得る．ただし，

$$\mathrm{area} = \mathrm{height} \times \mathrm{width} \tag{5}$$

である．

3.5 モデルの学習

抽出するノードの属性値ラベル集合を L，分類対象とする DOM ノード集合を X とし，以下のように表す．

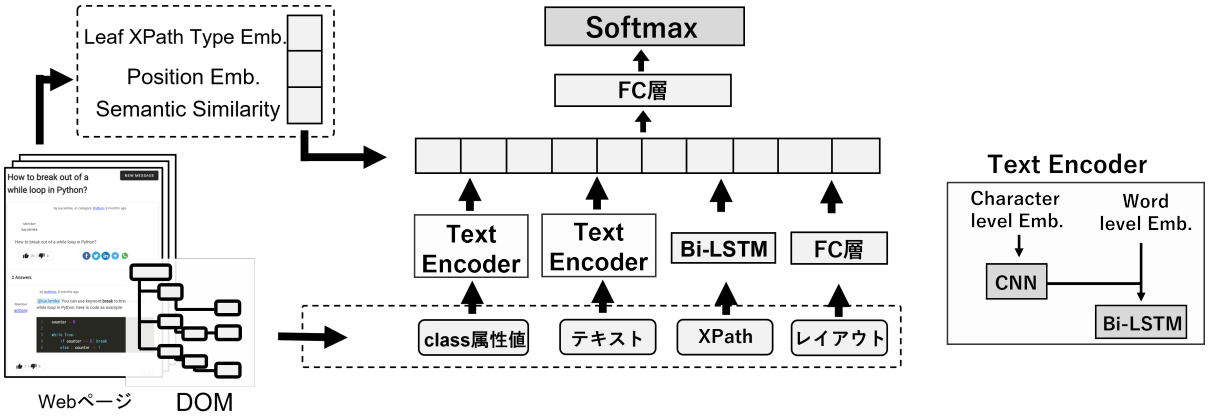

図 3　特徴量として class 属性値を追加した DOM ノード分類モデル.

$$L = \{l_1, l_2, ..., l_K\}$$
$$X = \{x^{(1)}, x^{(2)}, ..., x^{(N)}\}$$

ここで K は抽出する属性の個数，N はノードの総数を表す．式 (3) と式 (4) で定義した特徴量ベクトル $\boldsymbol{e}_d$ と $\boldsymbol{e}_s$ を結合し，新たに特徴量ベクトル $\boldsymbol{e}_n$ を定義する．

$$\boldsymbol{e}_n = [\boldsymbol{e}_s; \boldsymbol{e}_d] \tag{6}$$

この特徴量ベクトル $\boldsymbol{e}_n$ を全結合層に入力し，K 次元のベクトル $\boldsymbol{h}$（$\boldsymbol{h} \in \boldsymbol{R}^K$）を得る．その後ソフトマックス関数を適用し，各ラベルの予測確率 $\boldsymbol{p}$ を得る．

$$\boldsymbol{p} = \mathrm{softmax}(\boldsymbol{h}) \tag{7}$$
$$\hat{y}^i = \underset{k \in 1,...N}{\arg\max}\, p_k \tag{8}$$

抽出したい属性値以外の多くの情報が含まれている．そのため，HTML データにはどの属性値にも属さない「その他」属性が多く含まれる不均衡データである．そこで，学習時にクラスごとに重みを調整する．具体的には，損失関数として重み付き交差エントロピー誤差を用いる．損失関数は以下の式で表される．

$$E(X) = -\sum_{i=1}^{N}\sum_{k=1}^{K} w_k t_k^{(i)} \log y_k^{(i)} \tag{9}$$

$$t_k^{(i)} = \begin{cases} 1 & if \quad x^{(i)} \text{が} l_k \text{のノード} \\ 0 & \text{それ以外のノード} \end{cases} \tag{10}$$

ここで K は属性の数，N はノードの総数を表し，$y_k^{(i)}$ はモデルが出力した各ノードの推定属性である．また，w_k は属性ラベル l_k における重みである．t_k は各属性 l_k について，ノード $x^{(i)}$ がその属性 l_k を持つ場合は 1 を，持たない場合を 0 とする変数である．損失関数にクラスごとの重みを加えることで誤分類したときのペナルティを大きくし，負例への分類が多くなることを防ぐ．

4　評 価 実 験

4.1　使用するデータ

プログラミング技術の質問サイトである StackOverflow, OpenNMT，CodeProject，FindNerd，DevHubby を使用する．各 Web サイトについて，CSS データを除いた HTML データを 100 ページずつ，計 500 ページ収集した．ただし日本語ではなく英語のサイトを収集した．収集した HTML データを木構造に変換し，各テキストノードに抽出したい属性値のラベル付けを行った．具体的には,「作成日」,「作者」,「質問タイトル」,「回答数」,「記事に対する評価数」を意味するテキストに対しラベルを付与した．また，いずれにも当てはまらないテキストを持つノードに対しては「その他」のラベルを付与した．全ノード数は 163,818 ノードである．Web ページ中のテキストを含むノードは大半が「その他」のラベルを持つノードである．

4.2　負例フィルタリング

DOM には抽出したいテキストを持つノード（正例）よりも，それ以外のノード（負例）が圧倒的に多い．モデルに入力するデータの正例と負例の分布に大きな差があると，モデルの分類精度は低下する．そこで負例の数を減らす負例フィルタリングを行う．方法としては以下のルールに従うノードを負例とみなし，負例をフィルタリングする．

（1） テキストを含まないノードである．
（2） テキストが絵文字もしくは記号の文字だけで構成されている．
（3） 30 単語以上のテキストである．
（4） テキストがアルファベット 1 文字だけである．
（5） ノードのブラウザ上での描画面積が 0 である．
（6） ノードの XPath 中に特定のタグが含まれている．

XPath 中の特定のタグを表 1 に示す．Web ページ中のヘッダーや表内，プログラムのコード内，入力欄などには抽出項目が存在しないと考えられるため，表 1 のタグを決定した．

4.3　実 験 条 件

収集した 5 つの Web サイトを用いて実験を行う．4 つの Web サイトの計 400 ページを訓練データとし，残りの 1Web サイト計 100 ページを評価データとした 5 分割交差検証を行い評価をする．この分割方法は，評価データの中に訓練データとして使用した Web サイトを含まないようにするための分割方法である．1 つの Web サイトに含まれる 100 件の Web ページ

表 1　HTML タグとタグで囲まれた要素の役割.

対象の HTML タグ	タグが表す要素の役割
button	クリックできるボタンを表す
code	プログラムのコードを表す
footer	フッターを表す
header	見出しやナビゲーションを表す
input	ユーザからの入力を受け取る
label	入力欄に対してキャプションを付ける
nav	ナビゲーションリンクを提供する
p	テキストの段落を表す
sup	下付文字を表す
table	表形式のデータを表す
u	文字に下線を引くために使用する

は類似した HTML 構造をしているため，Web サイト単位で分割し学習を行う．エポック数は 50，バッチサイズは 126 とし，モデル学習の最適化アルゴリズムとして Adam（Adaptive moment）[7] を使用する．Adam のパラメータは，$\beta_1 = 0.9$, $\beta_2 = 0.99$ とし，学習率は 1.0×10^{-5} とした．単語分散表現として，GloVe（Global Vectors）[11] を使用する．特徴量の次元数 $\boldsymbol{d}_w$，$\boldsymbol{d}_{\text{xpath}}$，$\boldsymbol{d}_{\text{leaf}}$，$\boldsymbol{d}_{\text{pos}}$，$\boldsymbol{d}_{\text{class}}$，$\boldsymbol{d}_{\text{style}}$，はそれぞれ 100, 30，30，20，100，200 とした．損失関数は重み付き交差エントロピーを使用している．その重み w として,「その他」ラベルに対しては 1，それ以外の属性ラベルに対しては 10 を設定した．HTML データから DOM への変換には Python の LXML ライブラリを使用する．また，CSS プロパティの値は Python の Selenium ライブラリを使用して JavaScript を動かし，レンダリングを行うことで取得する．

4.4 評価尺度

ノードの予測を多クラス分類として性能を評価する．ただし，モデルの評価方法として 2 つの方法をとる．1 つ目は，モデルが出力したノード全てを評価対象とする方法である．2 つ目は，モデルが出力したノードの内，1Web ページにつきクラス予測確率が最も高いノード 1 つを評価対象とする方法である．1 つの Web ページに正例は 5 つ存在するので，1Web ページにつき 5 ノードが評価対象となる．以下に各評価方法について詳しく述べる．

4.4.1 評価方法 1

モデルが出力したノード全てを評価対象とする方法について述べる．ノードをクラス l_k と分類するタスクにおいて，実際にクラス l_k である場合を正例，そうでない場合を負例とする．クラスが l_k であるノードを実際に正例とモデルが予測できたときを真陽性（TP），負例と予測したときを偽陰性（FN），クラスが l_k でないノードを正例とモデルが予測したときを偽陽性（FP），不例と予測したときを真陰性（TN）とする．評価尺度として，適合率（Precision），再現率（Recall），F_1 値，ROC（Receiver Operating Characteristic）曲線の AUC（Area Under Curve）を用いる．ROC 曲線とは，正例と負例に分類する際の閾値を変更したときの真陽性率 TPR（True Positive Rate）と偽陽性率 FPR（False Positive Rate）の値をプロットした曲線である．ここで，TPR と FPR は以下の式で求められる．

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{11}$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{12}$$

AUC とは，ROC 曲線の下部分の面積のことである．AUC の値が 1 に近ければ近いほどモデルの性能が良いことを意味する．モデルがランダムな分類を行っている場合，AUC の値は 0.5 に近づき，正例と負例を正確に分類できている場合，AUC の値は 1 に近づく．なお，ROC-AUC は一つのクラスとそれ以外の全クラスの 2 値分類として評価を行う One vs Rest（OvR）で計算する．具体的には,「日付」の ROC-AUC を計算する場合は,「日付クラス」と「日付以外の全クラス」という形で計算する．これを各クラスで計算することで，最終的に多クラス分類として評価する．また，適合率 P，再現率 R，F_1 値 F_1 は以下の式で求められる．

$$P = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{13}$$

$$R = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{14}$$

$$F_1 = \frac{2\text{PR}}{\text{P} + \text{R}} \tag{15}$$

これらの評価尺度は，各クラスの評価値を平均した全体の評価値（マクロ平均）を用いる．マクロ平均とは，各クラスの評価指標の平均である．マクロ平均を使用することで各クラスのサンプル数の偏りに影響を受けることなく評価指標を計算することができる．マクロ平均適合率 P_{macro}、マクロ平均再現率 R_{macro}，マクロ平均 F_1 値 F_{macro} は以下の式で求められる．

$$P_{\text{macro}} = \frac{1}{K}\sum_{i}^{K} P_i \tag{16}$$

$$R_{\text{macro}} = \frac{1}{K}\sum_{i}^{K} R_i \tag{17}$$

$$F_{\text{macro}} = \frac{1}{K}\sum_{i}^{K} F_i \tag{18}$$

ここで K は分類する属性の数である．

4.4.2 評価方法 2

モデルが出力したノードの内，1Web ページにつきクラス予測確率が最も高いノード 1 つを評価対象とする方法について述べる．本研究で解く問題は，入力した Web ページの作成日や評価数を抽出することである．そのため，モデルが最も高い確率で予測したノードが抽出したいテキストを含んでいればよい．そこで，各クラスについてモデルが最も高い確率で予測したノードが正例であるかを評価する．正例を分類できるかを評価するため，評価尺度としては適合率を用いる．適合率は式 (13) を用いて計算される．

以上の評価尺度を用いてモデルの性能評価を行う．

4.5 実験結果

4.5.1 評価方法 1 についての実験結果

各属性ごとの抽出結果を表 2，表 3 に示す．class 属性値とレ

表 2　LANTERN を用いたときの各ラベルに対する評価尺度（評価方法 1）.

属性名	適合率	再現率	F_1 値	ROC-AUC
その他	0.96	0.71	0.80	0.75
日付	0.19	0.43	0.21	0.97
作者	0.03	0.29	0.05	0.81
質問タイトル	0.58	0.58	0.31	0.95
回答数	0.45	0.44	0.42	0.99
評価数	0.02	0.06	0.03	0.68

表 3　class 属性値とレイアウト情報を考慮し拡張したモデルを用いたときの各ラベルに対する評価尺度（評価方法 1）.

属性名	適合率	再現率	F_1 値	ROC-AUC
その他	0.98	0.98	0.98	0.73
日付	0.07	0.03	0.03	0.77
作者	0.06	0.02	0.03	0.74
質問タイトル	0.06	0.46	0.10	0.96
回答数	0.2	0.02	0.04	0.86
評価数	0.0	0.0	0.0	0.77

表 4　各ページにおいて最も高い値で予測した属性に対する適合率（評価方法 2）.

属性名	LANTERN	拡張モデル
日付	0.45	0.13
作者名	0.06	0.10
質問タイトル	0.27	0.22
回答数	0.77	0.10
評価数	0.04	0.29

イアウト情報を特徴量に追加することで,「作者」と「その他」の適合率が僅かに上昇したが，それ以外の属性は大きく減少した．また再現率も減少しているため F_1 値も減少している．これら評価尺度の値は小さいが，ROC-AUC は概ね 0.8 付近の値を取った．使用したデータは正例と負例に大きな偏りのある不均衡データなため適合率及び再現率の値がかなり低くなったが，ROC-AUC の値より，一定の分類能力持っていると考えられる．

4.5.2　評価方法 2 についての実験結果

各属性ごとの抽出結果を表 4 に示す．拡張したモデルの方が,「作者」と「評価数」の適合率が高いことがわかる．しかし，それ以外の属性は減少した．特に「回答数」の適合率が大きく減少している．特徴量として class 属性とレイアウト情報を追加することで，全体的に分類精度が悪化してしまっていることがわかる．

4.6　考　　察

DOM ノードの分類精度を向上させるために既存モデル LANTERN に HTML の class 属性とレイアウト情報としてフォントサイズ，ノードの縦横サイズ，ブラウザ上での描画面積を特徴量として追加し拡張した．しかし評価方法 1 の結果より，拡張モデルの分類精度は既存モデルを上回らなかった．この原因について考察する．うまく分類が行われなかった原因として，まずデータ中の正例と負例の割合に差があるためだと考える．DOM には抽出したいテキストを含むノード以外が大量に含まれているため，データの前処理として負例フィルタリングを行った．負例フィルタリングを行うことで，抽出したいテキストを含んだノード以外の負例を減らしたが，正例と負例の割合が同じ均衡データにすることができなかった．損失関数にクラスごとの重みを加えることで不均衡データを考慮した学習を行ったが，表 2，表 3 より負例に分類が偏ってしまっていることがわかる．しかし既存モデル LANTERN よりも拡張したモデルの方が適合率，再現率ともに減少していることから，データの不均衡性だけが低い分類精度となった原因ではないと考えられる．他の原因として，新たに特徴量を加えたことによる過学習が考えられる．類似したレイアウト情報を持つノードがその他属性に多く，うまく学習を行うことができなかった可能性がある．データの不均衡性に加え，類似した特徴を持つデータが多く存在したことが精度の低下につながったと考える．

評価方法 2 では表 4 より,「作者名」と「評価数」の項目で精度が向上した．特に評価数の項目で大きく上昇した．この理由について考察する．評価数属性の精度が向上した理由としては 2 点ある．1 つ目は，その他属性以外のノードにおいて，評価数のテキストを持つノードのレイアウト情報，縦横の幅が突出した特徴をしていたためだと考える．評価数を表すノードが持つテキストは数字のみであることが大半であった．そのためノードの高さ，横幅の大きさがほぼ等しく，Web ページ上での描画領域は正方形であった．同じ数を表す回答数を表すノードは，数字とともに “Anwers” などの単位を表した単語も含まれていることがあった．そのためノードの横幅が大きくなり，数字を表すノードであるが評価数とは異なるレイアウトをしている．他にも，質問タイトルや作者などには複数の単語や文字が含まれているため，自然とノードの横幅が大きくなる．このように他のノードと異なるレイアウト情報の特徴を捉えることができたため，評価数の属性は精度が向上したと考える．2 点目は，class 属性値による影響である．評価数を表すノードの class 属性値には，“like” や “vote” といった他者からの評価を表すような単語が含まれていることが多かった．しかし，それ以外の属性については class 属性値が存在しないなどがあった．class 属性が適切に含まれていたため精度が向上したと考えられる．

以上のことから，新たに特徴量として class 属性値とレイアウト情報を加えることは一部の属性では効果が見られた．しかしデータ内の正例と負例の偏りや，類似したレイアウト情報を持つノードが多いことから負例データの削減や，使用する特徴量の選択が必要であると考える．

5　まとめと今後の課題

本研究では，HTML 文書から Web ページに対する評価数や作成日，タイトルなどの Web ページ自体の属性を抽出するこ

とに取り組んだ．既存の属性抽出モデルに特徴量として HTML の class 属性値とレイアウト情報を追加し，モデルの拡張を行った．拡張したモデルの評価を行うために，プログラミング情報 Q&A サイトを用いて評価実験を行った．評価実験を行った結果，全体的に分類精度が低下してしまったが，一部の属性においては精度改善に繋がった．今後の課題としては，データ不均衡性への対処と使用する特徴量の選択が必要であると考える．また，抽出した属性を用いてサムネイルの生成にも取り組む．

謝辞　本研究は JSPS 科学研究費助成事業 JP21H03774, JP21H03775，による助成を受けたものです．ここに記して謝意を表します．

文　献

[1] Eytan Adar, Jaime Teevan, and Susan T Dumais. Large scale analysis of web revisitation patterns. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pp. 1197–1206, 2008.

[2] Andy Cockburn and Bruce McKenzie. What do web users do? an empirical analysis of web use. *International Journal of human-computer studies*, Vol. 54, No. 6, pp. 903–922, 2001.

[3] Susan Dziadosz and Raman Chandrasekar. Do thumbnail previews help users make better relevance decisions about web search results? In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 365–366, 2002.

[4] Tomas Gogar, Ondrej Hubacek, and Jan Sedivy. Deep neural networks for web page information extraction. In *Artificial Intelligence Applications and Innovations: 12th IFIP WG 12.5 International Conference and Workshops, AIAI 2016, Thessaloniki, Greece, September 16-18, 2016, Proceedings 12*, pp. 154–163. Springer, 2016.

[5] Qiang Hao, Rui Cai, Yanwei Pang, and Lei Zhang. From one tree to a forest: a unified solution for structured web data extraction. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pp. 775–784, 2011.

[6] Binxing Jiao, Linjun Yang, Jizheng Xu, and Feng Wu. Visual summarization of web pages. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pp. 499–506, 2010.

[7] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[8] Anurendra Kumar, Keval Morabia, Jingjin Wang, Kevin Chen-Chuan Chang, and Alexander Schwing. Cova: Context-aware visual attention for webpage information extraction. *arXiv preprint arXiv:2110.12320*, 2021.

[9] Bill Yuchen Lin, Ying Sheng, Nguyen Vo, and Sandeep Tata. Freedom: A transferable neural architecture for structured information extraction on web documents. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1092–1102, 2020.

[10] Shenwei Liu and Keishi Tajima. Wildthumb: a web browser supporting efficient task management on wide displays. In *Proceedings of the 15th international conference on Intelligent user interfaces*, pp. 159–168, 2010.

[11] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

[12] Jaime Teevan, Edward Cutrell, Danyel Fisher, Steven M Drucker, Gonzalo Ramos, Paul André, and Chang Hu. Visual snippets: summarizing web pages for search and revisitation. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pp. 2023–2032, 2009.

[13] Thijs Vogels, Octavian-Eugen Ganea, and Carsten Eickhoff. Web2text: Deep structured boilerplate removal. In *Advances in Information Retrieval: 40th European Conference on IR Research, ECIR 2018, Grenoble, France, March 26-29, 2018, Proceedings 40*, pp. 167–179. Springer, 2018.

[14] Allison Woodruff, Andrew Faulring, Ruth Rosenholtz, Julie Morrsion, and Peter Pirolli. Using thumbnails to search the web. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 198–205, 2001.

[15] Yichao Zhou, Ying Sheng, Nguyen Vo, Nick Edmonds, and Sandeep Tata. Learning transferable node representations for attribute extraction from web documents. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pp. 1479–1487, 2022.

[16] 稲垣有哉, 岩田一, 白銀純子, 深澤良彰. ウェブページの構造と顕著性マップの組み合わせによる重要領域の視覚化手法. Web インテリジェンスとインタラクション研究会 予稿集 第 14 回研究会, pp. 45–50. Web インテリジェンスとインタラクション研究会, 2019.

[17] 鶴田雅信, 増山繁. レイアウト情報を用いた web ページの主要な dom ノードの抽出法. 人工知能学会論文誌, Vol. 25, No. 6, pp. 742–756, 2010.

[18] 廖宸一, 廣井慧, 梶克彦, 河口信夫ほか. Html 構造解析と機械学習に基づくイベント情報抽出システムの提案. 研究報告ユビキタスコンピューティングシステム (UBI), Vol. 2015, No. 13, pp. 1–7, 2015.