構造と内容に基づくWebページからの評判抽出における パターンの構成法

長谷川博之† 工藤 峰一† 中村 篤祥†

† 北海道大学 大学院情報科学研究科 〒 060-0814 北海道札幌市北区北 14 条西 9 丁目
E-mail: †{hase,mine,atsu}@main.ist.hokudai.ac.jp

あらまし サーチエンジンを用いて収集される不特定サイトの Web ページから，与えられたキーワードに関連したテキストを抽出する問題を考える．我々は [1] においてキーワードと関連テキストとの関係を表現するパターンと，テキストの内容に基づく抽出方式を提案した．提案方式ではパターンの抽出においては頻出部分木マイニング手法を用い，テキスト分類技術と組み合わせてテキストの内容に基づいて抽出を行なう．本論文では，精度向上のためのパターンの構成法と，抽出結果の優先順位のつけ方について検討する．ラーメン屋の名前をキーワードとして収集された Web ページからその店の評判を抽出する実験では，新方式により適合率が約 10%up した．また，提案方式による優先順位において，誤って抽出されたテキストは下位にランク付けされた．

キーワード Web マイニング, 半構造データ，テキストマイニング ，Wrapper , パターンマッチング

Creation of Better Pattern Set for Reputation Extraction Using Both Structural and Content Information

Hiroyuki HASEGAWA†, Mineichi KUDO†, and Atsuyoshi NAKAMURA†

† Graduate school of Information Science and Technology Hokkaido University, Sapporo, 060-0814 Japan
E-mail: †{hase,mine,atsu}@main.ist.hokudai.ac.jp

Abstract We consider the problem of extracting texts related to a given keyword from Web pages collected by a search engine. We proposed a method using both structural and content information [1]. Our method is a combination of frequent tree mining and text classification technologies. In this paper, we propose a method of creating better pattern set to improve its precision and a ranking method of output results. According to our experimental results on extracting reputations of a given *Ramen* shop, the new method of pattern-set creation made the performance improve by about 10 %, and wrongly extracted texts were low-ranked by our ranking method.

Key words Web Mining， Semi-Structural Data，Text Mining，Wrapper , Pattern Matching

1. はじめに

近年，大量で多様な情報が Web ページという形で WWW 上に集積されるようになり，利用者は簡単に情報を得ることができるようになった．WWW 上にある情報の中で利用者にとって有用なものに，ある固有名詞（飲食店の名前，商品（製品）名）に関する評判がある．評判は自分の好みに合った飲食店を探すためや商品を購入するための判断材料として利用される．現在では飲食店や商品に関して評価を行なっている Web サイトが多くあるため，これらの評判は WWW を利用すると容易に発見することができる．しかし，サイトに自分の目的とする飲食店の評判がないことや，一つの Web ページに多数の飲食店に関する評判が書かれていて探すのが面倒なことがあり，判断材料として充分な量の評判を収集するのに多くの労力を必要とする．そのため，多数のページからある固有名詞に関する評判を自動的に収集できる方法が望まれる．

本論文では，Web ページから利用者によって与えれる固有名詞（以下，キーワード）に関する評判を抽出する問題を考える．

この問題を解決するための重要な技術として，Web ページから特定の箇所を自動的に抽出する Wrapper がある．実際に，特定サイトのページから評判を抽出する場合は，サイトの HTML の記述フォーマットが決まっていることが多いため LR Wrapper [2] や TreeWrapper [3] などの Wrapper を用いることでキーワードと評判の組を抽出できる．しかし，特定サイトのページだけでは判断材料として充分な量の評判が得られないことがある．特に，マイナーなキーワードに関する評判は個人のホームページ

や新しいサイトの情報まで含めないと充分な情報が得られない場合が多い．しかし，LR Wrapper や TreeWrapper の Wrapper の構築法では，抽出を行なう対象のページを検索エンジンを用いて収集した不特定のサイトのページに拡張できない．これらの手法は抽出すべき部分のまわりの構造に共通のパターンを求めるが，不特定のサイトのページには共通のパターンが存在しないためである．

この問題は，すべての訓練ページに共通な一つのパターンを求めるのではなく，どの訓練ページにも，それにマッチするようなパターンが存在するようなパターン集合を求める [4] ことによりある程度対処可能である．しかし抽出すべき部分とまわりの構造が全く同じでも抽出すべきでない部分もあり，まわりのパターンのみでは限界がある．まわりのみでなく抽出すべき部分に出現するパターンも考慮する方法 [4] も提案されているが，これは E メールアドレスや電話番号といった特殊な記述書式を持つテキストに有効であり，評判のような普通の文章で構成されているテキストには適用できない．

このように，不特定のサイトのページから情報を抽出するには，パターンマッチングのみでは不可能である．そこで，我々は HTML 構造のパターンマッチングとテキスト分類を組み合わせ，キーワードに関する評判を抽出する Wrapper の構築方法を提案した [1].

提案手法のパターンマッチングに用いるパターンは，与えられたキーワードと評判との HTML における構造的な関係を表現する．このパターンと，ページの HTML 構造とのパターンマッチングを行ない，抽出する候補を求める．テキスト分類は，パターンマッチングによって得られた候補を絞るために用いる．

提案手法で用いるパターンは HTML タグをラベルとして持つ順序木であり，二つの葉を持つ．二つの葉のうち一つはキーワードを表し，もう一つは評判を表している．このパターンは TreeWrapper [3] や $L_{tagpath}$ [4] に用いられるパターンと似ているが，柔軟性のあるマッチング方式 [5] を用いているところが異なる．つまり，パターンで親子関係にあると表現された二つノードは，マッチングの際には親子関係だけではなく先祖と子孫の関係であってもマッチする．これにより一つのパターンで多くの抽出すべき部分とマッチする．しかし，評判ではない部分とのマッチも多くなってしまう．そこで，パターンにマッチした箇所のテキストを，テキスト分類技術を用いて ”評判” と ”評判以外” に分類する．テキスト分類にはいかなる分類器 [6] も利用できるが，提案手法では良い性能を持つといわれている support vector machines（SVM）[7] [8] を用いた．

論文 [1] の手法では，抽出結果のうち約 $\frac{1}{5}$ は誤りであった．これらの誤りが利用者を誤った判断に導いてしまったり，気になる評判を実際のページで確認しなければ信用できないなど，利用者に余計な負担をかける可能性がある．

本論文では，[1] で提案した Wrapper の構築法において適合率の向上を目的としたパターン集合の構成方法を提案する．集合に含まれる個々のパターンの教師データに対する適合率を求め，適合率の低いパターンを，より特殊なパターンと置き換えることで適合率の向上を図る．また，抽出結果の信頼性の目安として Wrapper の抽出結果に優先度を付け，優先度の高い結果から順に表示する方法について検討する．

表 1　本論文で用いる記法
Table 1　Notations used in this paper.

T.root	木 T の根ノード.
N_T.tag	ノード N_T の tag 属性の値.
N_T.text	ノード N_T の text 属性の値.
N_T.id	ノード N_T の id.
lca($N_{T,1}, N_{T,2}$)	ノード $N_{T,1}$ と $N_{T,2}$ の LCA を返す関数
contain(s_1,s_2)	文字列 s_2 が文字列 s_1 に部分文字列として．含まれるとき真を返すブール関数.
first($\mathcal{Q}$)	リスト $\mathcal{Q}$ の先頭の要素を求める関数.
del_first($\mathcal{Q}$)	リスト $\mathcal{Q}$ の先頭の要素を削除する関数.

本手法の有効性を検証するために行なった実験では，クロスバリデーションを行った結果，新方式により適合率が約 10%up した．また，提案方式による優先順位において，誤って抽出されたテキストは下位にランク付けされた．

1.1 関連研究

立石ら [9] は不特定のサイトのページから評判を抽出する評判検索エンジンを開発した．彼らの検索エンジンでは評判の抽出にタグや構造など HTML 特有の特徴は用いず，自然言語処理技術を用いている．教師データの作成の他に，抽出規則に用いる評価表現辞書も人手で作成する必要があるため，本手法よりも手間がかかるものと考えられる．

山田らの手法 [10] の抽出方法では本手法のように HTML の木の構造とテキストの持つ特徴を利用する．しかし，利用するテキストの特徴は LR-Wrapper が使用しているようなテンプレートとなる文字列であり，これは木のノードが持つテキストの一部を抽出するために用いている．

また，SVM を利用する Wrapper として Kashima らの手法 [11] がある．彼らの手法では本手法のように SVM をテキスト分類器として用いるのではなく，カーネルを利用して HTML の木の分類に用いている．

2. 問題設定

2.1 本論文で用いる記法

順序木 $T = (V, B, \preceq)$ は，T に含まれるノード N_T の集合 V と親子関係を表す枝 $(N_{T,i}, N_{T,j}) \in V \times V$ の集合 B から構成され，V の一部に兄弟順序 $\preceq$ を持つ木である．順序 $\preceq$ は同じ親ノードを持つすべてのノード間に定義されており，グラフとして図示するときは左から右の順に書く．また，木のなぞりを行なうときも左から行なうものとする．

DOM 木は HTML テキストを DOM（document object model）の概念に基づく順序木に変換したものである．DOM 木のノードは tag と $text$ という二つの属性を持ち，tag の値は ”body” や ”table” といった HTML のタグであり，text の値は文字列とする．text の値はノードの tag の値が ”#text”(注1)であるとき

(注1)：HTML には ”#text” という tag は存在しないが，本論文では text ノードはこの tag の値を持つと仮定する．

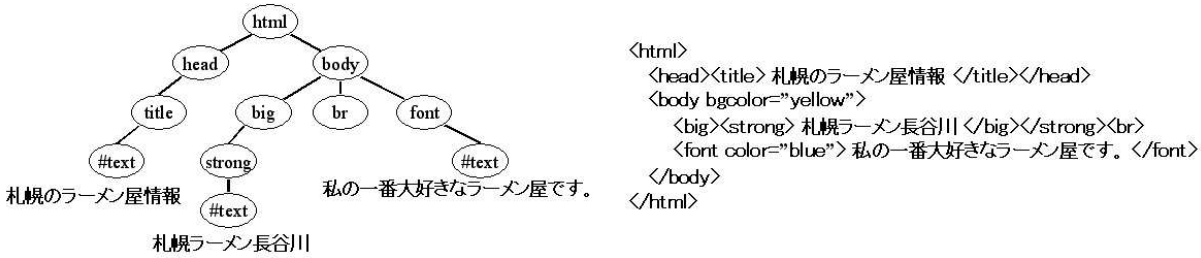

図 1　DOM 木と DOM 木に変換する前の HTML テキストの例.

Fig. 1　The example of a DOM-tree and HTML text which is original.

のみ持つものとする．以下ではこのノードのことをテキストノードと呼ぶことにする．DOM 木と DOM 木に変換する前の HTML テキストの例を図 1 に示す．図 1 で示すように，DOM 木のノードの属性として前述した二つの属性以外は利用しない(注2)．

ノード N_T の id は T に対して深さ優先のなぞりを行なったときの前順における N_T の出力順番である．

本論文で用いる記法を表 1 に示す.

2.2　本論文で扱う抽出問題

キーワードを W，text 属性値に部分文字列として W を含むノード（キーワードノード）を持つ DOM 木を T，W と関連する抽出対象のノード（ターゲットノード）を N_T^* とし，教師データはこれら三つの組 (W, T, N_T^*) から構成されるとする．このとき，本論文では以下の問題を扱う.

P_{ROBLEM} 2.2.　**与えられた教師データの集合 $\mathcal{D}$ から，N_T^* を出力する関数 $f(W,T)$ を学習せよ.**

N_T^* は，T に含まれるノードのうち W に関する評判を text 属性値として持つすべてのテキストノードの LCA（Least Common Ancestor）とする．T に N_T^* として適当なノードがない，すなわち T に W の評判がないときは N_T^* を $null$ とする.

3.　構造と内容に基づく抽出法

この節では，論文 [1] において我々が提案した構造と内容に基づく抽出法について説明する.

3.1　予測関数 f

関数 f は三つの関数 f_{pat}，f_{con}，f_{dec} によって構成する．それぞれの関数は以下に示す役割持つ.

パターンマッチング関数 f_{pat} ： DOM 木 T から W に関するターゲットノードの候補を，パターンマッチングを用いて予測する関数である．W と T の関数であり，候補ノードの集合 $\{N_{T,1}, N_{T,2}, ..., N_{T,k}\}$ を出力する.

コンテンツベース関数 f_{con} ： f_{pat} が出力したノードを，テキスト分類技術を用いて ”評判” と ”評判以外” のクラスに分類する．ノード N_T の関数であり，出力はノード N_T の text 属性値がテキスト分類法によって ”評判” のクラスに分類されるとき 1，”評判以外” のクラスに分類されるとき -1 となる.

決定関数 f_{dec} ： 集合 $\{(N_{T,1}, l_1), (N_{T,2}, l_2), ..., (N_{T,k}, l_k)\}$ の関数である．ここでは $l_j = 1$ である組のうち，$N_{T,j}$.id が最小のノードを出力する関数を採用する．集合に含まれる組のラベルがすべて -1 であるときは何も出力しない.

この三つの関数を用いて，f を

$$f(W,T) = f_{dec}(\{(N_T, f_{con}(N_T)) : N_T \in f_{pat}(W,T)\})$$

と定義する．この f を教師データ (W, T, N_T^*) の集合 $\mathcal{D}$ から求める.

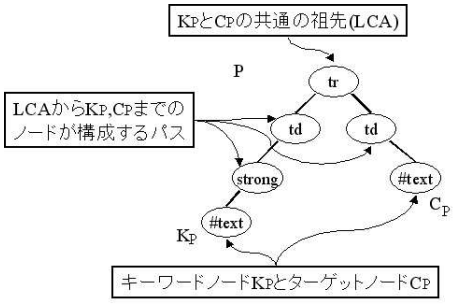

図 2　DOM 木パターン P の例

Fig. 2　Example of pattern of DOM-tree P.

3.2　パターンマッチング関数

f_{pat} は DOM 木に対してパターンマッチングを行ない，ターゲットノードの候補を求める．f_{pat} がパターンマッチングに用いるパターンは，DOM 木においてキーワードと評判のノードの間にどのような構造的な関係があるかを表現する．本手法ではこのパターンを，二つの葉ノードを持つ DOM 木のパターン P と，二つの葉ノードの id の差 r の組 (P, r) とする．P は根と二つの葉，根から葉までのノードのパスから構成される．葉の一方を K_P，もう一方をターゲットノード C_P で表す．K_P の tag 属性の値は ”#text” に限定する．またほとんどの場合，評判よりも店名が先に出現することから K_P.id< C_P.id を満たす木のみをパターンとして考える(注3)．DOM 木パターン P の例を図 2 に示す.

DOM 木 T において，パターン (P, r) に対し以下に示す七つの条件を満たす写像 ϕ が存在するとき，N_T と W との関係は (P, r) とマッチすると言い，このときの N_T がターゲットノードの候補となる．ただし，P のノードから T のノードへの写像 ϕ において，以下の条件のうち $N_{P,1}$ と $N_{P,2}$ に関する条件は P のすべてのノードが満たしていなければならないものとする.

(注2)：TreeWrapper [3] では ”bgcolor” などの属性も利用している.

(注3)：この制約を取り除いてもアルゴリズムを少し変更することにより対応可能である.

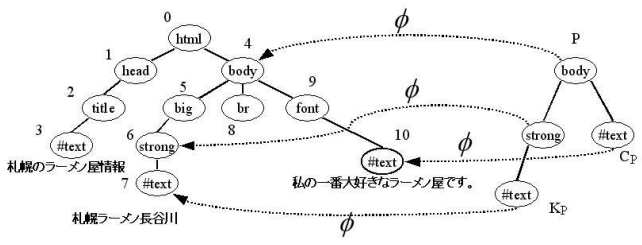

図 3　パターン $(P, 3)$ が DOM 木 T とマッチするときの例.
Fig. 3　Example of pattern $(P, 3)$ matches a DOM-tree T.

1　ターゲットノードがマッチする．すなわち $N_T = \phi(C_P)$ が成り立つ.
2　ノードの tag 属性の値が等しい．すなわち $N_{P,1}$.tag $= \phi(N_{P,1})$.tag が成り立つ.
3　$\phi(K_P)$ の text 属性値が W を含む．すなわち contain($\phi(K_P).text, W$) が真である.
4　パターン P においてノード $N_{P,1}$ が $N_{P,2}$ の子であるならば $\phi(N_{P,1})$ は $\phi(N_{P,2})$ の子孫である.
5　キーワードノードがターゲットノードよりも先に出現する．すなわち $\phi(K_P)$.id $< \phi(C_P)$.id が成り立つ.
6　キーワードノードとターゲットノードの id の差が r 以下である．すなわち $|\ \phi(K_P)$.id$-\phi(C_P)$.id$| \leqq r$ が成り立つ.
7　LCA がマッチする．すなわち lca($\phi(K_P), \phi(C_P)$)$= \phi(P.root)$ が成り立つ.

パターン (P, r) が DOM 木 T とマッチするときの例を図 3 に示す．この例では $r = 3$ とする．キーワードが「札幌ラーメン長谷川」であるとすると，id が 10 のテキストノードは $(P, 3)$ によってターゲットノードの候補になる.

このようなパターン (P, r) の集合 $\mathcal{P}$ を教師データの集合 $\mathcal{D}$ から求める．f_{pat} の出力は $\mathcal{P}$ に属するパターンのどれかにマッチするノードすべての集合とする.

3.2.1　パターン集合 $\mathcal{P}$ の算出

パターンの集合 $\mathcal{P}$ は，教師データ (W, T, N_T^*) の集合 $\mathcal{D}$ において頻出するパターン (P, r) の集合である.

本手法では $\mathcal{P}$ を以下の手順に従い構成する.

手順 1.　$(W, T, N_T^*) \in \mathcal{D}$ において，キーワード W を含むノード N_T^W とターゲットノード N_T^* を含む最小の連結部分グラフ S を求める．すべての教師データから求めた S の集合を $\mathcal{S}$ とする.
手順 2.　$S \in \mathcal{S}$ を，S.root.tag を l，N_S^*.tag(N_S^* は N_T^* に対応する S のノード．) を t とし，(l, t) の組み合わせで分類する．この集合を $\mathcal{E}_{l,t} = \{S \in \mathcal{S} : (S.root.tag, N_S^*.tag) = (l, t)\}$ とする.
手順 3.　各々の $\mathcal{E}_{l,t}$ から共通極大埋め込み部分木 [5](P, r) の集合 $\mathcal{P}_{l,t}$ を求める．(l, t) のすべての組み合わせを $\mathcal{G}$ とすると，パターン集合 $\mathcal{P}$ は $\mathcal{P} = \bigcup_{(l,t)\in\mathcal{G}} \mathcal{P}_{l,t}$ として求まる.

手順 3 では，Agrawal らによって提案された *AprioriAll* アルゴリズム [12] を用いる．AprioriAll アルゴリズムはシーケンシャルパターンのマイニングに用いられる．本手法では，S の根から葉までを辿るノードのパスを tag のシーケンスとして考え，AprioriAll アルゴリズムを用いて頻出埋め込み部分木を求める.

3.3　コンテンツベース関数

f_{con} は与えられたノードをその text 属性値に基づき ”評判” または ”評判以外” のクラスに分類する．本手法ではテキスト分類器として SVM を用い，分類器の特徴ベクトルとして索引語ベクトル [6] を用いる．索引語とはテキストの内容を特徴付けるうえで重要な単語のことを呼び，この索引語の重みを要素とするベクトルでテキストを表現する．重みは二進重みや索引語頻度，TF-IDF 重みなどがよく用いられている.

テキスト分類器の構築のための教師データは，パターン集合 $\mathcal{P}$ を構成するときにも用いた $\mathcal{D}$ と f_{pat} から作成する．テキスト分類器の正例は $\mathcal{D}$ に含まれる N_T^* の text 属性値とする．負例は $(W, T, N_T^*) \in \mathcal{D}$ のすべてに対して関数 f_{pat} を実行し，その出力のうち N_T.id$\leqq N_T^*$.id を満たす(注4)ノードの text 属性値とする.

3.4　課　　題

論文 [1] における提案手法における Wrapper の課題の一つとして，抽出結果の適合率の向上が挙げられる.

論文 [1] の手法では，抽出結果のうち約 $\frac{1}{5}$ が誤りであった．誤った結果がキーワードではない他の店の評判であるとき，その情報が判断材料として利用者にとって重要である場合，利用者を誤った判断へ導く可能性がある．また，抽出結果の信頼性が低いとなれば，利用者は気になる情報に関して実際のページを見て確かめたりすることになり，余計な手間が生ずることになる.

4.　パターンの構成法

集合 $\mathcal{P}$ に含まれるパターンには，コンテンツベース関数と組み合わせて適合率の高いパターンと低いパターンがある．本節では，抽出結果全体の適合率の向上を目的として，集合 $\mathcal{P}$ を教師データに対して適合率の高いパターンだけから構成する方法（PS アルゴリズム）について述べる.

パターン $(P, r) \in \mathcal{P}$ の教師データ $\mathcal{D}$ に対する適合率を

$$Prec_{(P_i,r_i),\mathcal{D}} = \frac{positive((P, r), \mathcal{D})}{output((P, r), \mathcal{D})}$$

と定義する．ここで，$output((P, r), \mathcal{D})$ は D に含まれる木 T においてパターン (P, r) にマッチし，f が抽出したノードの数であり，$positive((P, r), \mathcal{D})$ はそのうちの正解数である．この値は $\mathcal{D}$ に対し実際に構築した Wrapper を適用することで求まる．閾値 $\alpha(0 \leqq \alpha \leqq 1)$ を設定し，あるパターン (P, r) の $\mathcal{D}$ に対する適合率が α 未満であるとき，(P, r) をより特殊なパター

(注4)：一つのデータに複数の評判が記述されているとき，この条件を除くとキーワード以外に関する評判が負例となることがある．分類器を用いる目的は” 評判” と” 評判以外” のテキストに分類することであるため，分類器の精度を上げるためにこの条件を付加する.

[PS アルゴリズム]

STEP0. パターンリストの生成　飽和パターン [13] 集合 $\mathcal{P}_{l,t}$ を求める．$\mathcal{P}_{l,t}$ に含まれるパターンをサポートの高い順に格納したリストを $\mathcal{Q}_{l,t}$ とする．(l,t) のすべての組み合わせの集合を $\mathcal{G}$，適合率の閾値を α とする．

STEP1. パターン集合の初期化　パターン集合 $\mathcal{P}$ を，

$$\mathcal{P} = \{first(\mathcal{Q}_{l,t}) \ : \ (l,t) \in \mathcal{G}\}$$

とする．

STEP2. Wrapper の構築　f_{pat} に用いるパターン集合を $\mathcal{P}$ として，Wrapper を構築する．

STEP3. パターンの適合率を算出　訓練集合 $\mathcal{D}$ に対して 2 で構築した Wrapper を用いて抽出を行ない，$(P,r) \in \mathcal{P}$ の $\mathcal{D}$ に対する適合率 $Prec_{(P,r),\mathcal{D}}$ を求める．

STEP4. 最も適合率の低いパターンを選択　$\mathcal{P}$ に含まれるパターンのうち，最も適合率 $Prec_{(P,r),\mathcal{D}}$ の低いパターン

$$(P^*, r^*) = \arg \min_{(P,r)\in\mathcal{P}} Prec_{(P,r),\mathcal{D}}$$

を求める．

STEP5. パターンの置換　$Prec_{(P^*,r^*),\mathcal{D}} < \alpha$ ならば $(P^*,r^*) \in \mathcal{P}_{l,t}$ を満たす $(l,t) \in \mathcal{G}$ に対して，

$$del\_first(\mathcal{Q}_{l,t})$$

を行ない，STEP1 へ戻る．$Prec_{(P^*,r^*),\mathcal{D}} \geqq \alpha$ ならば終了する．

図 4　集合 $\mathcal{P}$ を構成する PS(Pattern Specification) アルゴリズム．
Fig. 4　Composition algorithm of pattern set $\mathcal{P}$.

ンと置き換える方法を提案する．

教師データ D に対して最も適合率が低くかつ α 未満であるパターン (P^*, r^*) を，LCA およびターゲットノードのタグが同じパターンの中で (P^*, r^*) の次にサポートが高いパターンと置換する．この操作を適合率が α 以下であるパターンがなくなるまで繰り返す．

PS アルゴリズムの詳細を図 4 に示す．

パターンのサポートは，教師データの集合 $\mathcal{D}$ のうちパターンがターゲットノード N_T^* を候補として挙げることができるデータの数である．

飽和パターンは，パターン (P,r) の P において，P を構成するノードを一つでも加えると，(P,r) のサポートが小さくなるようなパターンである．

5. 抽出結果の優先度

Wrapper の抽出結果として，予測関数 f により出力されたノードを根とする部分木に含まれる text 属性値を列挙したものを考える．この抽出結果に優先度を付け，優先度の高い結果から順に表示する方法について提案する．優先度に抽出部分の信頼度をうまく反映させることができれば，利用者はその値も判断基準に使うことができ，判断を誤る可能性が減ると考えられる．

本手法では，優先度を構成するものとして二つの基準を用いる．一つは，抽出結果の f_{con} における SVM の値である．f_{con} で分類器として用いた SVM は特徴ベクトルを入力すると実数値（識別境界からのマージン）を出力する．ノードに付加するラベルは分類器 SVM の出力を m としたとき，$sign(m)$ の値である(注5)．本手法では，優先度の基準としてこの SVM の出力である実数値 m を用いる．この値が大きいノードに対する text 属性値は，小さいものよりも識別器によってより評判らしいとされたテキストである．もう一つの基準は，木 T におけるキーワードノード N_T^W と抽出したノード N_T の id の差 $|\ N_T^W$.id$-N_T$.id$|$ である．これは，木 T において N_T^W と N_T がどれだけ離れているかを示す．この値が小さいノードは木においてキーワードの近くにあるため，キーワード以外の評判である可能性よりもキーワードに関する評判である可能性が高い．

これら二つの値を組み合わせた優先度を考える．Wrapper の抽出結果のノードの集合を $\mathcal{O}$ とし，$o \in \mathcal{O}$ の SVM の出力値を m_o，id の差を r_o とする．ただし $m_o \geqq 1$ の場合は $m_o = 1$ とする．$r_{max} = \max_{o\in\mathcal{O}} r_o$ とし，o の優先度を

$$Priority(o) = m_o + (1 - \frac{r_o}{r_{max}})$$

と定義する．このように定義した優先度を Wrapper のすべての抽出結果に対して求め，優先度の高い結果から順に表示する．

表 2　実験に使用したデータの詳細
Table 2　Details of the data used for the experiment.

店番号	総データ数	評判の数	#ONE	#MANY
1	34	26	8	18
2	19	9	2	7
3	73	46	15	31
4	24	14	4	10
5	26	12	4	8
6	23	17	7	10
7	21	11	6	5
8	33	21	6	15
9	20	13	5	8
10	28	28	5	15
合計	301	189	62	127

6. 評価実験

6.1 実験条件

本手法の有効性を検証するために実験を行なった．使用するデータはラーメン屋に関する情報が記述された HTML テキストである．地方タウン情報誌(注6)に記載されている人気ラーメン店ランキングトップ 100 に含まれる店について，検索エンジン Google(注7)を用いて「店の名前」と「電話番号」で AND 検索した．そして，検索結果のページ数が 15 以上であるときそのページを実験データとし，これをランキング上位から 10 店分収集した．データは合計 301 ページ，このうち評判が書かれているページは 189 である．Wrapper に入力するキーワードはラーメン屋の名前であり，抽出するテキストはその店の評判

(注5)：$sign$ 関数は引数の値が正ならば 1，負ならば -1 を返す．
(注6)：Hokkaido Walker 2002 年 No.3
(注7)：http://www.google.co.jp/

表 3　実 験 結 果
Table 3　Result of experiment.

閾値 α	教師データ		テストデータ				
	平均適合率	平均再現率	f_{pat}	f_{con}	f_{dec}	平均適合率	平均再現率
比較手法	0.943	0.938	166	95	85	0.630 (0.793)	0.450 (0.566)
α=1.0	1.0	0.631	92	63	58	0.707 (0.917)	0.306 (0.391)
α=0.9	0.980	0.730	112	77	71	0.717 (0.929)	0.375 (0.486)

である．表 2 に実験に使用したデータの詳細を示す．表 2 の #ONE は店に関する評判が一つしか書かれていないデータの数であり，#MANY は多数の店に関する評判が書かれているデータの数である．データを店ごとに分割し，クロスバリデーションにより適合率と再現率を求めた．

テキスト分類法に利用した索引語は名詞，動詞，形容詞，副詞の四つである．また，索引語の切り出しには形態素解析エンジン茶筌(注8)を用いた．索引語ベクトルの要素の重みには索引語の頻度を用い，ベクトルの正規化にはユークリッドノルムを用いた．分類器の SVM として SVMTorchII(注9)を用い，カーネルとして三次元の多項式カーネル $K(x,y)=(xy+1)^3$ を用いた．

f_{pat} に用いるパターンの集合 $\mathcal{P}$ は，最小サポートを 0.1 としたときの飽和パターンを用いた．PS アルゴリズムに用いる適合率の閾値 α は 1.0 と 0.9 とし，各値ごとに Wrapper を構築してその精度を計算した．また，PS アルゴリズムを用いることによる精度の変化を確認するために，論文 [1] の手法と比較した．

6.2　実 験 結 果

実験結果を表 3 に示す．テストデータに対する適合率と再現率の列で括弧の付いていない値は，教師データ (W,T,N_T^*) としてラベルを付けたノードのテキストだけを正解として求めた．括弧内は，上記の精度の求め方では不正解とされたテキストに関し，以下に示す基準のいずれかを満たしていれば正解とした値である．

[基準 A]

1. 抽出すべき部分の一部である．

2. 抽出すべき部分を含み，キーワード以外の情報を含まない．

3. 抽出すべき部分を含んでいないが，キーワードに関する情報であり，評判とみることもできる．

閾値 α の値が 1.0 および 0.9 のときの結果は，PS アルゴリズムを適用しない方法の結果と比較して適合率が 7〜8（12〜13）% ほど向上している．

表 3 の f_{pat}，f_{dec} の値は，それぞれの関数が出力したノードに含まれる正解のノードの数であり，f_{con} の値は，ラベル 1 が出力されたノードの内の正解数である．

$\alpha=0.9$ のときの出力に優先順位を付けた場合の不正解の順位に関する結果を表 4 に示す．この表においては，基準 A を満たさないものを不正解としている．また，提案した優先度を用いる方法の他に，優先度として SVM の出力値の基準 m_o のみ用いる方法と，ノードの id の差の基準 $(1-\frac{r_o}{r_{max}})$ のみ用いる方法の結果も示す．提案した優先度を用いる方法では全体的に誤って抽出されたノードの順位は下位になっている．

Wrapper の出力結果の例を表 5 に示す．これは α=0.9 のときに店番号 10 についての評判を抽出した結果であり，優先順位の上位から順に記述してある．

6.3　考　　察

表 6 に，教師データの集合 $\mathcal{D}$ に対して適合率が 1.0 未満の初期パターンと各パターンの教師データ $\mathcal{D}$ に対する適合率および

表 4　α=0.9 のときの Wrapper の出力に優先順位を付けた結果
Table 4　Result of putting order of priority on output of Wrapper at α=0.9.

店番号	出力結果数	不正解の結果の最高順位			不正解数
		$m_o+(1-\frac{r_o}{r_{max}})$	m_o	$1-\frac{r_o}{r_{max}}$	
1	15	-	-	-	0
2	6	-	-	-	0
3	25	16	18	1	2
4	9	-	-	-	0
5	7	6	5	7	1
6	7	7	3	7	1
7	4	-	-	-	0
8	13	11	12	10	2
9	6	-	-	-	0
10	7	7	6	5	1

表 5　Wrapper の出力結果の例．○は正解，△は基準 A を満たすもの，×は不正解．
Table 5　Example of output of Wrapper. ○ is correct, △ is the one to fulfill the standard A and × is mistake.

Priority=1.627 : ○ : 某製麺会社に働いていたという店主こだわりの麺は固めでモチモチしています．一番人気は辛味噌ラーメンだそうですが，辛いのが苦手な私はパスして，味噌ラーメンを頼みましたがこちらも大変おいしかったです．しょうがとニンニクがきいていて，ゴマがかなり多く入っていました．もやしなどの野菜もおいしかったのですが，残念なのはチャーシューがちょっと固いかも … のれんが白地だと（主に昼間）店主，青地だと（夜の部）息子さんがラーメンを作っています．

Priority=1.002 : ○ : まだラーメンなどが雑誌に特集されていない頃，並ぶ店で有名に成ったことがある．辛みそラーメンが美味しいね．

Priority=0.938 : ○ : 職人技がきらりと光る寡黙 (かもく) なマスターが作るこだわりのラーメン…．４日間くらいじっくりと寝かせたタレの旨味とコクがはっきりとわかるラーメンだぞ．また，ゲンコツや豚足のなかにある骨髄から本来の旨さを引き出し，煮立て過ぎない澄んだスープにもこだわっている．仕込みで一度に使う骨ガラの量が何と 30 キロというのには驚きだ．辛味噌なのに単に辛いだけのラーメンとは違い，スープの旨味が分かるだけのじわっとした辛さに押さえるのが秘訣とか．ここまでスープにこだわっているからには，当然ながら麺や具にもこだわりを持っている．麺の場合，生の麺を発酵させない微妙な程度に寝かせてコシを作り，茹でた後でも心地良い食感を持たせているんだ．具にしても，チャーシューには厳選された豚のモモ肉を使い，ラーメンを出すときに濃い口醤油ダレをかけ，注文が来てから初めて味付けを整えているんだ．たかが，ラーメン．されど，ラーメン…．しかし，こうした面倒なほどのこだわりが，「本当に旨い札幌ラーメンを食べて貰いたい」という元製麺会社に勤めていたマスターだからこその寡黙な技なんだな．

Priority=0.736 : △ : A パパ：みそ，しお，正油ともにもやし野菜が入っていてうまい！しおがうまい．味の濃さに好き好きがあるが，A パパは塩ラーメンでは，(トン骨系を除く) サッポロで一番だと思ってしまった．

Priority=0.651 : ○ : ランキング１２位は，これもまた白石区代表！ロングセラー的な有名店．武蔵です．武蔵といえば辛みそラーメンです．ピリッと，くる味が皆さんに大きな印象を与えているのでしょうか？ラーファンの間では，かなり支持されているようです．順位は落ちたものの，２００１年に続き２年連続ランクインは実力の高さの証デス！

Priority=0.497 : △ : 人気の秘訣は，昔ながらのサッポロラーメン作りへのこだわりと，麺に対する職人魂がそのまま味に伝わる事にある．

Priority=0.465 : × : Hokkaido Walker に「北海道産の小麦で作るコシのあるモチモチとした食感の麺はもちろん，やわらかいチャーシューやもとダレもすべて手作り，独特の味わいをとことん追求するこだわりを持ち，２２時間かけてじっくり抽出するトンコツスープのうまさは絶品だ！」３味とも５５０円

（注8）：茶筌 2.3.3.（URL:chasen.aist-nara.ac.jp/hiki/ChaSen）

（注9）：http://www.idiap.ch/ bengio/projects/SVMTorch.html

表 6　PS アルゴリズムを適用する前のパターン (P, r) と $\mathcal{D}$ に対する適合率の例（一部を抜粋. 実際は 35 パターン程度.）

Table 6　Example of patterns before construction method of pattern is applied and its precision to $\mathcal{D}$.(Only part was excerpted. There are actually about 35 patterns.)

番号	サポート	r	P	適合率
1	5	270	div #text(1) −1 #text(2) −1	0.833
2	15	256	td #text(1) −1 #text(2) −1	0.888
3	1	2	tbody tr td #text(1) −1 −1 −1 tr(2) −1	0.5
4	9	173	tbody tr td #text(1) −1 −1 −1 tr td(2) −1 −1	0.875
5	12	56	table tr td #text(1) −1 −1 −1 tr td(2) −1 −1	0.846
6	19	452	table tr #text(1) −1 −1 tr td #text(2) −1 −1 −1	0.923

表 7　表 6 のパターンを閾値 α=1.0 とした PS アルゴリズムを用いて置換した結果.

Table 7　Result of having substituted pattern of Table 6 at threshold α=1.0.

番号	サポート	r	P	適合率
1	2	270	div table tr td tr #text(1) −1 −1 −1 −1 −1 table tr td #text(2) −1 −1 −1 −1	1.0
2	2	47	td strong #text(1) −1 −1 table tbody tr td #text(2) −1 −1 −1 −1 −1	1.0
3	-	-	なし	-
4	4	7	tbody tr td strong #text(1) −1 −1 −1 −1 tr td(2) −1 −1	1.0
5	3	56	table tr td strong #text(1) −1 −1 −1 −1 tr td table tr td(2) −1 −1 −1 −1 −1	1.0
6	4	193	table tr td table #text(1) −1 −1 −1 −1 tr td #text(2) −1 −1 −1	1.0

サポートを示す．パターンは深さ優先の木のなぞりで前順出力を行なった場合のタグ列で表現されている．ただし，親ノードに移動するたびに ”−1”を出力するものとする．この表からわかるように，いくつかのパターンは $\mathcal{D}$ に対する適合率が低い．論文 [1] の手法では，このようなパターンも利用して Wrapper を構築する．表 7 は本稿で提案した PS アルゴリズムにおいて閾値 $\alpha = 1.0$ とし，表 6 のパターンに対して PS アルゴリズムを適用した結果である．このように，$\mathcal{D}$ に対する適合率が低いパターンが，適合率が 1.0 であるパターンと置換されている．

パターン集合 $\mathcal{P}$ の構成に PS アルゴリズムを用いた Wrapper では PS アルゴリズムを用いない比較手法と比べて，教師データだけでなく，テストデータに対する適合率も向上している．したがって，本手法のパターンの構成方式は Wrapper の抽出結果に含まれる誤りを減らすのに有効であることがわかる．

PS アルゴリズムを適用した結果，比較手法の結果と比べて再現率が低下しているのは，パターンの選択において適合率の低いパターンを適合率が高くかつサポートが低いパターンと置き換えているためである．

Wrapper の抽出結果に本手法で提案した優先度を用いて順位を付けた結果では不正解が下位にランク付けられており，上位から数えて不正解が含まれない割合の上限は，抽出結果全体の 60% であった．優先度として m_o のみ利用したときのこの割合は約 30% であり，$(1 - \frac{r_o}{r_{max}})$ のみ利用したときは不正解が一位となることがあった．したがって，これらの値を組み合わせることで優先度の信頼性が増したといえる．特に店番号 6，10 の結果では誤りの情報が最下位となっており，提案した優先度はある程度有効に働いているといえる．

7. おわりに

木のパターンマッチングとテキスト分類を用いて Web ページから固有名詞に関連した評判を抽出する Wrapper の構築法に関し，適合率の向上を目的としたパターン集合の構成方法と抽出結果の優先度付けの方法について提案した．提案法の有効性を検証する実験では，提案したパターンの集合の構成方式を用いることで適合率が約 10% 向上し，優先度を付加することで誤りの情報の順位が下位ランク付けされるという結果が得られた．

本論文では抽出するテキストをキーワードに関する評判と限定した．しかし，なんらかの分類器を用いて ”抽出する情報” と ”それ以外” のクラスに分類できれば，評判ではなくても本手法を用いて Wrapper を構築できる可能性はある．例えば，E メールアドレスや電話番号，住所等がこれにあたる．また，本手法の抽出精度向上のための拡張として，パターンの表現方法として HTML テキストの繰り返し構造を利用する [14] ことも検討すべき事項である．

抽出した評判から店のランキングを自動で行なったり，評判の要約を作成するなど，抽出結果の効果的な加工法についても今後検討していきたい．

本論文で提案した方法を用いた WWW 評判検索システムの構築を現在検討中である．WWW 上に蓄積される情報が増えるなかで，利用者の目的に特化することで情報を効率よく収集できる検索システムの需要は今後増えると予想される．

謝　辞

北海道大学の有村 博紀先生には本論文を書くにあたり有益なコメントをいただきました．ここに感謝いたします．

文　献

[1] H. Hasegawa, M. Kudo and A. Nakamura, Reputation Extraction Using Both Structural and Content Information, Hokkaido university TCS Technical Report TCS-TR-A-05-2(http://www-alg.ist.hokudai.ac.jp/tra.html), 2005.

[2] N. Kushmerick，Wrapper Induction : Efficiency and expressiveness, Artificial Intelligence，118，pp.15 − 68，2000.

[3] 村上 義継，谷口 力昭，坂本 比呂志, 有村 博紀, 有川 節夫, “HTML からのテキストの自動切り出しアルゴリズムと実装, ” 情報処理学会論文誌：数理モデル化と応用, vol. 42, no. SIG14-006, pp.39-49, 2001.

[4] W. W. Cohen, M. Hurst, and L. S. Jensen, A flexible learning system

for wrapping tables and lists in html documents, Proc. 11th Int＇l World Wide Web Conf., pp.232-241, 2002.

[5] M. J. Zaki，Efficiently mining frequent trees in a forest, Proc. SIGKDD＇02, pp.71-80, 2002.

[6] R. Baeza-Yates and B. Ribriro-Neto，Modern Information Retrieval, ACM Press，New York, NY, 1999.

[7] V. N. Vapnik, The Nature of Statistical Learning Theory, Springer, New York, NY, 1995.

[8] T. Joachims，Text categorization with support vector machines: Learning with many relevant features, Proc. 10th European Conference on Machine Learning, pp.137-142, 1998.

[9] K. Tateishi, Y. Ishiguro, and T. Fukushima, A Reputation Search Engine that Collects People's Opinions by Information Extraction Technology, The Information Processing Society of Japan (IPSJ) Transactions on Databases (TOD), vol. 45, no. SIG 07, 2004.

[10] 山田 泰寛, 池田 大輔, 廣川 佐千男, 半構造化文書に対する木構造と文字列を組み合わせたラッパーの自動生成法, 情報処理学会研究報告, vol. 2003, no. 98, pp.115-122, 2003.

[11] H. Kashima and T. Koyanagi, Kernels for Semi-structured Data, Proc. 19th International Conference on Machine Learning (ICML), pp.291-298, 2002.

[12] R. Agrawal and R. Srikant, Mining sequential patterns, Proc. 11th Int'l Conf. on Data Eng., pp.3-14, 1995.

[13] 宇野 毅明, 清見 礼, 有村 博紀, " 頻出・飽和・極大頻出集合の効率的な列挙アルゴリズムとその実装, " 第 4 回 データマイニングワークショップ, 1341-870X, no. 29, pp.47-54, Sept.2004.

[14] C. H. Chang and S. C. Lui, Iepad: Information extraction based on pattern discovery, Proc. 10th Int'l World Wide Web Conf., pp.4-15, 2001.