Vol.2010-FI-98 No.6
Vol.2010-DD-75 No.6
2010/3/5

繰り返し構造の検出に基づく Web ページの見出しの階層構造の解析

沙鵬[†]　松本章代[††]　小西達裕[†]
高木朗[§]　小山照夫[‖]　三宅芳雄[¶]　伊東幸宏[†]

文書中には類似した特徴を持つ見出しが反復的に現れる構造（繰り返し構造）がみられる．繰り返し構造を構成する見出し群は，文書の階層構造上では同一レベルに属すると考えられる．我々は先行研究において，Web ページ中の繰り返し構造を検出することにより見出しの階層構造を解析する手法を提案しているが，本稿では繰り返し構造の検出手法を改善することにより，見出しの階層構造の解析精度の向上を試みる．また提案手法の効果を実験的に評価した結果を報告する．

Analysis of Hierarchy of Headlines in Web pages Based on Detecting Repeated Structure

Peng SHA[†]　Akiyo MATSUMOTO[††]
Tatsuhiro KONISHI[†]　Akira TAKAGI[§]
Teruo KOYAMA[‖]　Yoshio MIYAKE[¶]　Yukihiro ITOH[†]

We have proposed a method to analyze a hierarchy of headlines in Web pages by detecting repeated structures. Our method can analyze the structure of Web pages that is not well structured. In this paper, we extend the method detecting repeated structures. In addition, we show an experimental evaluation of our method.

1. はじめに

WWW 上の情報量は現在も増え続けており，それに伴い，ユーザが求める情報を素早く正確に提示する検索システムの必要性も高まっている．しかし，現在の検索エンジンは不適合ページを相当数含む結果となることが少なくなく，十分な性能とは言いがたい．検索エンジンが不適合ページを誤検出してしまう原因の一つとして，検索に用いたキーワードが全く別の文脈で独立に使用されているページであっても適合ページと判定してしまうことが挙げられる．

そこで我々は，ページ内において検索キーワードがどのような関係を持って存在しているかという点に着目して，検索エンジンの性能を向上させることを試みてきた．検索キーワードとして複数の語が用いられた場合，それらの間には何らかの意味的な関係があると考えられる．したがって，意味的関係を表現しうる構造中に検索キーワードが含まれているページを優先的に扱うことにより，検索エンジンの性能の向上が期待できる[1].

キーワード間の意味的関係を表現しうる構造の一つに，見出しの階層構造がある．見出しの階層構造を利用した検索エンジンの性能向上については，先行研究[2]で検討しているが，システムが見出しの階層構造を正確に検出できないという問題点があった．これは， Web ページの多くが見出しの階層構造を表現するのに意味マークアップではなくレイアウト機能を用いていることが主な原因である．さらに，ページの制作者によって記述形式が異なることも問題を困難にしている一因である．

見出しの階層構造を正しく検出するため，我々はこれまでに Web ページ中の繰り返し構造を検出し，その情報を用いて見出しの階層構造の解析精度を向上させることを試みた[3]. しかし先行研究では Web ページ中に実際に現れる繰り返し構造がもついくつかの典型的特徴を考慮しておらず，検出手法に改善の余地があった．そこで本研究では，(1)繰り返される要素の分割に用いられる典型的記号（セパレータ）に着目した解析手法，(2)見出しと地の文の特徴の差異に着目した解析手法，(3)内部に繰り返し要素を含むブロック間の比較における要素間の反復回数の差異を柔軟に取り扱える解析手法を提案し，繰り返し検出精度を向上させる．そして，その効果を実験的に検証し，

[†] 静岡大学
Shizuoka University
[††] 青山学院大学
Aoyama Gakuin University
[§] 言語情報処理研究所
NLP Research Laboratory
[‖] 国立情報学研究所
National Institute of Informatics
[¶] 中京大学
Chukyo University

Vol.2010-FI-98 No.6
Vol.2010-DD-75 No.6
2010/3/5

繰り返し構造および見出しの階層構造の検出精度の向上効果を評価する．

2. 関連研究

Web ページの階層構造の解析を目的とした研究には以下のようなものがある．

文献[4]では，繰り返し構造の検出を行うことによって同じレベルの情報のセグメンテーションを行い，Web ページの構造化を行う手法を提案している．繰り返し構造に着目するという点で本研究と類似しているが，人間が構造を理解する上で大きな役割を果たしていると思われるレイアウトに関する情報（フォント色，フォントサイズ，画像サイズなど）を用いていないという点が本研究とは大きく異なる．

文献[5]では，テキストセグメント同士を比較し，教師あり機械学習によって親子関係を決定するという手法を提案している．階層の判断の材料には①DOM（Document Object Model）のパス②インデント情報③言語情報（先頭の記号やテキストの長さ，文末の句読点の有無，文末の品詞等）の 3 種類を用いている．しかし階層構造を表現するのに用いられることが多い，文字に関する視覚的な情報（背景色，フォント色など）は利用していない．

文献[6]では，携帯電話などの画面の小さな端末に Web ページを表示するために，DOM ツリーを手がかりに Web ページを分割する手法を提案している．しかし DOM だけに注目した場合，規則に則って書かれたページに対しては高い精度で解析できるが，イレギュラーな構造を含むページに対しては対応が困難である．

3. 先行研究

3.1 見出しの階層構造の解析

Web ページ中に現れた検索キーワードを含む見出しの間に，見出しの階層構造上の親子関係があるかを調べることにより，キーワード間の意味的関係の強さを判定できる．そのために先行研究[2]では，Web ページ中の見出しの階層構造を抽出する手法を以下のように提案した．

3.1.1 見出しの定義

(ア)一行の短い文で書かれており，他の見出しや文，図，表に対し一目で内容がわかるように付けられた表題．

(イ)事柄をいくつかに分けて書き並べている 1 つ 1 つ．他の見出しや文，図，表などの表題にはならないものもある．箇条書き．

3.1.2 見出しの検出手法

以下の特徴をもつ要素を見出しであると判定する．

- 以下のような見出しを表すタグで修飾されている要素
  - 見出しタグ（h1~h6），リストタグ（ul,ol,li），用語定義タグ（dl,dd,dt）
- レイアウト機能を用いて表現される以下の特徴をもつ要素
  - 文頭に数字，連番，記号がある行
  - 全体が強調されている行
  - 全体にデフォルト色以外の色が指定されている行
  - 全体がリンクである行
  - 末尾に“：”がある行
- クラス名において以下の特徴をもつ要素
  - クラス属性値が“midashi”もしくは“title”を含む．

3.1.3 見出しの階層構造の検出手法

2 つの見出し同士を比較し，親子関係を決定する．見出しの親子関係の判断材料としては，“文字の大きさ”，“文頭の記号の有無”，“背景色”，“強調の有無”，“下線の有無”，“インデント量”の六つを考慮する．ヒューリスティックに基づいて親子関係と，親見出しの支配範囲を決定するアルゴリズムを図 1 に示す．

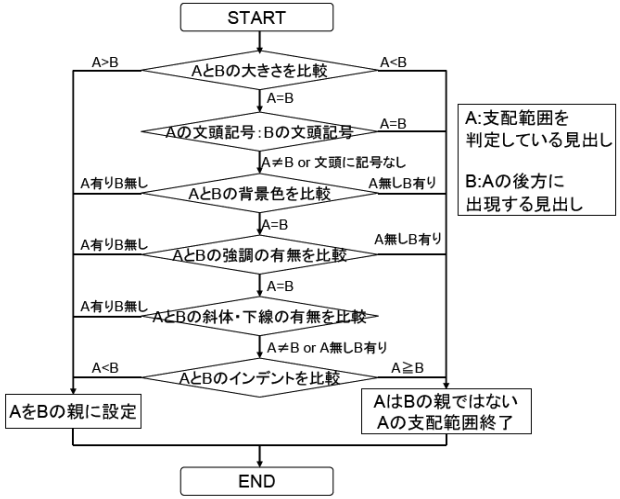


図 1 親子関係判定アルゴリズム

3.2 繰り返し構造の解析

3.1 の手法では局所的な情報のみをもとに構造を解析しているが，これを原因とする誤りが多い．先行研究[3]において，同レベルの見出しが反復的に現れる構造（見出しの繰り返し構造）に着目することにより，これらの誤りの解消を試みた．この研究では見出しの繰り返し構造を各見出しが持つ特徴の共通性に着目して解析する手法と，得られた繰り返し構造を見出しの階層構造の解析に反映させる手法を提案した．以下

Vol.2010-FI-98 No.6
Vol.2010-DD-75 No.6
2010/3/5

に手法の概要を示す．

3.2.1 用語の定義

・ブロック：見出しとボディで構成されるか見出しのみで構成される．
<ブロック>::=<見出し> <ボディ> ｜ <見出し>

・繰り返し構造：1 つ以上のブロックで構成される
<繰り返し構造>::=<ブロック> <ブロック>+

・ラベリング：Web ページ中の各見出しが持つ属性情報（詳細は 5.2 表 3 を参照）を解析し，各見出しに対して，全く同一の属性値を持つ見出しに対しては同じ値になるように整数値を割り振る．この数値を以下ではラベルと呼ぶ．これによって，Web ページはラベル列に変換される．以下ではこのラベル列を解析して繰り返し構造を検出する．

3.2.2 繰り返し構造の検出

ラベル列中のひとつのラベルに着目し，そのラベルが先頭になるようにラベル列を切り分ける．切り分けられたラベル列を以下ではブロックと呼ぶ．たとえばラベル列「1231231243」は 1 に着目すると「123/123/1243」という 3 つのブロックに切り分けられる．隣接するブロック同士を比較し，一致するもしくは一定以上の類似性がある場合にはこれらが繰り返し構造をなすと判定する．ここでブロック間の類似性の判定にはペアワイズアラインメントを用いる．これは，生物界でアミノ基の配列の類似性を判定するのに用いる手法であり，データ列の類似性を数値化することができる．この結果に対して一定の閾値を定めることによって上記の判定を行う．上の例では，ブロック「123」と「1243」がペアワイズアラインメントを用いて比較される．

さらに検出された各ブロック内のラベル列に対して，再帰的に同様の処理を行う．これにより，階層構造を持つ繰り返しを検出することが可能である．

以上の処理をそれぞれのラベルに着目点を移しながら繰り返す．最後に，最も多くのブロックを繰り返し構造とみなせる処理結果を最終的な処理結果として採用する．

但し上述のブロックの切り分けの際，要素が 1 ラベルのみのブロックはブロックとして認めず，その箇所までで繰り返しが途切れるものと考える．これは，1 ラベルのみのブロックを認めると，地の文を構成する各文がすべてブロックとなり，これらが繰り返し構造として認識されることを防ぐためである．

3.2.3 繰り返し構造を考慮した見出しの階層構造の検出手法

3.2.2 の方法で検出された繰り返し構造を利用して，3.1.3 の方法で検出した見出しの階層構造を以下のように修正する．

(1)繰り返し構造の 1 つのブロックの先頭行が見出しであると判定されているにもかかわらず，同じ繰り返し構造に属する同レベルのブロックの先頭行が見出しでないと判定されている場合，誤りである可能性が高い．よって後者は見出しであると判定する．またその支配範囲はそれぞれのブロックの末尾までとする．

(2)ある見出しが繰り返し構造のあるブロックに含まれる場合，その見出しの支配範囲は最長でもそのブロックの末尾までである．よって，これと異なる支配範囲が検出されている場合には修正する．

(3)ある見出しの支配範囲が繰り返し構造のブロックの先頭を含む場合には，支配範囲がそのブロックの途中で終わることはない．よって，これと異なる支配範囲が検出されている場合には修正する．

3.3 先行研究の問題点

問題点 1： セパレータを含む文書の繰り返し構造の検出

一般に Web ページなどの文書において，語・句・見出しなどを同一行に複数連記する場合に，一定の記号をその間に挟んで表記することが行われる．このような記号をセパレータと呼ぶことにする．セパレータの例としては，図 2 中の “-” や “>”, “/” などがある（詳細は 4.1.2 表 1 を参照）．3.2.2 に述べた従来の手法でラベル列を生成する際にはセパレータも通常の語句と同格の要素としてラベルを与えられていた．しかしこれらはそれ自体意味を持たず，支配範囲を持つ見出しにもなりえない．従来手法ではこのことを考慮していないため，明らかに誤った解析結果を出力するケースがあった．たとえば図 2 の例で，従来手法でラベリングするとセパレータ「-」はラベル「2」を与えられる．ここでは「当サイトについて」などの語句は全て同じ属性を持ち，ラベル「1」を与えられたとする．すると図 2 のように 1212121 というラベル列となり，「12」を 1 つのブロックとする繰り返し構造が出力され，「1」が「2」を支配範囲として持つ見出しと判定される．しかし正しくは図 2 の上のように，「2」を支配しない「1」の繰り返しと認識すべきであると考えられる．

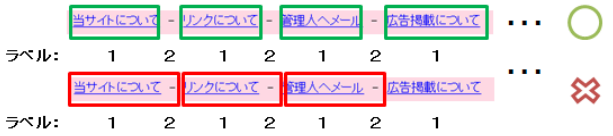


図 2 セパレータありの繰り返し構造

問題点 2： 1 つの要素のみで構成されたブロックには対応できない．

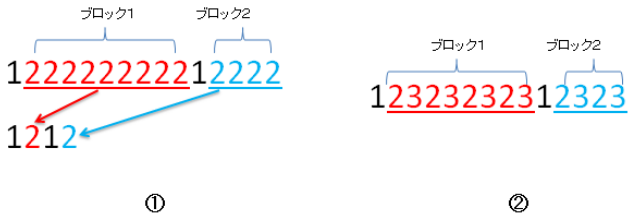
上述のように先行研究のアルゴリズムでは，要素が 1 つのみのブロックを排除していた．しかしこの制約は強すぎ，現実の Web ページの繰り返し構造の中でこれにより認識できないものが少なからず存在する．適切な制約となるように適用条件を加えるべきであると考えられる．

問題点 3： パターンの反復回数の差異を柔軟に吸収できない．

従来手法においてブロック間の類似性を判定する際に用いるペアワイズアラインメントは，ラベル列内に少々のノイズとなる要素が混入しても類似性を判定できる方法である．しかしながら，現実の Web ページにおける繰り返し構造では，図 3 のように，

Vol.2010-FI-98 No.6
Vol.2010-DD-75 No.6
2010/3/5

パターンの反復回数が異なるものの意味的には同レベルといえるブロックがしばしばみられる．図 3①のブロック 1 はラベル「2」が 9 回反復されており，ブロック 2 では 4 回反復されている．これらと直前のラベル「1」を組み合わせると，「「1」の後に「2」が n 回反復される」という構造が 2 回続けて現れたと解釈できる．しかし従来手法において，ブロック 1 とブロック 2 はペアワイズアラインメント手法では類似度が小さく評価されてしまい，上述のような繰り返し構造は認識できない．②も同様に「「1」の後に「23」が n 回反復される」という構造を認識できない．すなわち，パターンの反復回数の差異を許容できるようなブロック間の類似度判定手法が必要である．



① ②

図 3　先行研究の問題点

4. 基礎的考察

4.1 セパレータを含む文書からの繰り返し構造の抽出

セパレータを含む繰り返し構造の検出のために，まず一般にセパレータとして用いられる記号について調査した．後述するクローズドテスト用の Web ページデータ 49 ページを対象とする事例研究を行い，セパレータとして用いられている記号を全て抽出した．結果を表 1 に示す．

表 1　セパレータ（半角・全角の両方）

|	-	>	>>	,	/	＼	][	)(

システムはラベリングの際，これらの記号からなる要素には特別なラベル“Sn（n は記号毎に異なる整数）”を与える．Sn を含むラベル列に対しては，Sn の前後のラベル列を切り出してブロックを作り，それらを要素とする繰り返し構造の検出を試みる．但しその際，Sn は繰り返し構造の構成要素とはみなさない．

4.2 1 つの要素のみで構成されるブロックへの対応

3.3 問題点 2 で指摘したように，1 つの要素のみで構成されるブロックも現実の Web ページには存在する．これは言い換えれば，見出しのみで構成されたブロックである．このようなブロックを許容しつつ，地の文において文ひとつひとつがブロックになることを防ぐ必要がある．そのために，見出しであることを以下の条件で判定し，これらのいずれかを満たすブロックであれば，要素数が 1 でも認めることとする．

① 一行全体がリンクである
② 文の先頭が記号である
③ 文の先頭に番号があり，前後のブロックと連続した値をもつ
④ タグによって修飾されている

4.3 パターンの反復回数の差異を吸収できるようなブロック比較アルゴリズム

先行研究[3]では繰り返し構造のブロックの比較をラベル列そのもので行っていたが，3.3 問題点 3 で述べたことを考慮すると，ブロックの比較前に各々のブロック内に存在する繰り返し構造を再帰的に検出しておき，もし検出されたならばブロック間の比較の際，内部に含まれる繰り返しの回数の差異は問題にしないようなアルゴリズムを採用すべきである．たとえば図 3②の例では，まずラベル 1 に着目してこれが先頭となるように切り分けると「123232323/12323」となる．次いで，「123232323」「12323」のそれぞれに対して，再帰的に繰り返し検出処理を適用する．途中経過は略すが，これによりそれぞれのブロックは「「1」＋（「23」の 4 回の反復）」「「1」＋（「23」の 2 回の反復）」と解析される．これを受けてこれらのブロックを比較し，「（「1」＋（「23」の n 回の反復）の 2 回の反復）」であることを認識する．

5. 繰り返し構造の検出方法の改善案

本章では 4.の基礎的考察に基づいて改善した繰り返し構造の検出処理手法について述べる．全体の流れは先行研究[2]，[3]に準拠している．今回改善した部分については文中で特記する．

5.1 前処理

前処理として，解析対象である HTML ファイルを簡単に整形しておく．すなわち省略された終了タグの補完と開始タグ・終了タグの対応関係の修正を行う．またテーブルタグを用いて記述された要素に対して，表を記述するのに用いられたものにラベル“T”を付与しておく．先行研究[1]により，レイアウトの調整のために用いられたテーブルタグと表を記述するためのタグの判別を自動的に行うプログラムが実装されている．

5.2 ラベル列への変換

HTML ファイル全体をスキャンしてタグ以外のテキストを行単位で区切り各々に ID 番号をつける．ここで行単位とは，厳密にいえば表示時の行ではなく，ソースコード中のタグのうち意味的な区切りを表すもので分割される単位を意味する．現在は表 2 に示すタグをこの行区切りに用いている．これと同時に各行の属性情報（表 3 に示す）を取得しておく．

Vol.2010-FI-98 No.6
Vol.2010-DD-75 No.6
2010/3/5

表 2　行を区切る時に用いるタグ情報

直接行を区切る	html	body	head	title	h1	h2	h3
	h4	h5	h6	table	tr	th	td
	dl	dt	dd	ul	ol	li	address
	blockquote	center	caption	div	hr	p	pre
条件付きで行を区切る	br:a の中，li の中，h1..h6 の中，以外の時に行を区切る						
	img:a の中，li の中，h1..h6 の中，画像サイズ 40*20 以上，以外の時行区切る						

次に，取得した属性情報のうち，“class 属性”，“リンクの有無”，“強調の有無”，“文字の大きさ”，“文字色”，“背景色”，“先頭の記号”，“先頭の連番”の 8 つに基づいて同じ属性を持つ行同士で 1 つのグループをなすように全ての行をグループに分類する．次に全グループに重複しないようにラベル（0,1,2,3…）を付ける．但し 4.1 で述べたセパレータと考えられる要素にはラベル Sn（n=0,1,2,3…）をつける．以上の処理によって，Web ページをラベル列に変換したことになる．

表 3　属性情報

1	見出しタグ（h1,h2,h3,h4,h5,h6）
2	リストタグ（ul,ol,li）
3	定義リストタグ（dl,dt,dd）
4	文字の大きさ
5	強調の有無
6	リンクの有無
7	class属性
8	文字色
9	背景色
10	先頭の記号（“○”，“◆”，“※”な
11	先頭の連番（“1.”，“2-1”など）
12	括弧
13	下線の有無
14	インデント量

5.3　ペアワイズアラインメント

3.2.2 で述べたように，2 つのブロックの類似度を量るためにペアワイズアラインメントを用いる．先行研究[3]で算出した各要素のペナルティー値（表 4）を用いて，ラベル間の差異の程度を表すスコア行列を計算する．これをもとに，ブロック間のラベルの出現順序からブロックの類似度を示すスコアを算出する．このスコアが閾値以上であれば 2 つのブロックは同じ繰り返し構造に属することができる程度に類似していると判断する．今回の閾値は，6 章で述べるクローズドテストに用いたデータセットを対象とした実験結果に基づいて，-5.0 とした．

表 4　属性値のペナルティー

文字の大きさ	−24.55	文字色	−11.25
強調の有無	−19.29	背景色	−12.85
リンクの有無	−5.13	先頭の記号	−20.17
class属性	−9.64	先頭の連番	−29.41

5.4　繰り返し構造の探索

5.4.1 セパレータありの繰り返し構造検出アルゴリズム

ページ中にセパレータ候補の記号（表 1）が現れた時，その記号の前後のブロックを比較し，“リンクであるか否か”，“色”，“背景色”の三つの条件が全て一致する場合，該当するセパレータの左側の要素と右側の要素が同じ繰り返し構造の要素をなすと判定する．この方法でラベル列を先頭から順番にスキャンし，セパレータありの繰り返し構造を検出する．

ここで，セパレータありの繰り返し構造に含まれる要素はブロックの先頭見出しになるとは考えられないため，以後の解析でブロック分割時に先頭要素の候補として着目されないように，特別なラベル Nx（x=0,1,2,3… 繰り返し構造毎にユニークな番号とする）を付与しておく．探索できた繰り返し構造は一定の表形式で保存する．これを繰り返し構造プールと呼ぶことにする．繰り返し構造プールへの保存方法については 5.4.3 で詳述する．

5.4.2 セパレータなしの繰り返し構造検出アルゴリズム

4.2 および 4.3 で提案した手法を以下のように処理アルゴリズムに反映させた．

(1)基本処理

① ラベル列 X 内において 2 回以上出現する，“T”，“Nx”以外のラベルの集合を L={L1,L2,L3,...Ln}，繰り返し構造を構成するブロックの集合を R={},X 内のラベルを指すポインタを i とする．
② ラベル集合 L からあるラベルを取り出す．取り出されたラベルを Lx とする．
③ ポインタ i を X の先頭にセットする．
④ i を一つずつずらしてゆき，Lx が出現した位置をブロック候補の先頭とする．
⑤ i を一つずつ後方へずらしてゆき，「Lx が次に出現する箇所の直前」か「R に含まれるブロックの先頭の直前」か「R に含まれるブロックの末尾」までをブロック候補の末尾とする．ただし，ブロック候補に含まれるラベル数が 1 の場合，4.2 で述べた見出し特徴付き要素でなければ，そのブロック候補は破棄する．
⑥ i を「ブロック候補の末尾＋1」にセットする．
⑦ ④～⑥の処理を i が X の末尾にたどり着くまで繰り返し，ブロック候補

Vol.2010-FI-98 No.6
Vol.2010-DD-75 No.6
2010/3/5

群を作成する．

⑧ ②～⑦で作成されたブロック候補群を R に追加する．

⑨ ②～⑧の処理を，L が空になるまで繰り返す．

⑩ 最後に探索できたブロックから最初に探索できたブロックへ逆順で各々と隣接しているブロックとの類似度を計算し，繰り返し構造を構成しうるかどうかの判定を行う．構成しうると判断された場合は繰り返し構造プールに保存する．ただし，類似度を計算する前に以下の処理を行う．これは，4.3 で述べたパターンの反復回数の差異を吸収するためである．(1)繰り返し構造プールを参照し，比較するブロック中に繰り返し構造があれば 1 つにまとめる．(2)同一のラベルが連続している箇所が比較するブロック中にあれば 1 つにまとめる．

(2)Lx の最適な着目順序を探索する処理

上述の基本処理では，②においてどのラベルを取り出すかに自由度がある．ラベルを取り出す順序によって，仮定されるブロックの切り分け位置が変わる．すなわち，最も妥当な繰り返し構造を検出するためには，ラベルの最適な取り出し順序を探索する必要がある．よって，上述の基本処理を，ラベルの取り出し順序を変えながら反復する．それぞれの処理結果において，検出された全ての繰り返し構造に含まれる繰り返し回数を合計し，最大になった取り出し順序を最適なものとする．ただし，最大回数となる取り出し順序が複数存在する場合，以下の方法で各繰り返し構造の重みを計算し，重みの合計が最大の取り出し順序を最適なものとする．

重みの定義には 6 章で述べるクローズドデータセットを用いた．これによれば 95%以上の繰り返し構造の先頭要素はリンク，記号，連番，リストタグで修飾された箇条書きのいずれかの属性を持っていることがわかった．よって，ブロックの先頭行が上述のいずれかの属性を持つ場合，そのブロック数を重みとしてスコア化する．

ラベルの種類数を n としたときラベルを取り出す順序の組み合わせの数は n!となり計算が困難になると思われるが，実際にははじめに 1 つのラベルを取り出してブロックの切り分けを行うとブロックが小さくなり，そこに含まれるラベルの種類数は(n-1)よりもさらに小さくなるため，階上オーダーの組み合わせを試行する必要はない．なお，実データに対する事例研究によれば，標準的な Web ページに含まれるラベルの種類数は平均で 10 程度であり，初回のブロック分割でおよそ 5～6 種類程度になる．結果としてシステムを実装した際に処理時間が問題になることはなかった．

5.4.3 繰り返し構造プールへの保存

繰り返し構造プールの要素は検出された繰り返し構造であり，各要素は属性情報として，繰り返し構造の ID，先頭要素の ID，末尾要素の ID，繰り返しの基礎単位となるラベル列（例：「121212」から「12」の反復を検出したなら「12」）をもつ．しかし，ブロックが完全には一致せず類似性の高さから繰り返し構造と判定した場合には，どのブロックを基礎単位となるラベル列とするかを決定する必要がある．そのためには各ラベル列の出現頻度を数え，最大のものを基礎単位とする．頻度が等しいものについては，ヒューリスティックスにより出現位置を基準に選択する．

6. 評価実験

以上述べた手法に基づいて繰り返し構造の検出システムを実装した．この章ではこのシステムの性能を実データに基づいて実験的に評価するとともに，先行研究[3]で構築した同じ目的のシステムの性能と比較し，提案手法の効果をはかる．

6.1 評価用テストセット

評価用テストセットの作成手順を以下に示す．

6.1.1 対象データの選定方法

① 2 語の検索キーワード対 118 組を用意しその 118 組を Google で検索し，検索結果上位から 10 ページずつ合計 1180 ページを用意する．

② サイズが 0 バイトのページを削除する．

③ 残りをサイズ順にソートする．

④ ③のサイズ順データの中間層から 200 ページを仮候補とする．

⑤ ドメインが同じページを削除し，減少した分を新たに追加する．

⑥ キーワード対の偏りをなくすために，キーワード対ごとに最大で 3 ページとなるように，ページの追加・削除を行う．

6.1.2 繰り返し構造，見出し，支配範囲情報の付与

① 被験者（大学生 3 名）に見出しとその支配範囲を抽出させ，3 人の判断が一致した 166 ページをテストセットとする．

② 被験者（大学生 3 名）に，繰り返し構造を抽出させ，3 人の判断が一致したもののみを繰り返し構造として扱う．

③ 166 ページに対して見出し，支配範囲，繰り返し構造に関する情報を付与する．

④ クローズドテストセット 49 ページ，オープンテストセット 117 ページとし，これらを評価用テストセットとする．

なお，4 章と 5 章で述べた手法はクローズドテストセットのみを分析対象として構築した．

6.2 繰り返し構造の検出の評価

6.1 の手順で作成したテストセットを用いて，5 章で述べた繰り返し構造の検出手法の性能評価について述べる．

6.2.1 評価用語の説明

・ 完全一致（Perfect）：正解の繰り返し構造とシステムが出力した繰り返し

Vol.2010-FI-98 No.6
Vol.2010-DD-75 No.6
2010/3/5

構造が完全一致する．

- 誤検出（Fa）：正解の繰り返し構造とシステムが出力した繰り返し構造に何らかの差異がある．
- 検出洩れ（Miss）：正解の繰り返し構造の全部もしくは一部をシステムが検出できない．
- 精度（Precision）：システムが出力した繰り返し構造にどのぐらい正しいものが含まれていたかの割合．
  Precision = Perfect　/（Perfect+Fa）
- 再現率（Recall）：システムが正解データをどのぐらい正しく検出できたのかの割合．
  Recall = Perfect　/（Perfect+Miss）
- F 値（F-Measure）は精度と再現率の調和平均を表す．
  F-Measure = 2 /　（1/Precison + 1/Recall）

6.2.2 先行研究と提案手法の評価結果

表 5　繰り返し構造の抽出実験（先行研究）

	PERFECT	FA	MISS	Precision	Recall	F-Measure
ClosedTest	98	119 （中の 70 ファイル部分一致）	186 （中の 70 ファイル部分一致）	0.452	0.345	0.391
OpenTest	199	296 （中の 161 ファイル部分一致）	462 （中の 161 ファイル部分一致）	0.402	0.301	0.344

表 6　繰り返し構造の抽出実験（提案手法）

	PERFECT	FA	MISS	Precision	Recall	F-Measure
ClosedTest	114	152 （中の 75 ファイル部分一致）	140 （中の 70 ファイル部分一致）	0.429	0.449	0.438
OpenTest	252	354 （中の 163 ファイル部分一致）	334 （中の 137 ファイル部分一致）	0.416	0.430	0.423

先行研究[3]と提案手法による繰り返し構造の検出能力を上述のテストセットを用いて比較した．closed テストにおいては，本手法の F 値は先行研究の手法より 4.7 ポイント向上している（向上率は（(0.438/0.391）-1=12%）．open テストにおいては，本手法の F 値は先行研究の手法より 7.9 ポイント向上している（向上率は（(0.423/0.344）-1=23%）．以上より，提案手法の有効性が検証された．

上記の評価尺度は,部分的にでも正解データと検出結果が一致しないと完全に失敗した場合と同等の評価になるような計算方法をとっている．そのため Precision, Racall ともに 50%を下回る低い数値となっている．しかし実際には正解の繰り返し構造とシステムの出力した繰り返し構造が一致している場合も多い（表 5，表 6 の Fa と Miss の欄にある「部分一致」）．これを加味し，部分的一致をその一致度に応じて Perfect に算入すると，提案手法の closed テストではシステムの出力のおよそ 60%が，open テストではシステムの出力のおよそ 56%が正解と重なっている（Precision に相当）．また closed テストでは正解データのおよそ 62%が，open テストでは正解データのおよそ 58%がシステムの出力データによりカバーされている（Recall に相当）．

6.2.3 考察

実験結果を分析したところ，以下のような問題点を発見した．繰り返し構造のブロック内のラベル列の差異に，現在想定している属性に基づく類似度計算ではうまく評価できないものがある．現在の Miss のうち約 3 割がこれを原因とする．失敗事例から形式的に類似性を判定しうる要素を探すと，div，span，table タグなどの中に含まれるパス情報が一致するブロックが連続して現れた場合，ブロック内のラベル情報に比較的大きな差があっても，人間の目から見て類似性の高いブロックと感じられることがわかった．今後，このような現在利用していない特徴を組み込んだ類似度計算手法を開発する必要がある．

6.3 見出し検出の評価

6.3.1 先行研究と本手法の評価結果[a]

表 7　見出しの抽出実験（先行研究）

	HIT	MISS	FA	Precision	Recall	F-Measure
ClosedTest	1381	537	531	0.722	0.707	0.714
OpenTest	2659	1207	1226	0.684	0.688	0.685

表 8　見出しの抽出実験（提案手法）

	HIT	MISS	FA	Precision	Recall	F-Measure
ClosedTest	1567	265	441	0.780	0.855	0.816
OpenTest	3411	928	1001	0.773	0.786	0.780

先行研究[3]と提案手法による見出し検出能力を上述のテストセットを用いて比較した．closed テストの F 値については，本手法の評価結果は先行研究より 10.2 ポイント向上した（向上率は（(0.816/0.714）-1=14%）．open テストの F 値は，9.5 ポイント

a ）　Hit: プログラムで正解を検出できたもの
Fa: プログラムで不正解を誤検出したもの
Miss: プログラムで正解を取りこぼしたもの
Precision: Precision（精度）=Hit/(Hit+Fa)　Recall（再現率）=Hit/(Hit+Miss)

向上した（向上率は（(0.780/0.685）-1=14%）.

6.3.2 **考察**

失敗したケースを分析したところ，3.1.2 節で述べた見出し判定材料の中で，「行全体がリンクの時に見出しと判定」というヒューリスティックスが比較的多くの誤検出の原因となっていることがわかった．誤検出例では，そのリンクが文書中であまり重要でない位置に出現したケースが多かった．よって，見出し候補が文書中で位置的に重要と思われる場所にあるかどうかを尺度として判定に加えることで誤検出を減らせる可能性がある．一般に，文書中のある領域（段落など）の中で，先頭に現れる要素は末尾に現れる要素よりも重要性が高い場合が多いと考えられるので，このような特徴を考慮した判定手法を今後検討する．

6.4 見出し先祖子孫関係を検出する手法の評価

ここでは繰り返し構造の検出結果を利用して見出しの抽出とその支配範囲の推定を行い，その結果得られた見出し先祖子孫関係の正しさを評価する．

2 つの見出しの先祖子孫関係を特定することができれば，検索エンジンにおいて，ユーザに入力されたキーワードが見出しの位置に現れた時，見出しの先祖子孫関係の有無により，2 つの離れたキーワードがお互いに意味的に係り受け関係があるかどうかを判断することができる[2]．これにより，該当の Web ページにユーザの意図している情報が存在しているかどうかを判断できる．従って先祖子孫関係の抽出の正確さは，提案手法を検索エンジンの性能向上に役立てる上で重要な意味をもつ．

6.4.1 **先行研究と現手法の評価結果**[a]

表 9　見出し二項関係抽出実験（先行研究）

	HIT	MISS	FA	Precision	Recall	F-Measure
ClosedTest	2598	1513	1088	0.705	0.632	0.670
OpenTest	3876	3131	2331	0.624	0.553	0.586

表 10　見出し二項関係抽出実験（提案手法）

	HIT	MISS	FA	Precision	Recall	F-Measure
ClosedTest	2709	1409	1102	0.711	0.658	0.683
OpenTest	5519	3626	3180	0.634	0.603	0.619

先行研究[3]と提案手法による見出し検出能力を上述のテストセットを用いて比較した．closed テストの F 値については，本手法の評価結果は先行研究より 1.3 ポイント向上した（向上率は（(0.683/0.670）-1=2%）．open テストの F 値は，3.3 ポイント向上した（向上率は（(0.619/0.586）-1=6%）．

6.4.2 **考察**

先祖子孫関係の判定失敗例をみると，その多くは見出しの抽出の段階での誤りが原因となっており，正しく抽出された見出しに対して支配範囲の判定を誤ったことで起こっている失敗は比較的少ない．よって，先祖子孫関係の抽出能力の向上のためには，特に上位にあり広い支配範囲をもつ見出しの抽出能力を向上させることが最も効果的と考えられる．

7. むすび

本研究では，先行研究で提案された文書中の繰り返し構造の検出方法を改善し，検出精度を向上させた．見出し階層構造の解析においても先行研究より良い結果が得られることを確認した．しかし，特に繰り返し構造の抽出精度についてはまだ向上の余地があると考えている．現在アルゴリズムやヒューリスティックスの開発に用いているクローズドテストデータは 49 ファイルであり総計で 254 個の繰り返し構造しか含んでいないため，Web 上の文書の特徴を把握するには少々不足があり，オープンテストデータに含まれる繰り返し構造の特徴をカバーできない例も多々みられる．よって今後の課題のひとつとして，クローズドテスト用データを増やす必要がある．またブロック間の類似性の判定のために，タグ中のパス情報の共通性などこれまで用いていなかった情報を取り入れることも今後の課題である．

参考文献

1) 松本章代，小西達裕，高木朗，小山照夫，三宅芳雄，伊東幸宏：表構造における意味的関係に基づく WWW 検索性能の向上，電子情報通信学会論文誌，D Vol.J91-D，No.3，pp.560-575(2008)
2) 西口直樹，松本章代，小西達裕，高木朗，小山照夫，三宅芳雄，伊東幸宏：見出しの階層関係を利用した WWW 検索精度の改善，電子情報通信学会技術研究報告，NLC2005-114，Vol.105，No.595，pp.1-6(2006)
3) 池田彰吾，松本章代，小西達裕，高木朗，小山照夫，三宅芳雄，伊東幸宏：繰り返し構造を考慮した Web ページの見出しの階層構造の解析，情報処理学会研究報告，DD，Vol.2008，No.34，pp.31-38(2008)
4) 南野朋之，齋藤豪，奥村学：繰返し構造に基づいた Web ページの構造化，情報処理学会論文誌，vol.49，No.9，pp.2157-2167(2004)
5) 松本吉司，高橋哲朗，乾健太郎，松本裕治：Web ページのテキストセグメント階層構造の抽出，言語処理学会，大 11 回年次大会，発表論文集，vol.11th，pp.49-52(2005)
6) Yu Chen,Wei-Ying Ma,Hong-Jiang Zhang：Detecting Web Page Structure for Adaptive Viewing on Small Form Factor Divices.In Proc. World Wide Web Conference 2003,2003