階層的見出し構造に着目したWebページ検索システム

真鍋　知博†　　田島　敬史††

†,†† 京都大学大学院情報学研究科　〒606–8501 京都府京都市左京区吉田本町
E-mail: †manabe@dl.kuis.kyoto-u.ac.jp, ††tajima@i.kyoto-u.ac.jp

あらまし　本論文では，Webページ中の見出し構造を利用したWebページ検索の精度向上手法と，その評価について述べる．見出し構造とは，見出しの付いたまとまった記述単位であるブロックの包含関係から成る階層的な論理構造である．ブロックの境界では話題が変わることが多く，また，各ブロック中では，そのブロックやそれを包含する上位のブロックの見出しに含まれる語はしばしば省略される．そこで，提案手法ではブロック単位でスコアを計算することとし，また，その際に，そのブロックにかかる階層的見出し中の語を補った上でスコアを計算する．評価は，通常のWebページ検索結果を提案手法によってリランキングした場合の，前後の精度の比較により行う．

キーワード　Web検索，セグメンテーション，構造化文書，Web閲覧支援

1. はじめに

Web情報の増大と利用局面の拡大により，Web検索の重要性が増している．様々なWeb情報源の中でも，テキストで記述されたWebページは，主要な情報源であり，またハイパーリンクや参照により，画像や動画などの各種Web情報を結びつけるハブの働きも果たしている．そのため，Web検索の中でも，キーワードによるWebページ検索は特に重要である．

人間に代わりWebページ検索を行う高性能なシステムの構築のためには，人間がページの内容を理解する際にとる閲覧行動を分析する必要がある．我々はそのような閲覧行動のうち，見出しを利用した必要な情報の記載部分の特定に着目した．本研究における見出しとは，章や節・ブログの記事・検索結果アイテムなどのまとまった記述の先頭付近に置かれ，その内容を端的に表す部分である．見出しによって構造化されたWebページは多く，それらは見出し構造を持つと言える．

見出し構造とその人間による利用には，共に多くの特徴が存在する．しかし，Webページが明示的に持つタグによる木構造と暗黙に持つ見出し構造との乖離のため，見出し構造の自動利用は十分に研究されてはいない．そこで我々は，前著[1]で，Webページからの見出し構造の抽出手法を提案し評価した．

本稿では，Webページから抽出された見出し構造を利用した，通常のWebページ検索結果のリランキング手法を提案し評価する．提案手法では，見出しの付けられた各部分は，ある話題に関する記述の単位であると見なし，それらの部分ごとにスコアを計算してランキングに用いる．また，各部分とその階層的な見出しの間での文脈の継承の扱いや，各部分が別の部分を包含する場合にどこまでを単位としてスコアを計算するかなどを考慮して，複数の手法を提案し，実験による比較を行う．

以下，本稿では，章2.で見出しとその利用に関する仮説を立てて見出し構造を詳細に定義し，章3.で前著[1]に基づく見出し構造の抽出手法の概要を述べ，章4.で見出し構造に着目したリランキング手法を提案し，章5.でリランキング手法を評価し，章6.で関連研究を紹介し，章7.で結論を述べる．

京都水族館
京都水族館は，日本の京都市にある梅小路公園内の水族館．
概要
日本最大級の内陸型水族館である．
動物約250種・総数約15,000匹を展示している．
利用案内
免責事項もご覧ください．
休館日
なし，年中無休．臨時休業あり．
営業時間
午前9時から午後5時．午後4時まで受付．
沿革
詳細は京都水族館の沿革を参照．
2010年
・**7月**　水族館の建設着工．
2012年
・**2月**　*下旬*　水族館の建設竣工．
・**3月**　計画通り，京都水族館開業．
・**7月**　*上旬*　入館者数が100万人に．

図1　見出し構造を持つWebページの例．

2. Webページ中の見出し構造

本章では，本研究が対象とするWebページ中の見出し構造とは，どのようなものであるか述べる．

2.1 見出し構造について考える意義

人間によるWebページ閲覧行動の理解の必要性については，章1.で述べた．本研究では，閲覧行動のうち，特に見出し構造の利用に着目する．階層的見出しによる構造化は，ある程度長い文書である書籍・論文などの構造化に昔から広く用いられており，人間による文書の閲覧行動にも大きく影響すると考えられる．階層的見出しはWebページの構造化にも広く用いられており，現在，最も普及しているWebページ記述言語であるHTMLには，6レベルの見出しを表すタグが定義されている．階層的見出し構造を持つWebページの例を図1に示す．

しかし，見出しを表すタグを用いて記述された見出しは，ブラウザによる表示上は見た目が強調された文字列となり，見た目を指定するタグを用いて強調された文字列と区別がつかない．

このため，見た目を指定するタグを用いて記述されており，人間には見出しであると認識できるが，タグ名のみから機械的には見出しであると認識できない見出しが多く存在する．そこで，見出し構造の利用にはまず，その自動抽出が課題となる．

前著 [1] で我々は，見た目上の特徴と木構造上の特徴を用いた，見出し構造の自動抽出手法を提案したが，当該手法による見出し構造抽出の理論的根拠は十分には示さなかった．本稿では，人間による閲覧行動を分析し，見出し構造とその人間による利用に関する四つの仮説を立て，当該手法がそれらに沿うことの確認，および，Web ページ検索結果の，各ページ中の見出し構造を利用したリランキング手法の提案と評価を行う．

2.2 見出し構造に関する仮説

見出し構造に関する，本稿の基礎を成す仮説は四つある．

まず，見出し構造そのものの特徴に関する仮説を述べる．

取捨 見出しは対応部分の極めて簡潔な要約である．そのため人間は，見出しを閲覧することで，その対応部分を読む必要性をある程度判断できる．この最たる例は，新聞の見出しである．

継承 見出しの対応部分では，その見出し中の語の指す事物に関する言及は，その語を直接用いず暗黙に行われ得る．例えば人間は，図 1 から「京都水族館の利用案内によれば，その営業時間は午前 9 時から午後 5 時である」という情報を得る．

次に，人間による見出し構造の利用に関する仮説を述べる．

走査 取捨の仮説で述べた見出しの特徴のため，人間は見出しだけを走査して文書中で読む必要のある部分を絞り込める．ここで走査するとは，単に次々に読むことである．見出しとページ番号のみから成る書籍の目次は，この例である．具体例として，図 1 のページから営業時間を知りたい場合，人間はまず大見出しだけを「概要」,「利用案内」,「沿革」の順に走査し，営業時間に関係の深い「利用案内」の部分を読むであろう．

走査順序 上の例では，人間は引き続き小見出しを「休館日」,「営業時間」の順に走査し,「営業時間」の部分を読むであろう．このように人間は，複数レベルの見出しを含む文書に対しては，まず見出しの見た目を元にそれらの間の大小関係を理解する．次に上位の見出しを走査し，興味のある見出しの対応部分だけを閲覧する．そこに下位の見出しが含まれる場合，それらを走査する．つまり人間による見出しの走査は，レベルが高い順に，かつレベルごとに行われる．この例は，初めは大見出しだけが表示され，見出しをクリックすると小見出しを含む対応部分が表示されるウィキペディア・モバイル(注1)の記事である．

2.3 諸概念の定義

仮説の通り，見出し構造とその利用には様々な特徴がある．本節では，それらを考慮しつつ，見出し構造を詳細に定義する．

2.3.1 見出しとブロック

まず取捨の仮説にあるように，本研究では**見出し**を，文書のある部分に置かれ，その部分の内容を端的に表す箇所であると定義する．この定義の通り，見出しには対応部分が存在し，本研究では見出しの対応部分を**ブロック**と呼ぶ．ブロックは，見出しに要約可能な程度の意味的まとまりを持つ，一続きの記述である．ブロックの例として，文章における章や節，ブログの各記事，検索結果アイテムなどがある．このように，本研究における見出しとブロックとは一対一に対応する．

なお，本研究では見出しのみから成るブロックを考慮しない．これを考慮する場合，任意の文や画像は見出しかつブロックであるが，それらは容易に抽出でき，検索の際に見出しと非見出し部分の間で情報を補い合う必要もないためである．

2.3.2 スタイル

更に取捨の仮説によれば，見出しはブロックを読む必要性の判断に用いられるため，ブロックの中で初めに閲覧されるべき箇所である．このため見出しは，ブロックの中でも前方に配置されることが多く，他の部分とは異なり目を引くスタイルによって強調されることが多い．ここで**スタイル**とは，文字列に関する文字サイズや文字の太さなど，見た目の情報である．

2.3.3 リスト

見出しのスタイルの特徴を踏まえた上で，走査の仮説によれば，人間は Web ページ中の見出しだけを走査して，ページ中で読むべき部分を絞り込める．このため Web ページ中には，人間が走査しやすいよう統一されたスタイルの見出しの並び，ひいてはブロックの並びが存在することがある．見出しが同スタイルであるブロックの並びを，本研究における**リスト**と定義する．

本研究におけるリストは，それを構成するブロック間にリストとは無関係な記述を挟み，不連続でもよい．例えばカテゴリ別の商品の並びは，間にカテゴリ名を挟み不連続でも，商品の見出しのスタイルが統一されている限りリストである．この例で，単一の商品のみを含むカテゴリが存在する場合を考える．このカテゴリだけを見れば，商品を独立のブロックとして扱う必要性は定かではない．しかし，複数の商品を含む他のカテゴリも存在するなら，人間は商品の見出しを走査する可能性がある．そこで前述の単一の商品も含め，全ての商品が不連続なリストを構成すると見なせば，全商品を独立のブロックとして正しく扱える．不連続なリストにはこのような有用性がある．

本研究におけるリストは，日常会話や先行研究における，類似の構造を持つブロックの並びを指す狭義のリストよりも広い構造を含む．そして走査のため，この意味のリストは有用であり，実際の Web ページも数多くのリストを含む．文章の章や節の並び，タイトル以外に共通の属性を持たない各種のメディアが混在する検索結果アイテムの並びなども，またリストである．図 1 の例の小規模なページであっても，章のリスト,「休館日」・「営業時間」の節のリストなど多くのリストを含む．

2.3.4 ブロックの包含とレベル

一般にブロックは，内部にいくつか他のブロックを**包含**することがある．ブロック間の包含関係の例は，各章がいくつかの節を含む文書，カテゴリ別の商品一覧など多数ある．図 1 の例では,「沿革」ブロックは「2010 年」と「2012 年」ブロックを包含し，前者は 1 つ，後者は 3 つのブロックを更に包含する．

包含を考慮すると，包含するブロックを上位，包含されるブロックを下位とする，ブロックの**レベル**を考えることができ，ページ中のブロックは階層構造を成す．見出しとブロックとは一対一に対応し，走査順序の仮説における見出しの大小とは，レベルの上下に対応する概念である．そこで，ブロックのレベ

(注1)：http://ja.m.wikipedia.org/

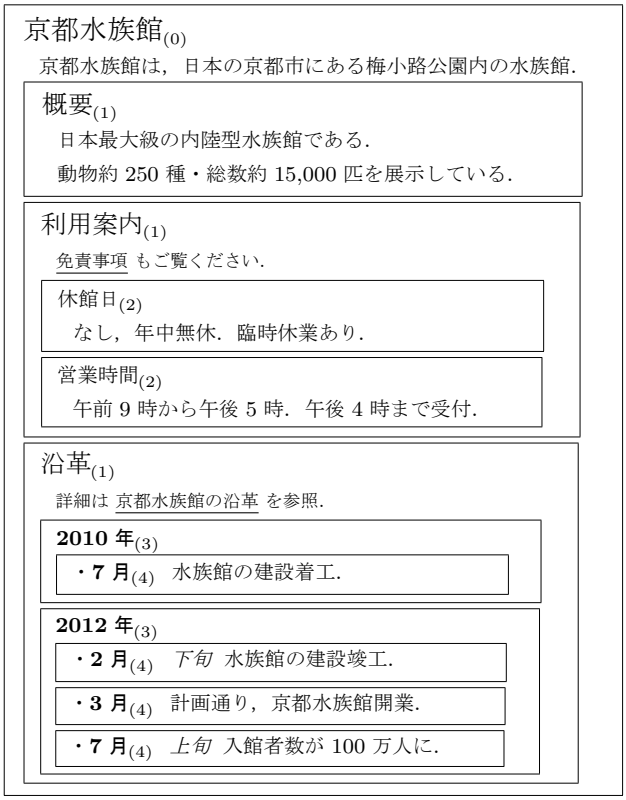

図 2　見出しに添字を付し，対応するブロックを囲み線で表し，見出しの添字の別によって対応するブロックが構成するリストを区別した，図 1 のページ中の見出し構造.

ルをもって，見出しのレベルとする．人間による見出し構造の理解においては，逆に，スタイルに基づく見出しのレベルの推定により，ブロックの包含関係が理解されると考えられる.

更に走査順序の仮説の通り，人間は見出しのレベルをそのスタイルによって判断すると考えられるので，同スタイルの見出しのレベルは等しいとする．リストを構成する全てのブロックの見出しは同スタイルであるので，リストを構成する全てのブロックのレベルは等しく，それをもってリストのレベルとする.

2.3.5　見出し構造

以下に，ここまでの定義を用いて走査順序の仮説を書き直す.

走査順序 *2*　人間は，文書を閲覧する際，まずスタイルを元に各見出し・各リストを発見しそれらのレベルを理解する．次に，最上位のリストを構成するブロックの見出しだけを走査し，興味ある見出しを持つブロックだけを閲覧する．そのブロックが下位のリストの一部または全部を包含する場合，それを構成するブロックの見出しだけを走査し，興味ある見出しを持つブロックだけを閲覧する．このような閲覧行動を再帰的に行う.

タイトルや URL は，Web ページ全体の見出しであるとも考えられる．そのため，ページ全体はブロックの条件を満たし，一つのブロックから成るリストでもある．Web ページの**見出し構造**とは，ページ全体を表す最上位のリストを含め，レベルの異なるいくつかのリストによる階層構造である．具体的な例として，図 1 のページ中の見出し構造は，図 2 のようになる.

3.　見出し構造の自動抽出手法

我々は前著 [1] で，Web ページ中の見出し構造を自動的に抽出する手法を提案した．本稿では詳細は省略するが，当該手法は走査順序 2 の仮説に基づき，人間の閲覧行動を模倣して見出し構造を抽出する手法であるといえる．すなわち当該手法に沿うシステムは，見出しを上位から順に，レベル毎に走査し，リストされたブロックの切り分けにより見出し構造を抽出する．見出し構造の自動抽出手法の概要は以下の通り．入力は木構造を持つ単一の Web ページであり，出力は見出し構造である.

（1）　全てのノードのスタイルを得る.

（2）　テキスト・img ノードをスタイル別の集合に分別する.

（3）　全てのスタイル別集合の「巣集合」を求める．詳細は略すが，巣集合とは巣ノードの集合であり，あるノードの巣ノードとは，注目するノードを含み，注目するノードと同スタイルの他のノードを含まない最大の部分木のルートを表す.

（4）　巣集合の深さの昇順，CSS の *font-size* 値の降順，*font-weight* 値の降順，文書順をこの優先順で用いて，スタイル別集合を「見出しの集合としてのレベルが高い順」に整列する.

（5）　文書全体はルートノードに対応する一つのブロック，その見出しは title ノード（不存在なら URL）とマークする.

（6）　整列されたスタイル別集合を，上位のリスト見出し集合らしい順に一つずつ取り出し，以下の操作を行う：

（a）　注目するスタイル別集合の巣集合の各要素から，次の巣ノードに達さず，かつ上位のノード列を外れない範囲で兄弟関係を文書順に辿り，各見出しの係り先のノード列を得る.

（b）　注目するスタイル別集合が以下のどれにも該当しなければ，それをリストされたブロックの見出しの集合，すなわちリスト見出し集合と見なす．(i) 上位のブロックにより個別に切られたノードの集合，(ii) 巣集合が，他の巣集合の真部分集合である集合，(iii) 見出しのみを包含するノード列が得られる集合．リスト見出し集合と見なされた場合は，その各要素を見出し，各要素に対応するノード列をブロックとしてマークする.

本稿で用いた手法の概要は上述の通りだが，前著 [1] の手法からは以下の変更を加えている．第一に，前著で区別していたセパレータと視覚的見出しは多くの場合に一致するため，セパレータのみを見出しとして抽出する．そのため，視覚的見出しの抽出に用いていた CSS の属性値の大小の情報を，スタイル別集合の整列に用いる．第二に，見出し以外の情報を含まないブロックを考慮せず，(iii) でそれらを除去する．前著で除去していた，あるブロック中で強調されずに残ったテキストである説明文の集合も，その多くが (iii) で除去される.

4.　見出し構造を利用した Web ページ検索結果のリランキング手法

本章では，各ページから抽出された見出し構造を利用した，Web ページ検索結果のリランキング手法を提案する.

4.1　見出し構造を考慮するランキングの意義

まず，見出し構造を考慮するランキングの意義を述べる.

一般の Web ページは複数の話題を含むことがある．ユーザは，目的の話題に関するページを見つけるため，目的の話題に関する検索語をいくつか，テキスト検索システムに入力する．ここで，ユーザが複数の検索語を入力したとする．通常の検索

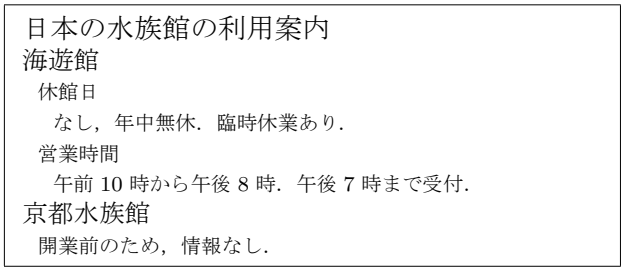

図 3　複数の話題を含む Web ページの例．

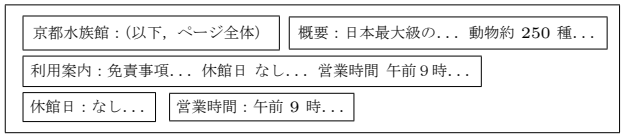

図 4　図 2 の見出し構造から，各ブロックの全ての内容を抽出して得られたドキュメント集合の一部．

システムは，それらの語が一つの話題に関する部分に出現しているか否かは考慮せず，単純に全ての検索語が出現するページを抽出する．しかし，複数の検索語はページ中で互いに異なる話題に関する部分に出現し，論理的関係を持たないかもしれない．複数の話題を考慮しない検索には，このような問題がある．

具体例のため，複数の話題を含む Web ページの例を図 3 に示す．ユーザが「京都水族館」・「営業時間」の二つの検索語を検索システムに入力したとする．この場合，ユーザは「京都水族館の営業時間」に関するページを探していると考えられる．図 3 のページは全ての検索語を含むが，ユーザが意図した情報を含んではおらず，システムのこのような動作は不適である．

複数の話題を含む Web ページをも対象とする，適切な検索のためには，ページ中で目的の話題に関する部分の特定が必要である．ページ中で，ある話題に関する部分とは，本研究が定義するブロックに相当する．ある話題に関しては，一つのブロックでまとめて述べられていることが多いと考えると，目的の話題に関するブロックは，ページ中のブロックを単位とする抽出・ランキングによる，第一位のブロックとして特定できると考えられる．そこで，各ページについて，まずブロックを単位として抽出・スコアリングし，その最高スコアをページのスコアとする手法が考えられる．このような抽出・スコアリングの単位の具体例として，図 2 の見出し構造から，各ブロックの内容を抽出して得られたドキュメント集合の一部を，図 4 に示す．ただし，ここでドキュメントとは検索の単位である．

一般に，検索システムがドキュメントの索引を作成することを索引付けという．検索システムの多くの実装は，効率化のため，検索語の入力以前に索引を作成しておき，それをスコアリングに使用する．そのため，Web ページ単位で索引付けし，ブロック単位でスコアリングするのは非効率的である．実装においては，ブロック単位で索引付けを行うのが効率的である．以下，本稿では，このような実装を行うものとして話を進める．

本節で述べた単純なブロック単位の索引付けには，三つの点で議論の余地がある．第一に，上位のブロックの見出しを考慮しない．このため例えば，図 4 中の営業時間ブロックのみからは，それが京都水族館の情報であることが不明である．第二に，見出しとその他の部分を区別せず，単純に結合する．第三に，

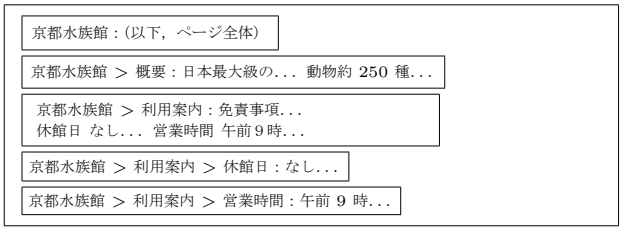

図 5　図 4 中のドキュメントに，それぞれ間接的な見出しを付加した例．

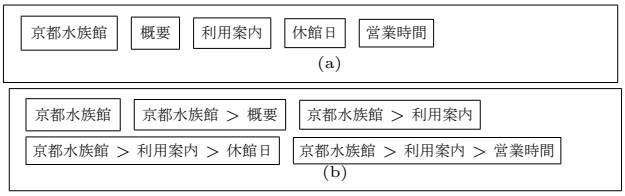

図 6　図 2 の見出し構造から，見出しのみを抽出して得られたドキュメント集合の一部．(a) は見出しの継承を考慮しない例，(b) は考慮する例．

あるブロックについて，下位のブロックを含め索引付けする．このため例えば，前述の「京都水族館」・「営業時間」の二つの検索語に対して，図 3 のページ全体を表すブロックが誤って抽出されてしまう．以下，これらの点について順に議論する．

4.2　継承の仮説と索引付け

2.2 節の継承の仮説によれば，人間は，あるブロックの内容に，その見出し中の語を補うことがある．ブロックはその直接の見出しを含むため，前節で提案した手法によっても，各ドキュメントは直接の見出しを含む．しかし，ブロック間の包含により，各ブロックには上位のブロックの見出しが階層的に存在することもある．索引付けの際には，それら**間接的な見出し**を考慮するべきである．そこで，Web ページ全体の見出しから，注目するブロックの一つ上位のブロックの見出しまで，全ての間接的な見出しを，注目するブロックの内容に加えて索引付けする手法が考えられる．具体例として，図 2 の見出し構造から，各ブロックの内容を抽出し，それぞれに全ての間接的な見出しを加えて得られたドキュメント集合の一部を，図 5 に示す．

4.3　見出しのみの索引付け

節 2.2 の取捨の仮説によれば，人間は，見出しを閲覧することで，対応するブロックを読む必要性をある程度判断できる．つまり見出しには，対応するブロックを取捨選択するために必要な情報は一通り含まれていることが多い．また簡潔な記述のため，見出しには対応するブロックの内容を表す上で重要な語が優先的に配されると考えられる．これらのことから，見出しのみを索引付けする手法が考えられる．この手法により得られるドキュメント集合の具体例を図 6 に示す．(a) は見出しの継承を考慮しない例，(b) は考慮する例である．節 2.2 で，人間が「京都水族館の営業時間」の情報を得る例を挙げた．図 6(b) からも，少なくとも人間は，元のページ中に「京都水族館の営業時間」の情報が書かれていることを読み取れるであろう．このように，見出しのみの索引付けには有用である可能性がある．

4.4　包含するブロックを除いた索引付け

ブロックとはある話題に関する部分である．その下位のブロックは，別のブロックであり，その話題は元の話題と同一と

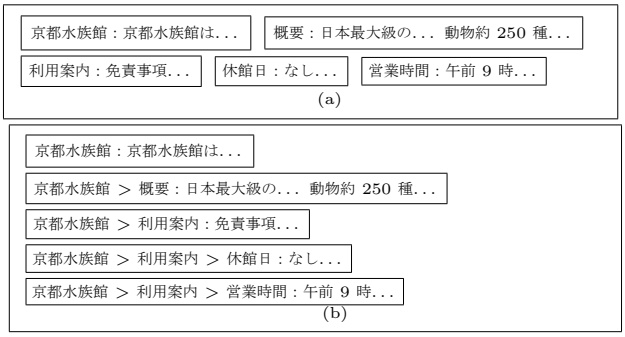

図 7　図 2 の見出し構造から，各ブロックの内容を抽出し，下位のブロックの内容を除いて得られたドキュメント集合の一部．(a) は見出しの継承を考慮しない例，(b) は考慮する例．

表 1　本稿で提案する見出し構造からのドキュメント集合の生成手法

見出しの継承 ＼ ブロックの内容	見出し	全体	補ブロック
考慮しない	HE	HEPS	HEPSe
考慮する	iHE	iHEPS	iHEPSe

は限らないが，一方，元のブロックの一部でもあり，元の話題に含まれない話題も含まない．そこで下位のブロックとは，元の話題の一部に絞って述べている部分であると仮定し，その話題を元の話題の**部分話題**と呼ぶことにする．あるブロックのうち，下位のブロックを除いた部分，すなわち**補ブロック**に注目する．補ブロックはどの部分話題にも属さない事柄，すなわちそれ自体が一つの話題である**補話題**に関する部分であると考えられる．節 4.1 の通り，ブロックを単位とする索引付けの意義とは，Web ページ中の各話題に関する部分をそれぞれスコアリングすることである．しかし，節 4.1 において述べた，ブロックが含む全てのテキストの索引付けによっては，補話題に関する補ブロックが，独立して索引付けされないという問題がある．

そこで，各ブロックについて，その補ブロックのみを索引付けする手法が考えられる．ただし，下位のブロックを包含しないブロックについては，その全体が補ブロックであると考える．この手法により得られるドキュメント集合の具体例を図 7 に示す．(a) は見出しの継承を考慮しない例，(b) は考慮する例である．実際には，節 4.1 の手法により各ブロックを索引付けするよりも，本節の手法によりページを補話題または部分話題に関する部分に重複なく分割し索引付けする方が，適当な粒度のドキュメントが得られることも多い．例えば図 3 のページをこのように分割すると,「京都水族館」・「営業時間」の二語を共に含むドキュメントは得られず，節 4.1 の誤った抽出も起こらない．

4.5　見出し構造からのドキュメント集合の生成手法

本章ではここまで，継承に関しては考慮する・しないの 2 通り，索引付けするテキストに関しては見出しのみ・全体・補ブロックのみの 3 通りの手法を提案してきた．継承の考慮とテキストの絞り込みは独立に施せる改善案であり，提案手法は表 1 の 6 通りとなる．以下，各手法をこの表中の名称で示す．

4.6　複数のランキングの組み合わせ

Cai ら [2] は，本研究と同様，ブロックを考慮したリランキングを行っている．当該論文においては，ブロックを考慮したランキングを，Web ページを単位とする通常のランキングと組み合わせ，第三のランキングを作成すると，その適合率は上昇することが示されている．当該論文におけるランキングの組み合わせは，以下の単純な順位の重みつき和により，この値が小さい順にページを整列し第三のランキングを得る．

$$\alpha \cdot rank_{DR}(q,p) + (1-\alpha) \cdot rank_{BR}(q,p) \quad where \quad 0 \leqq \alpha \leqq 1$$

ここで，q はクエリ，p はページである．$rank_{DR}(q,p)$ は q，p の対のページ単位での順位，$rank_{BR}(q,p)$ は q，p の対のブロックを考慮した順位である．α はパラメータである．本稿においても，この組み合わせ手法を用いる．

5.　評　　価

本章では，Web ページ検索結果を提案手法によりリランキングし，その前後の精度の比較による評価を行う．

5.1　基本的な評価手法

Web ページを単位とするランキングは，Text REtrieval Conference(注2) (以下 TREC) ワークショップの Web トラックで盛んに評価されており，TREC が提供するデータセットや，TREC における評価尺度は一般に用いられている．そこで本稿でも，提案手法によるランキングを，TREC のデータセットや評価尺度により評価する．評価手法は以下の手順から成る．

（1）　文書セット（Web ページの集合）P 中の全てのページを検索システムで索引付けする．

（2）　クエリセット中の各クエリ q について : (a)P 中のページを q に関して抽出・スコアリング・ランキングし，q に関する各ページ p の順位 $rank_{DR}(q,p)$ を得る．これをベースラインとする．ただし，q が含む全ての検索語を含むページのみを抽出する AND 検索を行う．(b)$rank_{DR}(q,p)$ が上位 100 件の p から，提案手法に基づきドキュメント集合 $d(q,p)$ を生成する．(c) 全ての p から得られた $d(q,p)$ を結合して，q に関する全ドキュメントの集合 $D(q)$ を得，$D(q)$ 中の全ドキュメントを検索システムで索引付けする．(d)$D(q)$ 中のドキュメントを q に関して抽出・スコアリングする．ただしページの抽出と同様，AND 検索を行う．(e)p のスコアとして，そこから生成されたドキュメントの最高スコアを用いて p をランキングし，q に関する各 p のブロックを考慮した順位 $rank_{BR}(q,p)$ を得る．

（3）　各 q に関するランキングを結合して，クエリセット全体に関する出力 $result_{DR}$，$result_{BR}$ を得る．

（4）　パラメータ α の値を変化させ，ある α の値に対する組み合わせによる出力 $result_{BR+DR}(\alpha)$ を得る．

（5）　各出力に関する評価尺度の値を算出し，比較する．

索引付けするテキストは，各手法が索引付けの対象とする部分的木構造からテキストノードを抽出し，それらを結合して得る．ただし，ノードの境界では語も切れるものと考え，ノードの境界にはスペースを補う．また，スタイルシートを直接記述するための style ノードの子孫と，スクリプトを直接記述するための script ノードの子孫のテキストノードは除く．

検索システムは，Apache Solr(注3) 4.0.0 を使用する．ドキュメントを索引付けするフィールドの型は，デフォルトのスキー

(注2) : http://trec.nist.gov/
(注3) : http://lucene.apache.org/solr/

マに存在する *text_en* とする．これは，英語の文書セットを用いるためである．ドキュメントやクエリに対する，字句解析やストップワード，ステミングの処理はこれに準ずる．

スコア関数は，以下の式で表される BM25 [3] を用いる．

$$\mathrm{BM25}(q,d) = \sum_{t \in q} \mathrm{IDF}(t) \frac{\mathrm{TF}(t,d)}{k_1 \left\{ (1-b) + b \frac{l_d}{avl_d} \right\} + \mathrm{TF}(t,d)}$$

$$\mathrm{IDF}(t) = \log \frac{n - \mathrm{DF}(t) + 0.5}{\mathrm{DF}(t) + 0.5}$$

ここで，q はクエリ，t は検索語，d はドキュメントである．$\mathrm{TF}(t,d)$ は d 中の t の出現回数である．l_d は d 中の語数である．avl_d は索引中の全ドキュメントの l_d の平均であり，n は全ドキュメントの数である．$\mathrm{DF}(t)$ は t の文書頻度であり，t を含むドキュメントの数である．また，k_1，b はパラメータであり，本稿ではそれぞれ通常用いられる 2.0，0.75 である [3]．

複数のランキングの組み合わせのパラメータである α は，0 と 1 を含む 0.1 刻みの 11 段階で変化させるものとする．上記の手順で評価を行うと，Web ページを単位とする検索によっては抽出され順位が付くが，ブロックを考慮する検索によっては抽出されないページが存在し得る．図 3 のページの，節 4.4 の手法による索引付けがその例である．このようなページに対しては，組み合わせの式の値は未定義である．本稿では単純に，どちらの検索によっても抽出され順位が付いたページのみを，第三のランキングに含める．この処理において，α が 0 である場合，第三のランキングにおけるページの順位はブロックを単位とする検索における順位に等しい．一方，α が 1 である場合，第三のランキングにおけるページの順位はページを単位とする検索における順位，すなわちベースラインに等しい．

本稿では提案手法について，既存手法との比較実験と，より新しいデータセットを用いた実験を行う．用いるデータセットと評価尺度については，各実験に関する節で述べる．

5.2 既存手法との比較

本節では，提案手法を既存手法と比較する．Cai らは，Web ページ中の余白を検出してページをブロックに切り分け，余白の幅に基づきブロック間の距離を推定する VIPS [4]，および VIPS と重複を許す固定語数の分割を併用した CombPS [2] を提案した．Cai らは，TREC2001 Web トラックで用いられたデータセットと評価尺度を用いて，これらの手法を評価している [2]．本節では，これと同様の評価を行い，結果を比較する．

5.2.1 データセット

TREC2001 Web トラックにおいて使用された文書セットは，WT10g ドキュメントコレクションである．これは英語の Web ページのみから成る文書セットで，文書数は約 170 万である．当該のトラックでは，この文書セットに対し，50 クエリとその正解セットを公開しており，本稿ではそれらを用いる．

5.2.2 評 価 尺 度

Cai らに倣い，Precision@10（以下 P@10）を用いる．P@10 とは，ランキングの上位 10 件のうち，適合するドキュメントの割合である．P@10 は適合率の尺度であるため，出力されるドキュメントが 10 件に満たないクエリに対しては，その中での適合するものの割合を求める．複数のクエリに関する評価を総合するためには，各クエリに対する P@10 の値を平均して用いる．このとき，出力されるドキュメントの存在しないクエリは，単に平均値の計算から除く．

表 2　TREC2001 のデータセットを用いた，手法ごとの P@10 の値の比較

Method	BR only	BR+DR best
Baseline (with Okapi)	.312	.312
VIPS	.316	.328
CombPS	**.326**	**.338**
Baseline (with Solr)	.271	.271
HE	.278	.298
iHE	**.333**	**.333**
HEPS	.247	.284
iHEPS	.247	.280
HEPSe	.257	.286
iHEPSe	.269	.303

5.2.3 評価に関する既存手法との相違点

Cai らの評価手法は，検索システムとして Okapi，スコア関数として BM2500 を用いている点が，本稿のそれと異なる．

また，後述の結果において，Cai らが測定したベースラインの P@10 の値が，本稿において新たに測定したそれと大きく異なる．これは，本稿で AND 検索を用いた箇所で OR 検索を用いたことによるものであると思われる．OR 検索とは，複数の検索語を含むクエリに対して，検索語が 1 つでも含まれているドキュメントを抽出する方法である．実際，本稿で用いる検索システムと OR 検索によるベースラインの P@10 の値は，0.310 となり，Cai らの値とほぼ等しい．しかし，多くの実際の Web 検索エンジンは，複数の検索語に対してデフォルトでは AND 検索を行うため，本稿では AND 検索を用いた．

5.2.4 結　　果

結果を表 2 に示す．表の上部は Cai らによる部分であり，下部は本稿で新たに測定した部分である．組み合わせについては，P@10 の値を最大化する α の値を設定した際の値を示した．

5.2.5 考　　察

ランキングの組み合わせを行わない場合，継承を考慮しつつ見出しのみを用いる iHE が，唯一 Okapi を用いたベースラインを上回っている．しかし，ブロックを考慮したランキングと，Web ページ単位でのランキングの組み合わせによっては，CombPS が iHE を上回っており，iHE がそれに続いている．

既存手法との比較　ランキングの組み合わせを行わない場合，提案手法の iHE は他の手法を上回っている．一方，組み合わせを行う場合，既存手法の CombPS がわずかに iHE を上回っている．組み合わせを行えば，これらの手法にはほぼ差がないと言える．つまり，継承を考慮した見出しのみの索引付けには，一定の有用性があることが示された．これは，見出しはブロックの内容の要約であり，その内容のみからブロックを読む必要性をある程度判断できるという仮説を裏付けるものである．他の提案手法は，組み合わせを行う場合でも Okapi を用いたベースラインを下回っており，少なくともこのデータセットに対しては有用ではないことが示された．WT10g 中の文書の中央長は 3.3KB と小さいため，見出しによる大まかな構造化よりも，

余白による細かな構造化の方が多用されており，後者を抽出する VIPS が有効に働いたと考えられる．また文書数も近年の文書セットに比べ少ないため，特に見出しに検索語が出現するページの数も少ないと思われる．実際，HE に対する 35 クエリ，iHE に対する 34 クエリで出力されるページが存在しなかった．

見出しの継承に関する考察　HE と iHE の間には，継承を考慮するか否かにより，P@10 の値に差がみられる．見出し語には，対応するブロックの内容を端的に表現するため，ブロックにおける重要語が充てられると考えられる．しかし単独の見出しは短いテキストであるため，検索語を全て含むことは少ない．これに対して，iHE により得られたドキュメントは，複数の継承関係にある見出しを結合したものであるため，単独の見出しと同様に重要語から成り，かつ単独の見出しに比べて長いテキストである．そして継承関係により，それらの重要語の間には論理的な関係があることが保障されている．このため，単独の見出しを用いる HE よりも，継承を考慮する iHE の方が優位であると思われる．他の手法の対の間には，ドキュメントに占める見出し語の割合が比較的低いためか，差はみられない．

ブロックの内容に関する考察　ブロック全体のテキストを用いる手法の P@10 の値がやや低くなっている．ブロックの全体を用いる手法と，下位のブロックを除去する手法との間では，後者の P@10 の値が高いため，少なくともこのデータセットでは，補ブロックも独立した話題を持つ場合が多いと考えられる．

組み合わせに関する考察　全ての手法で，組み合わせの前後で P@10 の値は上昇しているか等しくなっているが，組み合わせの式により，$\alpha = 0$ の場合の P@10 の値は元の値と等しくなるため，これは当然である．iHE については，組み合わせの前後で P@10 の値が等しい．つまり，組み合わせを行わない場合の性能が最もよい．また，組み合わせの以前から P@10 の値が比較的高い VIPS や CombPS については，組み合わせによる値の上昇は僅かである．組み合わせの以前から手法の性能が高い場合，組み合わせの効果は薄いと考えられる．

5.3 より新しいデータセットへの適用

前節 5.2 で用いた文書セットは，1997 年にクロールされたものであり，近年の Web をよく表してはいない．本節では提案手法を，2009 年にクロールされた ClueWeb09 ドキュメントコレクションに適用し評価する．当該の文書セットは，TREC2009 から 2012 までの Web トラックで使用されている．

5.3.1 データセット

ClueWeb09 ドキュメントコレクションは巨大な文書セットであり，TREC ではサブセットである ClueWeb09 Category B ドキュメントコレクションを用いた評価も認められている．これは英語の Web ページのみから成る文書セットで，文書数は約 5 千万である．TREC2009 から TREC2012 までの Web トラックでは，この文書セットに対して，それぞれ 50 クエリとその正解セットが公開されており，本稿ではそれらを用いる．

5.3.2 評 価 尺 度

TREC2010 の Web トラックより評価尺度として用いられている Expected Reciprocal Rank [5]（以下 ERR）を用いる．ランキングの上位 k 件に関する ERR の値は，以下の式による．

表 3　TREC2009，2010，2011，2012 のデータセットを用いた，手法ごとの ERR@20 の値の比較

	TREC '09		**TREC '10**		**TREC '11**		**TREC '12**	
	BR	**BR + DR**	**BR**	**BR + DR**	**BR**	**BR + DR**	**BR**	**BR + DR**
Baseline	**.144**	.144	**.068**	.068	.086	.086	.135	.135
HE	.119	**.161**	.064	.064	.093	.096	.106	.143
iHE	.136	.160	.056	.056	.093	**.108**	**.186**	**.186**
HEPS	.129	.147	.065	.073	.082	.085	.106	.133
iHEPS	.133	.150	.066	.070	.087	.091	.143	.148
HEPSe	.123	.153	.056	**.075**	.092	.090	.117	.141
iHEPSe	.139	.155	.061	.070	**.094**	.094	.128	.134

$$\mathrm{ERR@}k = \sum_{i=1}^{k} \frac{R(g_i)}{i} \prod_{j=1}^{i-1} \{1 - R(g_j)\}$$

ここで g_i とは，ランキングで i 番目のドキュメントの，TREC が定める適合度であり，0 から 4 までの整数をとる．$R(g)$ は正規化された適合度であり，$R(g) = \frac{2^g - 1}{16}$ である．より適合度の高いドキュメントが，より上位に表れるほど，ERR の値は大きくなり，その後のランキングが ERR の値に与える影響は小さくなる．本稿では，TREC2010，2011 に倣い，$k = 20$ とする．複数のクエリに関する評価の総合は項 5.2.2 と同様に行う．

5.3.3 結　　果

結果を表 3 に示す．組み合わせを行う場合については，ERR@20 の値を最大化する α の値を設定した際の値を示した．

5.3.4 考　　察

全体的に iHE の性能がよい．ランキングの組み合わせを行う場合の 3 つのデータセット，および行わない場合の，より新しい 2 つのデータセットにおいて，iHE によって最高かそれにほぼ等しい ERR@20 の値が得られている．TREC2010 のデータセットにおいては他とは異なる傾向がみられる．ベースラインの ERR@20 の値が 4 つのデータセット中で最小であることから，TREC2010 のデータセットは検索の難しいデータセットであると考えられ，そのことが結果に影響した可能性がある．また，HE に対する TREC2010 データセット中の 2 クエリ・TREC2011 データセット中の 8 クエリ，iHE に対する TREC2011 データセット中の 5 クエリを除き，全ての手法とクエリの対に関して少なくとも 1 ページが出力された．

見出しの継承に関する考察　TREC2009，2011，2012 のデータセットに関しては，継承を考慮する手法が考慮しない手法をおおむね上回っている．しかし TREC2010 のデータセットに関してはあまり差がない．TREC2010 のクエリ集合は，ストップワードを除き一つの検索語から成るクエリの割合が 0.52 と高く，TREC2009 における 0.36 や TREC2012 における 0.26 を大きく上回っている．また，TREC2011 のクエリ集合中のクエリは全て複数の検索語から成っている．短いテキストである単独の見出しであっても，全ての検索語を含む場合が多ければ，継承を考慮する利点は薄れる．継承の考慮は，複数の検索語から成るクエリに対して有効に働くものと考えられる．

ブロックの内容に関する考察　ブロックの内容と ERR@20 の値との関係は，データセットによりまちまちである．しかし

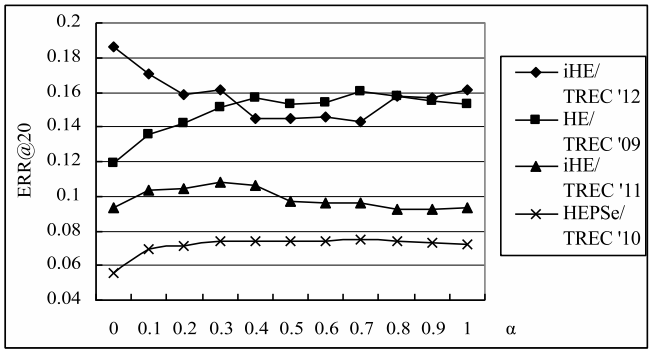

図 8　α の値による ERR@20 の値の変化

iHE については，前述の傾向のある TREC2010 のデータセットを除き，組み合わせの有無によらず ERR@20 の値が高い．ERR の定義によれば，継承を考慮した見出しのみによるランキングを行った場合，ランキングの上位を見るだけで適合度の高いドキュメントが得られる可能性が高いと考えられる．

組み合わせに関する考察　α の値と ERR@20 の値の関係を図 8 中のグラフに示す．このグラフでは，各データセットにおいて，組み合わせを行った場合に最も ERR@20 の値が高くなる手法についてのデータを示した．単調増加に近い形のグラフはなく，組み合わせによる ERR@20 の値の上昇とは，単に α の値が大きい場合の，ベースラインの精度に引かれた上昇ではないことがわかる．TREC2009，2011 に関する 2 つのグラフの形は山形に近く，データセットによっては，ブロックを考慮したランキングとベースラインを適当な比率で組み合わせた場合に，検索の精度を向上させる効果があることが分かる．しかし TREC2012 のデータセットにおいては，ブロックを考慮したランキング単体の性能が最も高い．このデータセットにおいては，ブロックを考慮したランキングの ERR@20 の値の，ベースラインからの伸びも大きく，やはりブロックを考慮したランキングの精度が元々よい場合，組み合わせの効果は薄いといえる．ERR@20 の値を最大化する α の値は，TREC2009 のデータセットにおいては 0.7，TREC2011 のデータセットにおいては 0.3，そして TREC2012 のデータセットにおいては 0 であり，適当な α の値の推定も課題であることも明らかになった．

6. 関 連 研 究

Cai らは本研究と同様，Web ページを階層的なブロックに分割し，ブロックを考慮したランキングにより Web ページ検索の精度を向上させている [2]．加えて，ブロックレベルでのリンク解析も行っている [6]．また Mizuuchi ら [7] は，リンク先のページではリンク元のページの内容は省略されることがあるという仮説を立てており，これはリンク構造における継承について述べているといえる．そして Web ページ中の見出し構造の抽出とは，Web ページをよく構造化された文書に変換することでもあるが，構造化文書である XML 文書に対してキーワード検索を行うための研究 [8] [9] [10] [11] は広く行われており，今後それらの提案する手法の導入・本研究の提案手法との比較は検討すべきである．特に田邊ら [12] は，データ指向の部分を文書指向の部分と区別した，XML のキーワード検索手法を提案している．Web ページ中の見出しはデータ指向の部分であると考えられ，当該手法の本研究の提案手法との関係は深い．

7. 結　　論

本稿では，階層的見出し構造に着目した Web ページ検索手法を提案し，評価した．本稿の貢献は以下のように整理される．

- 見出し構造を詳細に定義した．
- 見出し構造を利用した，Web ページ検索の精度向上手法を提案した．また提案手法は，近年の Web において，Web ページ検索の精度向上のために有用であることを示した．
- 見出しはブロックの内容の非常に簡潔な要約であり，ブロック中での重要語が配されるという仮説の正当性を示した．
- 間接的に，前著 [1] で提案した Web ページ中の見出し構造の自動抽出手法の有用性を示した．

見出し構造の利用価値は高く，本稿における Web ページ検索の精度の向上は，その応用の一つでしかない．今後，Web ページをはじめ，構造化文書は増加すると予測される．その中で，見出し構造の抽出と利用の果たすべき役割は大きい．

文　　献

[1] 真鍋, 田島 : “Web ページ中の階層的見出し構造の発見”, 第 4 回データ工学と情報マネジメントに関するフォーラム (DEIM2012) F4–4 (2012).
[2] D. Cai, S. Yu, J.-R. Wen and W.-Y. Ma: “Block-based web search”, SIGIR ’04, pp. 456–463 (2004).
[3] J. Pérez-Iglesias, J. R. Pérez-Agüera, V. Fresno and Y. Z. Feinstein: “Integrating the Probabilistic Models BM25/BM25F into Lucene”, CoRR, **abs/0911.5046**, (2009).
[4] D. Cai, S. Yu, J.-R. Wen and W.-Y. Ma: “Extracting content structure for web pages based on visual representation”, Proceedings of the 5th Asia-Pacific web conference on Web technologies and applications, APWeb’03, Berlin, Heidelberg, Springer-Verlag, pp. 406–417 (2003).
[5] O. Chapelle, D. Metlzer, Y. Zhang and P. Grinspan: “Expected reciprocal rank for graded relevance”, Proceedings of the 18th ACM conference on Information and knowledge management, CIKM ’09, pp. 621–630 (2009).
[6] D. Cai, X. He, J.-R. Wen and W.-Y. Ma: “Block-level link analysis”, SIGIR ’04, pp. 440–447 (2004).
[7] Y. Mizuuchi and K. Tajima: “Finding context paths for web pages”, Hypertext ’99, pp. 13–22 (1999).
[8] N. Fuhr and K. Grosjohann: “XIRQL: a query language for information retrieval in XML documents”, Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR ’01, pp. 172–180 (2001).
[9] K. Hatano, H. Kinutani, M. Yoshikawa and S. Uemura: “Information retrieval system for XML documents”, Proceedings of the 13th International Conference on Database and Expert Systems Applications, DEXA ’02, pp. 758–767 (2002).
[10] S. Pradhan: “An algebraic query model for effective and efficient retrieval of xml fragments”, Proceedings of the 32nd international conference on Very large data bases, VLDB ’06, pp. 295–306 (2006).
[11] M. Theobald, R. Schenkel and G. Weikum: “TopX: Efficient and versatile top-k query processing for semistructured data”, VLDB Journal, **17**, (2008).
[12] 田邊, 清水, 吉川 : “XML キーワード検索における要素の特性を考慮した検索結果の構築”, 第 4 回データ工学と情報マネジメントに関するフォーラム (DEIM2012) E11–1 (2012).