情報処理学会論文誌　データベース　Vol. 4　No. 1　1–13 (Mar. 2011)

有益な検索結果提示のための部分文書再構成手法の提案

欅　　惇志[†1]　波多野　賢治[†2]　宮崎　　純[†3]

XML 文書を対象とした情報検索では，文書単位よりも細かな粒度の部分文書を対象とした検索を行うことが可能である．従来の部分文書に対する検索技術では，各部分文書に対するクエリへの適合度の算出方法に焦点を当てているため，各適合度すなわちスコアリング結果から，検索結果である部分文書 1 つ 1 つをどのように構成するのかということに関しては十分に議論されてこなかった．しかしながら，各部分文書の持つスコアのみに依存した形での検索結果の提示方法では，1) 大きすぎる（不要な部分を含む）粒度の部分文書が抽出される，2) 適合部分文書の一部しか抽出できない，といった問題が起こりうる．これらの問題を解決するため，我々は各部分文書のテキストサイズと部分文書間の包含関係を考慮した適合部分抽出手法を提案する．さらに，先祖や子孫の部分文書の統計量を考慮したスコアリング手法を提案し，情報要求に強く合致する部分文書を検索結果上位にランキングすることを目指す．評価実験の結果，提案手法は従来手法と比較して約 8%検索精度が改善された．

A Proposal of a Reconstruction Method to Return Well-informative Search Results

Atsushi Keyaki,[†1] Kenji Hatano[†2] and Jun Miyazaki[†3]

We propose a method for identifying appropriate granular fragments for user information needs and obtaining more accurate search results in XML fragment search. Existing approaches simply generate a ranked list in descending order of each XML fragment's relevance to a search query. These approaches have problems, i.e., they may extract irrelevant fragments and overlook more relevant fragments. To address these problems, we generate a refined ranked list through two steps. First, we extract and reconstruct relevant fragments considering the sizes of XML fragments and relationships among XML fragments in a simple ranked list. Second, we score these XML fragments with useful statistics of its descendant/ancestor XML fragments. Our experimental results show that our method improves search accuracy by 8% compared with simple BM25E which neither reconstruct XML fragments nor use some kinds of statistics.

1. はじめに

構造化文書の 1 つである XML[⋆1]（Extensible Markup Language）はデータ交換の標準フォーマットとして広く利用されている．そのため，XML で記述された文書は現在までに数多く産出されており，今後ますます多くの XML 文書が作成されると考えられる．そのような膨大な数の XML 文書から，ユーザの要求を満たす情報を取得することは困難を極め，必然的に XML 文書のための情報検索技術に対する需要はますます高くなると考えられる．このような背景から，XML 文書に対する情報検索技術を開発することは非常に重要である．

XML 文書はいくつかの点でテキスト文書とは異なるが，その最たる特徴はその内部が構造化されているという点である．XML 文書は階層的な構造を持ち，文書中では開始タグと終了タグで囲まれたテキストごとにメタ情報を付与することが可能となる．XML 検索においては，それぞれのタグで囲まれた部分を 1 つの検索単位として扱っている．本論文ではこれを**部分文書**と呼んでいるが，この部分文書は同一文書から複数個取り出される性質上，互いに入れ子状の包含関係を持ち，それらに含まれるテキストノードに重複が発生する[⋆2]．

テキストノードに重複が存在する複数の部分文書をユーザに提示した場合に，ユーザがすべての検索結果を確認した場合には同じテキストが繰り返し提示されることになるが，すでに 1 度確認した内容を再度提示することがユーザにとって適切であるとは考えられない．このような理由から，部分文書検索の検索結果をクエリへの適合度の降順に並んだリスト形式で提示する際には，部分文書間に存在する重複を取り除かなければ検索精度[⋆3]が低下すると報告されている[1)]．したがって，先行研究の多くで重複を含まない結果を提示している[2)]．一般的に，検索結果はそれ単体で意味が通じる，もしくは，ある程度内容のまとまった箇所を提供する必要があるといわれている[3)] ため，十分に有益な部分文書を検索結果として選別する必要がある．本論文における有益な部分文書とは，クエリに対して適合度の高い文

†1 同志社大学大学院文化情報学研究科
Graduate School of Culture and Information Science, Doshisha University
†2 同志社大学文化情報学部
Faculty of Culture and Information Science, Doshisha University
†3 奈良先端科学技術大学院大学情報科学研究科
Graduate School of Information Science, Nara Institute of Science and Technology

⋆1 http://www.w3.org/TR/REC-xml/
⋆2 部分文書の概要とその重複に関しては 2.2 節で詳述する．
⋆3 文書検索における検索精度とは抽出された文書中の適合文書の割合で表されることが多いが，部分文書検索では抽出された部分文書中の適合箇所の割合から検索精度を算出する．

書中の，部分文書中に含まれる非適合箇所が少なく，かつ，適合箇所を多く含む部分文書のことを指す．また，部分文書の構造的な大きさを粒度と表現する．先行研究においては重複が発生した際，最も高いスコアの部分文書が抽出されるのみであるため，それ以外の部分文書は無条件で検索結果から破棄されることになる．このような提示方法では，ユーザにとって有益な検索結果を提示するという目的に対しては必ずしも適切であるとは限らない．なぜなら，検索結果として提示された部分文書に含まれるテキストがクエリに対する適合箇所を含んでいたとしても，仮に極端に小さなサイズの部分文書であった場合にはそれ単体で意味が通じないという問題が起こる．また，多くの情報検索技術ではクエリキーワードが出現する箇所を重要視し，局所的にクエリキーワードが頻出する箇所に高いスコアが付与される傾向があるが，それに対してユーザが情報を欲する場面においては，クエリキーワードが出現する箇所というだけでは有益な検索結果，すなわち情報要求に合致する最適な部分文書の条件としては不十分である．

そこで本論文では，適合する文書中の適切な粒度の部分文書を検索結果として抽出するために，部分文書のテキストサイズと重複関係を考慮した部分文書の再構成方法を提案する．さらに，各文書中から適切な粒度の部分文書を特定した後には，情報要求に対して，より合致する部分文書を特定することを目指す．その際，突出して高い統計量を持つ箇所を持つ部分文書や，情報に富んだ文書中に含まれる部分文書が条件を満たしていると考え，先祖や子孫の部分文書の統計量を考慮して該当する部分文書の発見を目指す．これらをふまえ，より精錬された検索結果を提示する．

以下，2 章，3 章ではそれぞれ XML 情報検索の基本事項と関連研究について，4 章では提案手法について述べる．また，5 章では提案手法の有効性を確認するために行った評価実験について述べ，6 章ではまとめと今後の課題について述べる．

2. XML 情報文書検索に関する基本事項

本章では，文書検索と部分文書検索の比較，部分文書の概要とその重複関係，そして XML 情報検索に関する研究の歴史について述べる．

2.1 文書検索と部分文書検索の比較

XML 部分文書文書検索と，一般的な Web 検索システムなどをはじめとした文書検索の差異について説明する．多くの Web 検索システムはクエリに適合する文書のリストを提示する際に，スニペット[3] と呼ばれる 150 文字前後の要約文もあわせて提示する場合が多い．スニペットはクエリキーワードとその周辺のテキストを抽出する技術であり，検索結果として提示された文書群からいずれの文書が閲覧するのに適切であるのかをシステム利用者が判断するための要約文である．多くの検索システム利用者がスニペットを利用している反面，文章の文脈を考慮しないために理解不能なスニペットが生成される可能性があるため，必ずしも満足な結果が得られているわけではない[4]．このことから，スニペットのみから情報要求を満たすことはできず，結局のところ文書検索システム利用者は文書を閲覧し，必要な情報を自ら発見しなければならない．

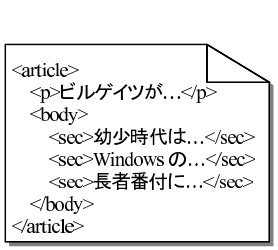


図 1　XML 文書
Fig. 1　XML document.

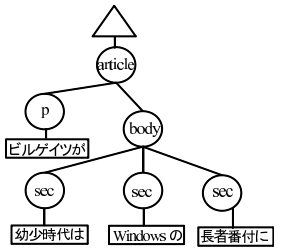


図 2　XML 木
Fig. 2　XML tree.

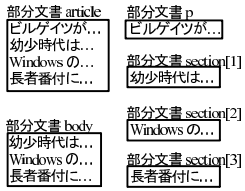


図 3　部分文書
Fig. 3　XML fragment.

これに対して，XML 検索における最大の関心は，クエリに適合する部分文書を抽出し，それらに順位付けを行い提示することである．多くの Web 検索システムがクエリに適合する文書のリストを提示するのに対して，XML 検索システムはクエリに適合する部分そのもののリストを提示することができる．これにより，ユーザは文書中から情報要求を満たす部分を探索する必要がなくなるために，情報検索を行う際のユーザの負担を大きく軽減することができる．

2.2 部分文書とその重複関係

1 章で述べた部分文書の概要と重複について説明するために，図 1〜3 を用いて具体例を示す．まず，**図 1** は XML 文書の例であり，**図 2** は XML 文書を木構造で表現した図である．構造化文書は一般的に木構造で表現することができ，文書構造の視認性の向上を目的としてたびたび木構造で表現される．本論文においても同様に，適宜 XML 文書を木構造と見立てて議論を進めることとする．このとき，XML 文書のそれぞれの開始タグと終了タグが XML 木の各要素ノード名に対応しており，タグの入れ子は要素ノードの親子関係によって表現されている．**図 3** の各部分文書は，図 2 の XML 木の各要素ノード以下に含まれるテ

キストノードと対応する[⋆1]．つまり，文書全体を表す article ノードは子孫に存在するテキストノードすべてを持ち，body ノードは子ノードである 3 つの section ノードに含まれるそれぞれのテキストノードを保持する．包含関係（先祖 · 子孫関係）を持つ部分文書間においてテキストノードの重複が発生するのはこのためである．なお，ある要素ノード（部分文書）に着目した際に，先祖にあたる要素ノードを粒度の大きなノード，逆に子孫にあたる要素ノードを粒度の小さなノードと表現する．

このとき，仮に情報要求を満たす内容が「幼少時代は…」と「Windows の…」，「長者番付に…」であった場合には，ユーザに対して body ノード以下の部分を提示することが適切であり，最も有益な検索結果である．

2.2.1　XML 情報検索に関する研究の歴史

部分文書検索に用いられる情報検索技術は，文書単位を検索粒度としたテキスト文書に対する情報検索技術を部分文書検索用に拡張して用いられることが多い．たとえば，代表的な部分文書用スコアリング手法である TF-IPF[5] は，ベクトル空間モデルの文書検索技術である TF-IDF[6] を XML の経路式を考慮し拡張させたスコアリング手法である．TF-IDF はある文書中の索引語の出現頻度（TF）と全文書集合中での各索引語ごとの文書頻度の逆数（IDF）の積から算出されるのに対し，TF-IPF はある部分文書に含まれる索引語の出現頻度と，XML の経路式ごとに個別に集計された部分文書頻度の逆数（IPF）の積で算出される．索引語数での正規化を行った正規化 TF-IPF[7] などの拡張も存在する．

同様に，確率モデルに基づいた文書検索用スコアリング手法である Okapi's BM25[8] を構造化文書検索用に拡張させた BM25F[9] や，部分文書検索用に拡張させた BM25E[10] なども存在する．BM25F ではタグに重みを付与することで，クエリキーワードの出現箇所ごとに出現に対する重みを調整し，効果的な構造化文書検索[⋆2]を目指している．それに対して，BM25E は TF-IDF から TF-IPF への拡張と同様 XML の経路式ごとに統計量を算出し，BM25F を部分文書検索に拡張している．現在の文書指向型 XML 文書検索に関する研究では，TF-IPF や BM25F（BM25E）をベースに拡張された検索技術が主流である．

部分文書検索の結果は抽出される部分文書間で重複が発生する可能性があるが，いくつかの先行研究では重複が発生することを考慮せずに，クエリ処理によって算出される部分文書の適合度に従ってランキングされた順位付きリストを提示していた．本論文では，このようなクエリ処理によって得られた，重複を排除するために特別な処理を行っていないリストを**重複リスト**と呼ぶこととする．また，文献 1) において重複リストによる検索精度の低下が報告されて以来，多くの研究ではリスト中から重複を排除した**非重複リスト**を用いている[⋆3]．現存する最大の XML 検索のためのプロジェクトである INEX（INitiative for Evaluation of XML retrieval）project[⋆4]の XML 検索用のトラック全般[11] においても非重複リストを使用している[⋆5]．

INEX project は文書指向型 XML 文書検索に関する研究において，最もさかんに研究が行われているプロジェクトであり，部分文書検索の検索精度を計測するためのテストコレクションを作成している．ユーザは情報検索を行う際において「検索結果上位数件のみ確認するといわれている[4]」ことからも，検索結果上位において高い検索精度を実現することが最重要課題である．そのため INEX project では特に再現率が 1%における検索精度である iP[.01] を公式尺度とし，提案システムの評価に用いている．また，INEX project で利用されるその他の評価尺度としては Mean Average interpolated Precision（MAiP）が存在する．MAiP は複数の再現率点における精度の平均から求められ，INEX project においては 101 個の再現率から計算される．

また，効果的な検索を目指すトラックである Ad hoc track では，部分文書単位の検索の効果を確認するため，検索結果として文書全体ではなく適切な粒度の部分文書を提示することを取り組むべき課題の 1 つとしているにもかかわらず，近年は部分文書検索よりも文書検索に関する研究が積極的に取り組まれている．これは，適切な粒度を特定することは困難な課題であり，多少の非適合箇所が含まれたとしてもすべての適合箇所を含む文書全体を検索結果とすることが部分文書検索よりも高精度であると考えられているためである．実際にここ数年間の INEX 公式検索精度上位のシステムの多くは文書検索である[11]．しかしながら，情報検索におけるユーザの負担を軽減させるために，ユーザに対してクエリに対する適合箇所のみを提示することは我々の研究の目標の 1 つであるため，本論文では部分文書単位での検索によって効果的な検索の実現を目指す．

⋆1 厳密には各部分文書にはテキストノード以外にも要素ノードなどのタグも含まれるが，評価ツールによって解釈されるのはテキストノード部分のみであるため，本論文において部分文書とは各ノード以下に含まれるテキストノードを統合した文字列であるとする．

⋆2 たとえば，タイトルや見出しなどに索引語が出現する場合は文書が適合する可能性が高いと見なし，それらのタグの重みを大きくする，などである．

⋆3 重複が発生していなければ，1 つの XML 文書から複数の部分文書を抽出することは制限されていない．

⋆4 http://www.inex.otago.ac.nz/

⋆5 各クエリに対して上限 1,500 件まで，抽出すべき部分文書が存在する限り抽出し提示する．

3. 関 連 研 究

XML 文書には 2 つのタイプが存在する．1 つはデータ指向型 XML 文書と呼ばれ，一般的に 1 つのテキストノード中に単語もしくは複合語が 1 つ含まれており，もう一方は文書指向型 XML 文書と呼ばれ，テキストノード中に文章が 1 つ以上含まれている[12]．いずれの XML 文書においても，XML 文書を木構造へ変換した際に，木全体からクエリに適合する部分木を切り取りユーザへ提示することを目的としている点では同様であるが，XML 検索技術を適用する際にはそれぞれの XML 文書の特性に合わせて異なるアプローチがとられている．我々の研究においては主に文書指向型 XML 文書を扱うが，データ指向型 XML 文書に関する先行研究の中には我々の研究と関連する研究も存在するため，ここでは，両方のタイプの XML 文書に関する研究を関連研究としてあげる．

3.1 データ指向型 XML 文書に対する検索

過去に行われたデータ指向型 XML 文書に関する研究の多くで，キーワード発見のための XML 木探索アルゴリズムが考案され，効率的な検索，すなわち検索速度に関する研究が中心に取り組まれてきた．それらの研究の多くにおいて，Lowest Common Ancestor（LCA）[13] の概念が用いられている．LCA は本来，任意の 2 つ以上のノードの共通祖先ノードを表す語であったが，多くの場合において，すべてのクエリキーワードを含む最も深いノードを表す．LCA を基にした，部分文書（木）検索のための研究は数多く存在するが[14)–16)]，これらの研究においては，LCA を根とする完全部分木もしくは，その完全部分木からさまざまな観点に沿って抽出した部分木をクエリに対する最適部分として扱っている．その結果，効率的な検索を実現するうえでは非常に大きな貢献を残しているものの，効果的な検索，すなわち正確な検索を行ううえでは問題をかかえている．なぜなら，LCA に基づいてキーワードを抽出する手法では単にすべてのクエリキーワードが出現することが担保されているのみであるが，前述のとおりクエリキーワードが出現するというだけでは情報要求を満たす箇所であるかどうか判断できないため，実際にクエリに対する適合部分であるとはいいきれない．その結果，LCA を根とする完全部分木を加工して抽出された部分をクエリに対する最適部分とした場合には検索精度が低下するためである[17]．

このような問題を解決するために，LCA を根とする完全部分木中からクエリに対する適合部分を特定することを目指す研究も存在する．それらの研究の 1 つである XSeek[17] ではキーワード間の関連性を考慮した Meaningful LCA（MLCA）を作成する．MLCA を作成するために，Document Type Definition（DTD）やクエリキーワードが出現する位置情報などを利用して，LCA を根とする完全部分木中からそれぞれの XML タグに対して分析，分類を行い，それらの処理結果を利用してユーザに対して検索結果を提示する．つまり，XSeek ではあらかじめ中間結果として LCA を根とする完全部分木を抽出しておき，中間結果から有用な検索結果を作成するために部分木の再構成を行っているというわけである．

また，eXtract[18] では，MLCA をさらに拡張させて，クエリを解析することでユーザの検索意図を特定し，よりクエリに対して適合する箇所を抽出することを目指している．クエリは 2 種類に分類され，1 つ目が明確な検索対象（解答）が存在する場合のクエリである．たとえば，“富士山の標高は何メートルであるか？” などといった情報要求の際のクエリが相当する．それに対して，もう一方のクエリは明確な検索対象が存在せず，クエリキーワードに関する知識を広く取得したい場合のクエリである．こちらのクエリは，“京都はどのような土地か？” といった情報要求の際のクエリが該当する．eXtract では，クエリの意図を明確にしたうえで，検索結果として最適な粒度の抽出に利用している．

これらの研究の成果からも，検索結果として最適な粒度の決定は，XML 検索においては非常に重要な観点であるといえる．これらの研究と我々の提案手法を比較した場合，ともに文書中からクエリに適合する部分文書（木）を特定し検索結果として提示するという点では目的を同じとする．しかしながら，提案手法は LCA を根とする完全部分木以下にクエリに対して最適な解が存在するとは仮定せず，文書から抽出され得るあらゆる部分文書がクエリに適合する可能性も考慮している．また，DTD のような文書自体以外の外部情報を必要とせずに，部分文書間の関係から文書中から適合箇所を抽出するという点でも異なる．

3.2 文書指向型 XML 文書に対する検索

テキストノード中に文が含まれている XML 文書は文書指向型 XML 文書と呼ばれ，Web 文書として広く利用されている XHTML[⋆1] などがあてはまる．文書指向型 XML 文書に対する検索では，単にクエリキーワードの出現箇所を発見するのではなく，情報要求を満たす部分を選別することを目的とした検索が行われることが多い．つまり，これまでの研究では主に，クエリに対する適合部分文書を効果的に発見するためのスコアリング手法の提案がなされてきた．

部分文書検索の目的はクエリに対する適合箇所そのものを抽出することであるため，文書中の重要部分抽出に類似していると考えられる．したがって，我々は過去の研究[19] において，重要部分抽出技術に関する研究の知見[3] を利用したスコアリング手法を提案した．文

⋆1 http://www.w3.org/TR/xhtml1/

献 3) の重要部分抽出の要件の 1 つである「クエリに関する最大限の情報を盛り込む」を満たすため，我々は既存の情報検索技術を拡張し，クエリが指定する構造から算出される統計量[*1]と，クエリが指定するキーワードから算出される統計量を考慮したスコアリング手法を考案した．前者の統計量を考慮して算出されるクエリ構造スコア（Query Structure score，以降 QS）は，TF-IPF のように XML の経路式ごとに統計量を計算するのではなく，クエリの制約を満たす部分文書全体に対して統計量を算出する．それに対して，後者の統計量を考慮して算出されるクエリキーワードスコア（Query Keyword score，以降 QK）は，部分文書中に存在するクエリキーワードの種類数から算出される．

評価実験により，QS と QK はいずれの手法においても従来の手法と比較し平均検索精度の向上が認められるという結果が得られたものの，これらの手法を用いることで，テキストサイズの大きな部分文書に大きなスコアが付与される傾向があるということが判明したため，情報検索におけるユーザの負担軽減のためには依然として課題を残している．

4. 提 案 手 法

本章では，高精度部分文書検索の実現のために我々が行った取り組みに関して説明する．提案手法では，クエリに対する適合箇所を特定し，より情報要求に合致する部分文書を検索結果上位に提示することを目指す．具体的な処理手順を以下に示す．

処理 (1)　まず最初に，部分文書検索技術を用いて各部分文書に対して初期検索を行い，スコアを計算する（**図 4** (1)）．この処理によって，重複リストを取得する．

処理 (2)　処理 (1) で得られた重複リストから，文書単位でクエリに対する適合部分文書を抽出する（図 4 (2)）．この処理では，抽出される部分文書のテキストサイズによる抽出制限や，部分文書間の各スコアと重複を考慮した部分文書の統合を行う．以後この処理で作成される部分文書集合を**最適部分文書集合**と呼ぶ．

処理 (3)　処理 (2) で作成された最適部分文書集合をリスト形式で提示するために順位付けを行う．その際，有益な検索結果を上位に提示することを目的としたスコアリング手法を提案する（図 4 (3)）．この手順で作成される，文書中の最適粒度の部分文書で構成され，検索結果が適切に順位付けされた非重複リストを**再構成リスト**と呼ぶ．

以降，それぞれの処理について詳細に述べる．

⋆1 XML 検索におけるクエリでは一般的に，XML 文書の持つ構造と，テキストノード中に含まれる索引語の両方が指定される．

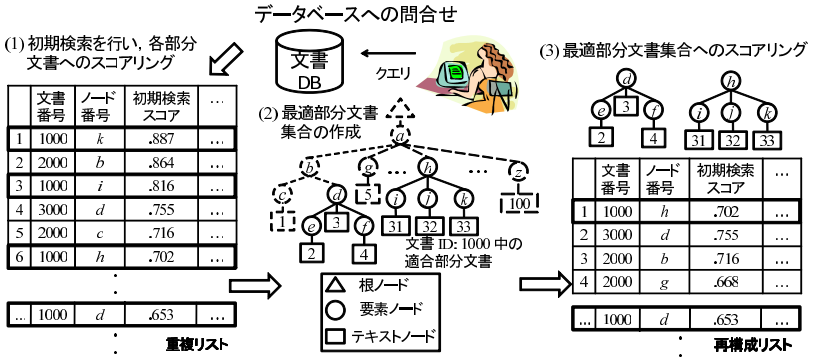


図 4　提案手法の概略図
Fig. 4　Overview of proposed method.

4.1 重複リストの作成

4.1.1 初期検索スコアリング手法の要件

処理 (1) ではその後の処理で利用する重複リストの作成を行う（図 4 (1)）．重複リストの構成はその後の処理結果に大きな影響を及ぼすため，まずは最適部分文書集合の作成にふさわしい重複リストを作成するスコアリング手法を選定する必要がある．提案手法にふさわしい重複リストの条件として以下の 2 つがあげられる．

- MAiP 値が高い．
- 重複リストの中にさまざまな粒度の部分文書が存在する．

1 つ目の条件に関して，MAiP 値が高いということは，適切に順位付けされているかはともかく，リスト中に含まれる適合部分の割合が多いということである．したがって，初期検索スコアリング手法の選定を行ううえでは検索結果上位の検索精度よりも全体的な検索精度である MAiP 値を考慮する必要がある．

2 つ目の条件に関して，重複リスト中にはさまざまな粒度の部分文書が存在している必要がある．なぜなら，クエリに対して最適な粒度の部分文書とは特定の粒度とは限らず，さまざまな粒度の部分文書が存在すると考えられるためである．それに対して，重複リスト中に粒度の大きな部分文書のみが抽出されている場合には適切な粒度の部分文書を抽出できないという問題が起こりうる．そのため，重複リスト中にはさまざまな粒度の部分文書が含まれている必要がある．なお，部分文書の粒度の大きさとテキストサイズの大きさには一定の相関があるため，部分文書の粒度を XML 文書のテキストサイズに対する各部分文書のテキ

ストサイズの割合と見なす．このとき，さまざまな粒度の部分文書を含む重複リストほど，それらの割合の散らばり（標準偏差）の値が大きくなる．つまり，標準偏差の値を計測することでさまざまな粒度の部分文書が含まれているのかどうかを確認する．

我々がさまざまな粒度を含むリストを作成することを目指す一方で，2.2.1 項で述べたとおり，近年の INEX project では文書全体を検索結果として提示することは部分文書検索を行うよりも検索精度が高くなるといわれている．しかしながら，文書全体，すなわち粒度の大きな部分文書のみを抽出することで発生する問題が存在する．具体例を示すために，1 つの文書中の複数箇所に適合部分が存在する場合を考える．図 4 (2) において，ノード d を根とする完全部分木と対応する部分文書（以降，このような部分文書を単に d と呼ぶ）と h を文書中の適合部分とする．すべての適合部分を抽出するように部分文書を 1 つ抽出（この場合は文書全体である a）すると不適合箇所を多く含む結果となってしまう．そのような事態を回避するため，1 つの XML 文書から複数の適合部分文書を抽出して適合箇所（d，h）のみを抽出することが望ましいと考えられる．

これらの理由から，提案手法で用いる初期検索用のスコアリング手法を選定する際に MAiP 値と部分文書の粒度の散らばり（標準偏差の値）を考慮している．

4.1.2　初期検索スコアリング手法の選定のための予備実験

適切な初期検索用のスコアリング手法を選定するために行った予備実験について述べる．予備実験において，我々は TF-IPF [5]，正規化 TF-IPF [7]，BM25E [10] の 3 種類のスコアリング手法に対して検索精度を評価した．BM25E を用いる際にはタグごとに重み付けを行うことが可能であるが，本論文ではすべてのタグに対して重み 1，パラメータに関しては，評価実験で利用したテストコレクションに対して経験的に最適な値を求め，$k_1 = 2.5$，$b = 0.85$ [⋆1]を設定した．予備実験に用いた INEX 2008 テストコレクション[11] については 5.1 節において詳細に述べる．

このとき，特定の粒度の部分文書のみに高いスコアが付与されるのは不適切であると見なすため，各文書から最も高いスコアを持つ部分文書のみを抽出して[⋆2]重複リストに含まれる部分文書の粒度に偏りが存在するかどうかを確認する．したがって，予備実験にあたっては，1 つの XML 文書からは 1 つの部分文書のみを抽出し，各スコアリング手法の点数の降順に並べて非重複リストを作成した．

予備実験の結果，**図 5** のとおり MAiP 値と標準偏差の値双方において BM25E が最も我々の目的に適したスコアリング手法であることが判明した．これらの結果をふまえて，以後，処理 (1) で利用する初期検索用スコアリング手法は BM25E とする．

4.1.3　仮説検証のための比較用手法選定の予備実験

初期スコアリング手法として適切な手法選定の条件として，MAiP 値の高さと重複リスト内の部分文書の粒度の散らばりの大きさをあげた．MAiP 値が高ければ検索精度の向上に直結することは自明であるが，さまざまな粒度の部分文書を含む重複リストを用いることが有益であるかどうかは定かではないために，仮説の検証を行う必要がある．検証を行うには，偏った粒度の部分文書，ここでは特に大きな粒度の部分文書に対して高いスコアを付与する手法に対しても同様に提案手法を適用し，どのような効果を及ぼすのか調査する必要があると考える．なぜなら，前述のとおり一般的に粒度の大きな部分文書を提示することで検索精度の向上に結びつくと考えられていることから，仮説検証のための比較対象として妥当であると考えられるためである．

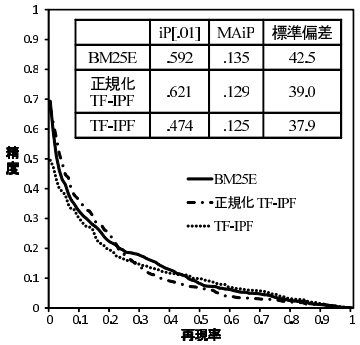
そこで，引き続き BM25E との比較用のスコアリング手法を選択するための予備実験を行った．3.2 節であげた我々の過去の研究[19] で提案したスコアリング手法を適用することで検索精度は向上するものの，テキストサイズの大きな部分文書に対して高いスコアが付与される傾向があるという知見が得られているため，BM25E と比較するための初期検索用スコアリング手法としては適切であると予想される．実際にどのような影響を及ぼすのか確認するため，BM25E を拡張させて QS，QK，両方のスコアリング手法を適用させた $QS{\cdot}QK$ の，計 3 種類のスコアリング手法に対して検索精度を評価した結果を**図 6** に示す．その結果，QS において最も高い MAiP 値を示し，抽出されるテキストサイズも BM25E を上回ったため，QS を BM25E と比較するためのスコアリング手法として用いることとした．

4.2　最適部分文書集合の作成

処理 (2) では，処理 (1) で得られた重複リスト中からクエリに対する適合部分を抽出することで最適部分文書集合の作成を行う（図 4 (2)）．つまり，より多くの適合箇所を含み，極力非適合箇所を含まない部分文書，すなわち検索結果として適切な粒度の部分文書の特定を行う．その際，従来手法と同様に，基本的には初期検索スコアの降順に部分文書の抽出を行うが，提案手法ではそれに加えていくつかの特別な処理を行う．大きな粒度の部分文書を抽出した場合には不適合な部分を含み，結果検索精度が低下する可能性が存在する．したがって，我々の提案手法では，大きすぎる部分文書が抽出されないように制限を設けつつ，1 つの部分文書から適切な粒度の複数の適合部分文書を抽出することを目指す．その際，従

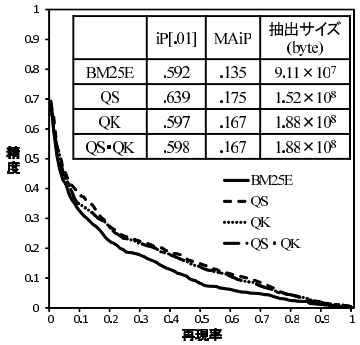
⋆1 ある部分文書 i における索引語 j の重みは $\frac{(k_1+1)tf_{i,j}}{k_1((1-b)+b\frac{cl}{avecl})+tf_{i,j}} \log \frac{N-df_i+0.5}{df_i+0.5}$ で算出される[10]．

⋆2 過去の INEX project には，1 つの文書中から 1 つだけ部分文書を抽出するタスクが存在する．



	iP[.01]	MAiP	標準偏差
BM25E	.592	.135	42.5
正規化 TF-IPF	.621	.129	39.0
TF-IPF	.474	.125	37.9

図 5　初期検索用のスコアリング手法
Fig. 5　Comparison of scoring methods for initial search.



	iP[.01]	MAiP	抽出サイズ (byte)
BM25E	.592	.135	9.11×10^7
QS	.639	.175	1.52×10^8
QK	.597	.167	1.88×10^8
QS・QK	.598	.167	1.88×10^8

図 6　過去の提案手法との比較
Fig. 6　Comparison with previous approaches.

来手法においては十分に議論されてこなかった重複に関しても考慮する．

これらをふまえ，我々は以下の 2 つの要件を満たすように最適部分文書集合を作成する．

要件 (1)　重複リストの中には粒度の大きな部分文書も存在するため，非適合箇所を含むような極端に大きな粒度の部分文書が抽出されないように制限を設ける．

要件 (2)　より多くの適合部分を抽出するために，より大きな粒度の部分文書を抽出する．

4.2.1　抽 出 制 限

要件 (1) を満たすために，我々は**抽出制限**（Extraction Limit，EL）を定義し，抽出される部分文書のテキストサイズに制限を設ける．テキストサイズの大きな文書ほど，さまざまな内容に関する記述を含むために，ユーザの情報要求に合致しない内容の記述が多くなると考えられる．したがって，適合部分として提示する際に適切な部分文書のテキストサイズは一定のテキストサイズ以下であるとし，各 XML 文書に対して制限を設けることとする．なお，テキストサイズと索引語数には一定の相関があり，さらに，各索引語の文字列長の影響を軽減するために，抽出制限は索引語数で制限を行う．

1 つの XML 文書中から抽出できる索引語数を EL，文書番号が $DocID$ の XML 文書に含まれるテキストノードのうち最適部分文書集合に含まれるテキストノードの索引語数を τ_{DocID} とすると，最適部分文書集合を作成する際に，$\tau_{DocID} < EL$ を満たす限りは初期検索スコアの降順に部分文書の抽出を繰り返す．τ_{DocID} が EL を超える場合は抽出を行わず，部分文書を破棄する．

4.2.2　部分文書統合

要件 (2) を満たすために，我々は抽出された部分文書を再構成して最適部分文書集合を作成する．その際，重複が発生した際の処理，すなわち部分文書の再構成を行う．

従来の手法で行われているような，重複が発生する際に単にスコアの高い部分文書を抽出することが，適合部分を抽出するうえでの妨げとなる場合がある．**図 7** を用いて具体例を示す．このとき c が適合部分であるとすると，c の初期検索スコアが d の初期検索スコアよりも高い場合には，c が検索結果として抽出されることになる．それに対して，c の初期検索スコアが d の初期検索スコアよりも低かった場合，まずは d が抽出され，その後 c の抽出が試みられるが，ここで重複が発生するために c は破棄される．その結果，すべての適合部分を網羅できなくなる．

このような問題を解決するため，我々は**部分文書統合処理**を行っている．部分文書統合処理では，重複が発生した場合に，初期検索スコアの高低は考慮せず，単に粒度の大きな部分文書を抽出し，粒度の小さな部分文書は検索結果から破棄する．粒度の小さな部分文書は粒度の大きな部分文書の一部分から構成されるため，結果としてすべての適合部分を抽出することが可能となる．しかしながら，部分文書の統合を繰り返すたびに検索結果として抽出される部分文書の粒度が大きくなるため，最終的に文書全体が検索結果となりうるという問題がある．したがって，要件 (1) を満たす限りは部分文書統合処理を行うことで，大きすぎる粒度の部分文書が検索結果となることを抑制する．

ここで再び図 7 を用いて部分文書統合処理の具体例を示す．d の初期検索スコアよりも c の初期検索スコアが高い場合は従来と同様 c を抽出するのに対して，d の初期検索スコアが c の初期検索スコアよりも高い場合には d の代わりに c を検索結果として抽出する．

4.3　再構成リストの作成

最適部分文書集合の作成が完了すれば，最適部分文書集合に含まれる各部分文書に対してスコアを付与し，最終的にユーザに提示する再構成リストを作成する（図 4 (3)）．処理 (2) では検索結果として提示するのに適切な粒度の部分文書を抽出したのであって，抽出された各部分文書をどのような順序で提示することが妥当であるのかに関しては考慮していない．そこで，高精度 XML 検索を実現するため，最適部分文書集合に含まれる部分文書の中から，より情報に富みユーザの情報要求を満たすことが期待される部分文書を特定するためのスコアリング手法を提案する．

初期検索スコアをそのまま用いることで再構成リストを作成することは可能である．しかしながら，初期検索スコアをそのまま利用するということは，従来の手法と同様，各部分文

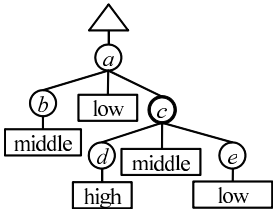


図 7　部分文書統合例
Fig. 7　Example of *Overwrite* fragments.

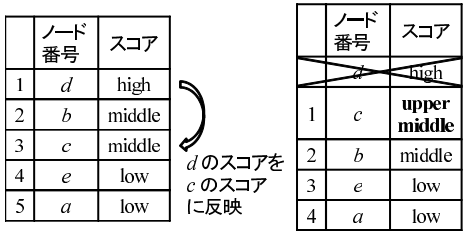


図 8　*Bottom-Up* スコアリング手法の概要
Fig. 8　Example of *Bottom-Up* scoring method.

書を独立した文書として個別にスコアを計算していることと変わりない．我々が部分文書の再構成を適用した理由は，同一 XML 文書間に存在する関係をふまえて検索結果を決定する必要があると考えたためである．したがって，最適部分文書集合へのスコアリング手法を提案するためには，各部分文書と関連する部分文書の持つ統計量を考慮する必要があると考えた．

以上の議論をふまえて，我々は同一 XML 文書中の部分文書の持つ統計量を用いた，2 種類のスコアリング手法を提案する．それぞれ，子孫部分文書の持つ統計量を先祖部分文書のスコア計算に用いる **Bottom-Up スコアリング手法**と，先祖部分文書の持つ統計量を子孫部分文書のスコア計算に用いる **Top-Down スコアリング手法**である．以下，2 つの手法について詳細に述べる．

4.3.1　Bottom-Up スコアリング手法

4.2.2 項の部分文書統合では，すでに抽出されている部分文書と新たに抽出される部分文書の間に重複が発生した際に，子孫にあたる部分文書を最適部分文書集合から取り除き，代わりに先祖にあたる部分文書を挿入する．このとき，初期検索スコアを比較すると，先祖部分文書は子孫部分文書よりも低いスコアを持っていたことになる．そのため，検索結果をランキングする際に，初期検索スコアをそのまま用いれば，先祖部分文書は子孫部分文書が提示されるはずであった順位よりも下位で提示されることになる．つまり，元は高い順位で提示されるはずであった子孫部分文書は先祖部分文書の一部として提示されることになるが，その際に本来子孫部分文書が提示されはずであった順位よりも下位で提示されることになる．

具体例を図 7，**図 8** を用いて説明すると，従来の方式で非重複リストを作成した場合に，まずは d が抽出され，その後 b が抽出される．それに対して，提案手法では部分文書統合が起こるために d の代わりに c が抽出される．このとき，初期検索スコアの降順に抽出した場合には b が抽出された後に c が抽出されるために，本来は b より上位で提示されるはずであった c に含まれるテキストノードが，b よりも下位で提示されることになる．我々はこのような高い初期検索スコアを付与された部分文書を子孫に持つ部分文書は，より上位の検索結果として提示することが妥当であると考える．そこで，高い初期検索スコアを持つ部分文書を子孫に持つ部分文書に対してふさわしいスコアを与えるための Bottom-Up スコアリング手法（以後 BU）を定義する．それによって，c に初期検索スコアよりも高いスコアを付与し，検索結果上位で提示されるようにする．

子孫部分文書の持つ統計量を考慮する必要があると考えられるが，先祖部分文書のスコアが低いにもかかわらず検索結果上位において提示することは不適切である．そのため，BU は，単に子孫部分文書の持つスコアを用いて先祖部分文書に対してスコアを付与するのではなく，先祖部分文書と子孫部分文書の初期検索スコアを適切に反映されなければならない．このとき，先祖部分文書のテキストノードの一部分は子孫部分文書のテキストノードから構成されているため，重複するテキストノードの部分が子孫部分文書の影響が及ぶ範囲であるとする．その場合，削除された部分文書と挿入される部分文書のテキストサイズの比と，それぞれの初期検索スコアから算出されるのが妥当であると考えられる．

なお，初期検索において上位で提示されていた部分文書を，リスト上位の検索結果として提示させるのであれば，子孫部分文書のうち最も高い初期検索スコアを持つ子孫部分文書の影響を強く反映させることで実現可能であると考える．なぜなら，仮にある先祖部分文書に含まれる子孫部分文書のうち多くの部分文書が高いスコアを持つのであれば先祖部分文書自体も高い初期検索スコアを持つことが予想される．したがって，前述のような部分文書統合の結果急激に初期検索スコアが低下するという問題が起こらず，すべての子孫部分文書を考慮する必要がないと考えられるためである．そこで，BU を定式化する際には統合処理によって削除された部分文書のうち最も高い初期検索スコアを持つ部分文書の統計量を利用する．

以上の議論より，BU を以下のように定式化する．このとき，f_a を先祖部分文書，f_d を子孫部分文書のうち最も初期検索スコアの高い部分文書とする．

$$s_{bu}(f_a) = \frac{|f_d|}{|f_a|} \cdot s(f_d) + \frac{|f_a| - |f_d|}{|f_a|} \cdot s(f_a) \quad (1)$$

ただし，$s_{bu}(f_a)$ を f_a の再計算されたスコア，$|f_x|$ を f_x のテキストサイズ，$s(f_x)$ を f_x の

初期検索におけるスコアとする．

4.3.2　Top-Down スコアリング手法

クエリキーワードの中には，語としてさまざまな意味を持つ索引語が存在する．語の持つ複数の意味から適切な意味を特定することは困難であるが，他のクエリキーワードとの共起情報は語の意味を特定するうえで大きな手助けになる．つまり，もしある部分文書中に多くの種類のクエリキーワードが出現するのであれば，その部分文書中のクエリキーワードはクエリによって意図される本来の意味を持つと考えられる．このような考えのもと，我々は過去の研究[19] において，部分文書中に含まれるクエリキーワードの種類数を考慮したスコアリング手法である QK を用いれば有益な部分文書を抽出することができるという知見を得た．

テキストサイズの大きな部分文書は，そのテキストノードに含まれる索引語数も大きくなる傾向があるために，多くの種類のクエリキーワードを含む傾向にある．その結果テキストサイズの大きな部分文書に高スコアが付与されるが，これでは実際は有益な部分文書であったとしても，テキストサイズが小さい部分文書に高いスコアが付与されにくいという問題が存在する．このような問題を解決するために，我々は各部分文書のテキストサイズに依存せずに有益な部分文書を特定するための Top-Down スコアリング手法（以後 TD）を提案する．

各部分文書の持つテキストは文書全体のテキストの一部であるために，文書の有用性は内包する部分文書に引き継がれると仮定する．つまり，有用と判定された文書中に出現する部分文書も同様に有益であると見なす．なお，最適部分文書集合を作成する段階で文書中の適合部分が特定されているために，同一文書内において適切な粒度を特定させる必要がない．そのため，いずれの文書（に属する部分文書）がより有益であるのかどうかを特定することが重要である．そこで，部分文書の属する文書全体に出現するクエリキーワードの種類数を用いてスコアの再計算を行う．f をスコア計算される部分文書，D_f を f が含まれる文書とすると，Top-Down スコアは以下の数式で表現することができる．

$$s_{td}(f) = s(f) \cdot keyword(D_f) \tag{2}$$

ただし，$s_{td}(f)$ を f の Top-Down スコア，$s(f)$ を f の初期検索スコア，$keyword(D_f)$ を D_f に含まれるクエリキーワードの種類数とする．

4.4　最適部分文書集合と再構成リスト作成例

4.1〜4.3 節の議論をふまえて，最適部分文書集合の作成と BU を利用した場合の再構成リストの作成の具体例を**図 9** に示す[⋆1]．ただし，今回の例では単純化のために文書番号 1,000 の文書についてのみ扱う．また，$EL = 100$ とする．

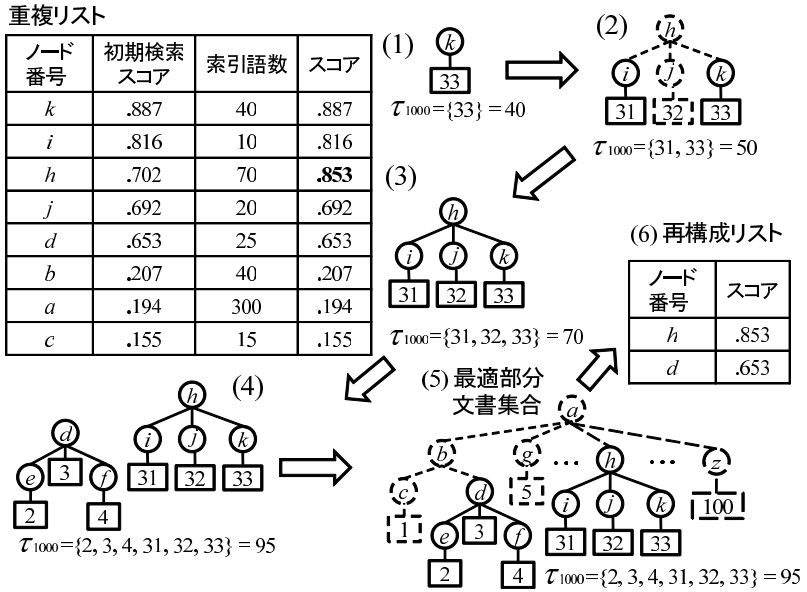


図 9　再構成リスト作成例
Fig. 9　Example of generating refined ranked list.

図 4 の処理手順に則り，まずはじめに初期検索を行って重複リスト（図 9 の左のテーブル）を得る．このリストを用いて，初期検索のスコアの降順に部分文書抽出を行う．重複リスト中で最も高いスコアを持つ部分文書は k であるため，まずは k の抽出を試みる．k はテキストノード 33 を含み，その索引語数は 40 であるため，k は最適部分文書集合に挿入され，$\tau_{1000} = 40$（$< EL$）となる（図 9 (1)）．

そして，k の次に初期検索のスコアが高い部分文書である i が抽出される．i に含まれるテキストノード 31 は索引語数が 10 であるため，i を抽出した場合，$\tau_{1000} = 50$（$< EL$）となり i の抽出が実行される（図 9 (2)）．

⋆1 TD はあらかじめ計算を行っておくことも可能であるが BU は部分文書統合が起こった際に再計算を行う必要があるため，ここでは BU を最適部分文書集合に対するスコアリング手法とする．

次の抽出されるべき部分文書の候補は h である．すでに抽出された i と k はいずれも h の子孫部分文書であるため，部分文書統合処理が発生する．このとき，新たにテキストノード 32 を抽出した場合には $\tau_{1000} = 70$（$< EL$）であるため，最適部分文書集合から i と k が取り除かれ，代わりに h が挿入される（図 9 (3)）．部分文書統合が発生した時点で，h に対して 4.3.1 項の BU スコアの計算が行われる（式 (1) より，$S_{BU}(h) = 40/70 \cdot 0.887 + (70 - 40)/70 \cdot 0.702 = 0.853$）．

重複リストにおいて h の次に高いスコアを持つ部分文書は j であるが，j はすでに最適部分文書集合の中に抽出されている h の子孫部分文書であるために，抽出は行われない．その後の処理では d の抽出が行われ（図 9 (4)），$\tau_{1000} = 95$（$< EL$）となる．その後，b，a，c と次々と部分文書の抽出が試みられるが，いずれも τ_{1000} が EL を超えるために実行されない．したがって，最終的に d，h が部分文書番号 1,000 に対する適合部分文書として選択されることになる（図 9 (5)）．最後に，d，h に付与されたスコアの降順に部分文書を並べて再構成リストの作成を行う（図 9 (6)）．

実際の最適部分文書集合作成の際には，上記の処理をすべての文書に対して行い再構成リストを作成する．

5. 評 価 実 験

ここでは，提案手法の有効性の確認と，従来手法との比較のために行った評価実験ついて説明する．

5.1 テストコレクション

評価実験には INEX 2008 テストコレクションを使用した．このテストコレクションは，は 3 つの要素から構成されており，1 つ目が INEX document colleciton，2 つ目が INEX topics，そして 3 つ目が INEX relevant assessments である．

INEX document collection は約 660,000 個の XML 文書の集合から構成されており，2006 年初期に収集された英語版の Wikipedia コーパスである．クエリ集合である INEX topics は合計 68 個のクエリ集合から構成されており，次節以降の評価実験ではすべてのクエリを用いた．INEX topics に含まれるクエリは 2 種類存在し，一方はキーワードのみを指定する CO（Content only）クエリであり，もう一方はキーワードと構造を指定する CAS（Content and Structure）クエリである．また，これらのクエリはすべて NEXI（Narrowed-Extended XPath I）[20] クエリで表現される．INEX relevance assessments は XML 部分文書検索用の評価ツールである．XML 検索システムは INEX relevance assessments に対して，非重複リストを提出することでさまざまな評価尺度による検索精度を評価する．なお，クエリに対する正解データは，人手によって文書中の適合箇所を選択することで作成している．

表 1　EL の変動にともなう検索精度
Table 1　iP[.01] at each EL.

EL	10	20	30	40	50	60	70	80	90	100	200	300
iP[.01]	.0	.0400	.218	.278	.350	.369	.396	.426	.453	.482	.536	.595

EL	400	500	600	700	800	900	1,000	2,000	3,000	4,000	5,000	6,000
iP[.01]	.629	.649	.648	.649	.655	.656	**.663**	.659	.659	.660	.662	.662

5.2 抽出制限のための予備実験

4.2.1 項で述べた抽出制限 EL を施すため，最適な索引語数の閾値を調査するために予備実験を行った．EL の値に対する iP[.01] の検索精度を**表 1** に示す．これにより，EL の値が 1,000 の場合に検索精度 iP[.01] が最大になることが判明した．この予備実験の結果をふまえ，以降の実験においては $EL = 1{,}000$ を設定した．

5.3 最適部分文書集合に対する評価実験

評価実験では最適部分文書集合と従来手法に対して検索精度を比較し，部分文書の再構成による効果を確認した．また，初期検索スコアリング手法として BM25E を用いた場合と QS を用いた場合とでどのような違いが発生するのかを調査した．なお，実験の際の従来手法は，提案手法としても用いる初期検索スコアリング手法によって作成された，1 つの文書から複数の部分文書を抽出した非重複リストである．

図 10 より，BM25E を初期検索スコアリング手法とした場合には，従来手法と比較し，公式尺度である iP[.01] をはじめとしてすべての再現率点において優位な結果を示した．結果として，高精度 XML 検索を実現するうえで，部分文書の再構成を行うことは非常に効果的であるということが判明した．このことから，部分文書の再構成を行うことで検索精度を改善することが可能であるという結果が得られた．

その一方で，QS を初期検索用スコアリング手法とした場合では，部分文書の再構成を行った場合と従来手法の検索精度に差が見られなかった．このことから，我々が当初仮定していたように，大きな粒度の部分文書に対してのみ高いスコアを付与するスコアリング手法よりも，重複リストの中にさまざまな粒度の部分文書を含むスコアリング手法が部分文書の再構成を適用させる際にふさわしいということが判明した．これは，部分文書統合が発生した際に，適合部分文書として適切な粒度の部分文書が存在していなければ，適合部分のみを抽出することができないためであると考えられる．

表 2　再構成リストに対するスコアリング手法の影響
Table 2　Effect of scoring mehtods for set of integrated XML fragments.

	BU	*BU·TD*	*TD*	初期検索スコアリング法	従来手法
iP[.01]	.6653	.6638	.6637	.6628	.6131
テキストサイズ（byte）	1.47×10^8	2.31×10^8	2.28×10^8	1.47×10^8	1.30×10^8

これらの結果から，我々の手法の検索精度は重複リストの構成部分文書に大きく左右されるということが判明した．したがって，以降の実験では BM25E を初期検索スコアリング手法と設定する．

5.4　再構成リストに対する評価実験

最適部分文書集合に含まれる部分文書に対するスコアリング手法の効果を確認するための評価実験を行った．比較対象は，*BU*，*TD*，両手法を適用させた *BU·TD*，そしてベースラインの手法として初期検索スコアリング法に対して比較を行った．その結果を**表 2** に示す．*BU*，*TD*，*BU·TD* すべてにおいて iP[.01] で初期検索スコアリングを上回ったことからも，子孫部分文書の持つ統計量や先祖部分文書の持つ統計量を利用したスコアリングを行うことで検索結果上位において高精度検索を実現することが可能であるということが判明した．

また，*TD* や *BU·TD* では抽出される部分文書のテキストサイズが大幅に増大した．これは，過去の研究[19] においてテキストサイズの大きな部分文書ほど多くの種類のクエリキーワードを含んだことと同様に，テキストサイズの大きな文書ほど多くの種類のクエリキーワードを含むためであると考えられる．いずれにせよ，適切な粒度の部分文書を提示するという本来の我々の目的を満たしていない．これに対して，*BU* では抽出される部分文書のテキストサイズを増加させることなく検索精度を向上させており，我々の目標に対して最も理想的な結果を示した．これらの議論をふまえ，我々の提案手法として *BU* を選択した．

提案手法と従来手法の比較では，iP[.01] において約 8%検索精度が向上した（表 2 参照）．さらに，統計的仮説検定を行うことで提案手法の有意性を確認する．提案手法（*BU*）と従来手法（BM25E）において，クエリごとに iP[.01] における検索精度を計測し，符号検定[21] を行った．その結果，有意水準 5%において提案手法と従来手法の間に有意に差がある（p 値 $= 0.0003580$）と判定されたため，提案手法は従来手法と比較して有意に検索精度が高いと判断できる．

また，他の INEX project 参加者のシステムと検索精度の比較を行うことで本提案の有効性を確認した結果を**表 3** を用いて説明する．比較対象として，我々と同様に CAS クエリと CO クエリ両方を用いて評価実験を行っている参加者 3 チームとの比較を行った[11] ところ，

表 3　INEX project 参加者の他システムとの精度比較
Table 3　Comparisons with other systems of INEX participants.

Team	iP[.00]	iP[.01]	iP[.05]	iP[.10]	MAiP
提案手法（*BU*）	.7054	.6653	.5332	.4625	.1888
Renmin Univ. of China	.5969	.5969	.5815	.5439	.2486
Queensland Univ. of Technology	.6232	.6220	.5521	.4617	.2134
Univ. of Amsterdam	.6514	.6379	.5901	.5280	.2261

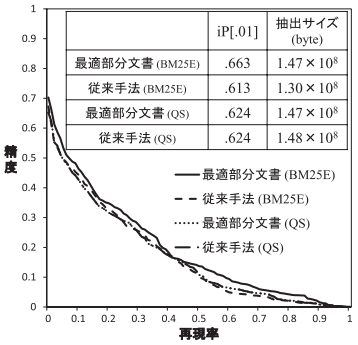
我々の提案手法は iP[.01] において最も高い検索精度を示した．これにより，提案手法は十分に有益な検索技術であると確認できた．

一方，提案手法の処理コストの計算量について述べる．再構成リストを生成するためには，1) 重複リストの作成，2) 重複リストから再構成リスト作成，の 2 つの手順が必要であるが，1) の重複リスト作成の計算量は提案手法の性能とは別問題であるためにここでは考慮しない．2) の再構成リスト作成の計算量に関して，重複リスト中の部分文書数を n，重複リスト中に含まれるユニークな文書数を d，部分文書のうち統合される部分文書の割合を p とし，d と p を定数とすると，各文書ごとに n/d 個の部分文書がある．リストを降順に 1 度走査することで部分文書の再構成を行うことができ，文書数分だけ同じ操作が必要となるので，再構成処理のコストは $O(n)$ となる．また，再構成されるたびに部分文書が減少するため，再構成リストに含まれる部分文書数は，$n(1-p)$ 個である．このとき，$p = 0$ だと再構成リストをソートし直す必要がないので，計算量は再構成の計算量を足し合わせて $O(n)$ である．それ以外の場合，つまり $0 < p \leq 1$ の場合においてはスコアの再計算にともないソートし直す必要がある．このとき，計算量が $O(n \log n)$ のソートを用いると，$p = 1$ はほぼ起こりえないので，結局計算量は $O(n \log n)$ となる．つまり，提案手法の計算量はソートの計算量に依存する．

5.5　部分文書検索と文書検索の比較

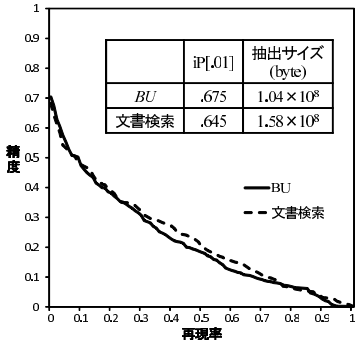
本論文で繰り返し述べたように，現在の XML 検索では，一般的に文書検索は部分文書検索よりも有益であるといわれている[11]．そのため，ここでは，提案手法である *BU* と文書検索のいずれがより効果的な検索を実現することができるのか評価実験を行い，比較した．実験の際の初期検索スコアリング手法としては BM25E を用い，検索結果として文書全体を提示することが可能であるクエリ 41 個を対象に行った．

評価実験の結果，**図 11** に示すとおり，iP[.01] における提案手法の検索精度は文書検索の検索精度と比較して約 5%精度が高いという結果が得られた．ここで，先ほどと同様に



	iP[.01]	抽出サイズ (byte)
最適部分文書 (BM25E)	.663	1.47×10^8
従来手法 (BM25E)	.613	1.30×10^8
最適部分文書 (QS)	.624	1.47×10^8
従来手法 (QS)	.624	1.48×10^8

図 10　再構成による検索精度への影響
Fig. 10　Effect on reconstructing fragments.



	iP[.01]	抽出サイズ (byte)
BU	.675	1.04×10^8
文書検索	.645	1.58×10^8

図 11　文書検索と部分文書検索の比較
Fig. 11　Comparison of XML fragment search and document search.

符号検定を行ったところ，有意水準 5%において提案手法と従来手法の間に有意に差がある（p 値 $= 0.01151$）と判明した．そのため，提案手法を適用した部分文書検索は，文書検索を行うよりも有意に検索精度が高いと検定された．

さらに，当然のことながら，部分文書検索を行うことで文書検索に比べて抽出されるテキストサイズを大幅に軽減することができた．これにより，より集約した有益な検索結果を提示することができ，情報検索におけるユーザの負担を軽減することができると考える．

6. おわりに

本論文では，高精度 XML 検索実現のための部分文書の再構成手法と同一文書内の部分文書の持つ統計量を利用したスコアリング手法を提案した．その際，これまでの XML 検索に関する研究では十分に考慮されてこなかった部分文書間に発生する重複を考慮しつつ適切な粒度の部分文書を検索結果として抽出し，情報要求に合致する部分文書を検索結果上位に提示することを目指した．

評価実験の結果，部分文書の再構成や，同一文書の部分文書の持つ統計量を考慮したスコアリングを行うことで検索精度を向上させることに成功した．その際，幅広い粒度の部分文書対してに高いスコアが付与される手法を初期検索スコアリング手法とすることでより効果的な検索が可能である．また，文書単位の検索よりも部分文書単位の検索においてより高い検索精度を示したことから，抽出される部分文書のテキストサイズや粒度を制限することは有益であるということが示唆された．

本論文において行った文書ごとに抽出可能なテキストサイズの閾値を設定する抽出制限では，その閾値の決定方法は人手で行っており，この閾値がテストコレクションごとに異なるのかどうかに関しても追求していない．したがって，今後の課題としては，抽出制限の閾値を自動設定できる手法の考案や，制限を設けることが適切である文書を特定する手法の考案を行う必要がある．

謝辞　本研究の一部は，文部科学省科学研究費補助金特定領域研究（課題番号：21013035），日本学術振興会科学研究費補助金基盤研究（A）（課題番号：22240005），日本学術振興会科学研究費補助金若手研究（B）（課題番号：22700248）によるものである．ここに記して謝意を表す．

参 考 文 献

1) Kazai, G., Lalmas, M. and de Vries, A.P.: The Overlap Problem in Content-Oriented XML Retrieval Evaluation, *Proc. 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.72–79 (2004).
2) Malik, S., Kazai, G., Lalmas, M. and Fuhr, N.: Overview of INEX 2005, *Advances in XML Information Retrieval and Evaluation*, Lecture Notes on Computer Science, Vol.3977, pp.1–15, Springer Berlin (2006).
3) Manning, C.D., Raghavan, P. and Schutze, H.: *Introduction to Information Retrieval*, pp.157–159, Cambridge University Press (2008).
4) 中村聡史：情報信頼に対する信頼性調査および結果，人工知能学会誌，Vol.23, No.6, pp.767–774 (2008).
5) Grabs, T. and Schek, H.-J.: PowerDB-XML: A Platform for Data-Centric and Document-Centric XML Processing, *Proc. 1st International XML Database Symposium*, Lecture Notes on Computer Science, Vol.2824, pp.100–117, Springer Berlin (2003).
6) Salton, G. and Buckley, C.: Term-Weighting Approaches in Automatic Text Retrieval, *Journal of Information Processing and Management*, Vol.24, No.5, pp.513–523 (1988).
7) Liu, F., Yu, C., Meng, W. and Chowdhury, A.: Effective Keyword search in Relational Databases, *Proc. 2006 ACM SIGMOD International Conference on Management of Data*, pp.563–574, ACM (2006).
8) Robertson, S., Walker, S., Jones, S., Hancock-Beaulieu, M. and Gatford, M.: Okapi at TREC-3, *The 3rd Text REtrieval Conference* (*TREC-3*), pp.109–126 (1995).

9) Robertson, S., Zaragoza, H. and Taylor, M.: Simple BM25 Extension to Multiple Weighted Fields, *Proc. 13th ACM International Conference on Information and Knowledge Management*, pp.42–49 (2004).
10) Liu, W., Robertson, S. and Macfarlane, A.: Field-Weighted XML Retrieval Based on BM25, *Advances in XML Information Retrieval and Evaluation*, Lecture Notes on Computer Science, Vol.3977, pp.161–171, Springer Berlin (2006).
11) Kamps, J., Geva, S., Trotman, A., Woodley, A. and Koolen, M.: Overview of the INEX 2008 Ad Hoc Track, *INEX 2008 Workshop Pre-proceedings*, pp.1–28 (2008).
12) Blanken, H., Grabs, T., Schek, H.-J., Schenkel, R. and Weikum, G.: *Intelligent Search on XML Data: Applications, Languages, Models, Implementations and Benchmarks*, Lecture Notes on Computer Science, Vol.2818, Springer-Verlag (2003).
13) Schmidt, A., Kersten, M. and Windhouwer, M.: Querying XML Documents Made Easy: Nearest Concept Queries, *Proc. 17th International Conference on Data Engineering*, p.321, IEEE (2001).
14) Xu, Y. and Papakonstaninou, Y.: Efficient Keyword Search for Smallest LCAs in XML Databases, *Proc. 2005 ACM SIGMOD International Conference on Management of Data*, pp.527–538, ACM (2005).
15) Li, G., Feng, J., Wang, J. and Zhou, L.: Effective Keyword Search for Valuable LCAs over Xml Documents, *Proc. 16th ACM Conference on Information and Knowledge Management*, pp.31–40 (2007).
16) Hristidis, V. and Koudas, N.: Keyword Proximity Search in XML Trees, *IEEE Trans. Knowledge and Data Engineering* (*TKDE*), Vol.18, No.4, pp.525–539 (2006).
17) Liu, Z. and Chen, Y.: Identifying Meaningful Return Information for XML Keyword Search, *Proc. 2007 ACM SIGMOD International Conference on Management of Data*, pp.329–340, ACM (2007).
18) Huang, Y., Liu, Z. and Chen, Y.: Query Biased Snippet Generation in XML Search, *Proc. 2008 ACM SIGMOD International Conference on Management of Data*, pp.315–326, ACM (2008).
19) Keyaki, A., Hatano, K. and Miyazaki, J.: A Query-oriented XML Fragment Search Approach on A Relational Database System, *Journal of Digital Information Management* (*JDIM*), Vol.8, No.3, pp.175–180 (2010).
20) Trotman, A. and Sigurbjörnsson, B.: Narrowed Extended XPath I (NEXI), *Advances in XML Information Retrieval*, pp.16–40 (2005).
21) Hollander, M. and Wolfe, D.A.: *Nonparametric Statistical Methods*, Wiley-Interscience (1999).

(平成 22 年 9 月 20 日受付)
(平成 23 年 1 月 10 日採録)

(担当編集委員　橋本 隆子)

欅　　惇志（学生会員）

同志社大学大学院文化情報学研究科文化情報学専攻博士前期課程在学中．2009 年同大学文化情報学部文化情報学科卒業．高精度 XML 情報検索の研究に従事．2009 年第 148 回データベースシステム・第 95 回情報学基礎合同研究発表会学生奨励賞受賞．日本データベース学会，同志社大学文化情報学会各会員．

波多野賢治（正会員）

同志社大学文化情報学部准教授．博士（工学）．1995 年神戸大学工学部計測工学科卒業．1999 年同大学大学院自然科学研究科博士後期課程修了．同年日本学術振興会未来開拓学術研究事業研究員，奈良先端科学技術大学院大学情報科学研究科助手．2005〜2006 年米国 AT&T Labs-Research 客員研究員．2006 年同志社大学文化情報学部専任講師，2008 年より現職．2007 年電子情報通信学会論文賞受賞．Web 情報検索，XML データベース等の研究に従事．電子情報通信学会，日本データベース学会，ACM，IEEE Computer Society 各会員．

宮崎　　純（正会員）

奈良先端科学技術大学院大学情報科学研究科准教授．博士（情報科学）．1992 年東京工業大学工学部情報工学科卒業．1997 年北陸先端科学技術大学院大学情報科学研究科博士後期課程修了．同大学助手を経て，2003 年より現職．2000〜2001 年テキサス大学アーリントン校客員研究員．2003〜2007 年科学技術振興機構さきがけ研究員．高性能・高機能データベースならびに情報検索の研究に従事．電子情報通信学会，日本データベース学会，ACM，IEEE Computer Society 各会員．