DEIM Forum 2010 C7-2

XML 部分文書の再構成に基づく検索結果の提示手法

欅　悼志† 波多野賢治†† 宮崎　純†††

† 同志社大学大学院文化情報学研究科
〒 610–0394, 京都府京田辺市多々羅都谷 1–3
†† 同志社大学文化情報学部
〒 610–0394, 京都府京田辺市多々羅都谷 1–3
††† 奈良先端科学技術大学院大学情報科学研究科
〒 630–0192, 奈良県生駒市高山町 8916–5
E-mail: †keyaki@ilab.doshisha.ac.jp, ††khatano@mail.doshisha.ac.jp, †††miyazaki@is.naist.jp

あらまし 本稿では，構造化文書検索における効果的な検索結果提示手法を提案する．従来の検索手法では，文書中の各部分に対して得点付けを行い，文書ごとにもっとも大きなスコアが付与された部分を検索結果として提示していた．しかし，それでは検索結果中にクエリに対して不要な部分が混じる場合や，適合部分の一部しか抽出できない場合が起こり得る．そこで，クエリに適合する部分をテキストノード単位で抽出することで検索結果を構築し，それぞれのノードの構造的な関係を考慮することで，既存手法では考慮されてこなかった “隠れた正解” の抽出も行う．また，評価実験の結果，検索結果を再構築することで検索精度が向上するという知見が得られた．
キーワード XML, 部分文書検索，効果的な情報検索手法，検索結果の再構成

A Method of Presenting Search Results based on the Reconstructing Ranked Results

Atsushi KEYAKI†, Kenji HATANO††, and Jun MIYAZAKI†††

† Graduate School of Culture and Information Science, Doshisha University
1–3 Tatara–Miyakodani, Kyotanabe, Kyoto 610–0394 Japan
†† Faculty of Culture and Information Science, Doshisha University
1–3 Tatara–Miyakodani, Kyotanabe, Kyoto 610–0394 Japan
††† Graduate School of Information Science, Nara Institute of Science and Technology
8916-5 Takayama, Ikoma, Nara 630-0192, Japan
E-mail: †keyaki@ilab.doshisha.ac.jp, ††khatano@mail.doshisha.ac.jp, †††miyazaki@is.naist.jp

Abstract In this paper, we propose a method for presenting meaningful XML fragments. In the past approaches, all XML fragment is scored, and a XML fragment with the largest score in every XML document is presented to the user as the meaningful XML fragment. In other words, they extract only one XML fragment from an XML document. In these ways, an XML fragment may contain much unsuitable part for a query or extract a part of appropriate fragments. To cope with the problems, we extract text nodes separated from an XML fragment. We also try to find the “hidden answer” utilizing the structural relation in each node. A result of the experiments shows our approach has a higher precision than traditional one.
Key words XML, XML fragment search, effective search technique, reconstruct search results

1. はじめに

現在，Web 文書やさまざまなアプリケーションのデータなど，多くのデータが XML で記述されている．これは，XML がデータ交換のための標準フォーマットあるため，異なるデータフォーマットであっても XML で記述されていれば互いにデータの交換が可能であるという点が重要視されているからである．従って，このように多くのデータが XML で作成された

結果，Web 上には膨大な数の XML 文書が蓄積している．これら XML データから必要な情報を発見することは困難な作業であるため，情報検索技術を利用して高速かつ正確な検索を行う技術の開発は強く望まれている．

XML 情報検索に関する研究には二つの目的が存在し，一つは効率的な検索の実現，もう一つは効果的な検索の実現である．XML はその性質上一つの文書からあらゆる粒度の複数の文書，すなわち部分文書を取り出すことが出来るため，検索対象となる部分文書数は元の XML 文書数に対して大幅に増大する．その結果検索速度が低下するという問題が起こるため，部分文書中のクエリに対する適合部分を効率的に発見することを目指す研究が積極的に行われてきた．これらの研究の成果や昨今のコンピュータやデータベースマネージメントシステムの高性能化によって効率的な検索が実現しつつある．一方，最近では二つ目の目的である効果的な検索の実現にも注目が集まっている．従来のテキスト文書検索は検索結果としてユーザの情報要求に適合する文書のリストをユーザに提示するのに対して，部分文書検索では文書のうちのもっとも適合する箇所を抽出して提示することができる．そのため，ユーザは文書全体を確認する必要がなくなり，情報検索を行う際の負担を軽減することができる．

検索結果はユーザにとって利便性の高い情報が提示される必要があるが，従来の部分文書検索手法では，各部分文書に対してスコアリングを行い，付与されたスコアのもっとも高い部分文書を文書中の最適部分としてそのまま提示しているのみである (注1)．しかし，実際のユーザの情報要求を満たす部分が一つの部分文書で表現できているとは限らない．なぜなら，例えば文書中の適合部分が複数箇所存在した場合に一つの部分文書で適合部分を抽出すると，部分文書中に不要な部分が多く混じったり，大きすぎる部分文書が検索結果となったり，一部の適合部分のみしか抽出できないという問題が起こりうるからである．また，従来の部分文書検索技術では低いスコアが付与された適合部分を抽出することができないという問題が存在する．なぜなら，クエリに対する適合部分文書に必ずしも高いスコアが付与されているとは限らないからである．

これらの問題を解決するために，本稿では部分文書検索技術によって付与された各部分文書に対するスコアを利用し，文書ごとに検索結果の再構成を行うことでクエリに対する適合部分文書の抽出を行う．更に，これまでの部分文書検索技術では発見することができなかった “隠れた正解”，すなわち，低スコアの適合部分の抽出も行うことでユーザにとって望ましい検索結果の提示を行う．

以下，2. では関連研究について，3. では提案手法について述べる．また，4. 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，5. ではまとめと今後の課題について述べる．

(注1)：つまり，一つの文書からは一つの部分文書のみを検索結果として提示する．

2. 関 連 研 究

これまで，効果的に検索結果を提示するためにさまざまな研究が行われてきた．ユーザが情報検索を行う際に，検索結果として提示された文書が有用なページか否かを判断するための Web ページの要約文をスニペット (Results Snippet) [5] と呼ぶ．スニペットは重要部分 (文) 抽出技術を利用して生成され，クエリ依存と文書依存の二つのアプローチを取り入れることでパーソナライズ化されたスニペット生成を行う研究なども存在する [8]．スニペットは自然言語処理によって重要と判断された部分を抽出するのに対して，我々の提案手法ではクエリに適合する部分文書の抽出を行うという点は異なるが，ユーザが情報検索を行う際の支援を目指すという点では方向性を同じとする研究である．

また，1. でも述べたように，効率的な検索の実現のためにこれまでさまざまな研究が行われてきており，中でも LCA (Lowest Common Ancestor) [6] の概念が多用されてきた．LCA とは任意のノードのもっとも深さの低い共通祖先ノードを指すが，一般的にはすべてのクエリキーワードを子孫ノードに持ち，かつ，どの子孫ノードもすべてのクエリキーワードを子孫として持たないノードを指す．なお，過去の研究においては LCA をそのまま利用するのではなく，SLCA (Smallest Lowest Common Ancestor) [7] や VLCA (valuable Lowest Common Ancestor) [3]，MCT (Minimum Connectiong Tree) [1] など，多くのキーワード探索アルゴリズムに拡張されている．これらの研究は検索速度の向上には大きく貢献を果たしているが，LCA を拡張して得られた部分をクエリに対する最適部分としたところ検索精度が低下するという問題が起こる．そこで，検索精度を向上させるための取り組みが行われており，そのうちの一つである MLCA (Meaningful LCA) [4] では，タグの分類や意味的な照合を行った上で LCA を根とする部分木からクエリと関係の強い部分のみを抽出して検索結果となる部分木の構築を行っている．eXtract [2] では MLCA に対して更に加工を行っているが，その際にクエリの分析を行うことでユーザにとって理想的な部分木の抽出を目指している．eXtract は LCA を元に検索結果となる部分木を作成しているのに対して，我々の提案手法は情報検索技術を元にして検索結果の部分文書を作成しているという点では異なるが，各 XML 文書に対して予め求めた結果を元に，ユーザにとって有益な検索結果を提示するために再構成を行うという点で関連研究といえる．

3. 提 案 手 法

本提案手法の概要を説明する．提案手法では既存の情報検索技術によってスコアリングされた部分文書を利用して検索結果を作成する (図 1 参照)．そのため，まずは各部分文書に対してスコアリングを行い，スコアリングされた部分文書群を得る (1)．これら部分文書群を文書番号ごとに集約し，木構造へ変換を行う (2) (注2)．その後に部分文書の抽出を行い検索結果の

(注2)：構造化文書はグラフ理論の木構造と交互に変換可能である．なお，部分

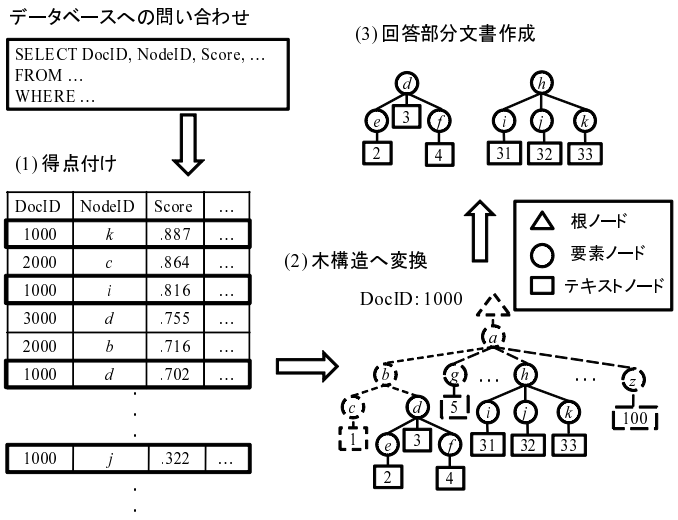
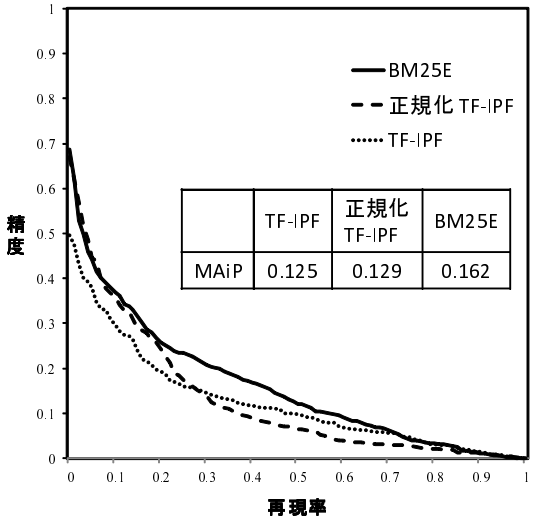


図 1　回答部分文書作成手順



	TF-IPF	正規化 TF-IPF	BM25E
MAiP	0.125	0.129	0.162

図 2　精度比較 (情報検索技術)

再構成を行う (3)．なお，今後検索結果を再構成して得られる部分文書を回答部分文書 (Answer XML Fragment, AXF) と呼び，XML　木のテキストノードに割り振られている数字は，XML 文書中で出現するテキストノードの順序を表す．

3.1　スコア付与のための部分文書検索技術選定

我々の提案手法を用いて検索結果を提示するにあたり，まずは効果的なスコアリング手法を選定する必要がある．なぜなら，我々の提案手法は部分文書に付与されたスコアを元に検索結果の再構成を行うため，部分文書に対して適切にスコアリングされていなければその後の再構成も適切に行うことができないと考えられるためである．従って，複数個の部分文書検索技術による評価実験を行い，もっとも効果的なスコアリングが可能となる部分文書検索技術の調査を行った．なお，評価実験におけるテストコレクションについては 4.1 で後述する．評価実験に用いた部分文書検索技術はそれぞれ，TF-IPF 法, 正規化 TF-IPF 法, BM25E であり，その際 BM25E のタグの重みはすべて 1 に統一した．これら情報検索技術では検索結果として不適切であると考えられる小さすぎる部分文書には大きな得点が付与されないように規制するなど，さまざまな取り組みが行われている．評価尺度には MAiP (Mean Average interpolated Precision) を利用した．MAiP とは，0% ～ 100% の再現率を一定の間隔で分割し，それぞれの再現率の点ごとの精度である iP を，全再現率点の平均を取った値である．本稿では，再現率を 1% 刻みに取る，計 101 点に対して MAiP を求めた．評価実験の結果，BM25E がもっとも高精度を示した (図 2 参照)．よって，以後提案手法で用いる部分文書検索技術は BM25E を利用することとする．

3.2　回答部分文書作成手法

回答部分文書の作成方法について説明する．1. でも述べた通り，クエリに対する適合部分を一つの部分文書で表すことが適切であるとは言い切れない．例えば，クエリに対する適合部分が文書中の離れた場所に複数存在する場合に，いずれの部分も含む部分文書を検索結果とすれば部分文書中にクエリに対して適合しない部分が多く含まれる結果になり，更に，テキストサイズが大きすぎる部分文書が抽出される可能性も存在する．それに対して，いずれかのみを含む部分文書を検索結果とした場合には別の適合部分を抽出できていないという問題が発生する．例えば，図 1 の (2) の木構造で表された文書のうちクエリに対する適合部分が (3) のようにノード d とノード h を根とする完全部分木であったとする．この場合に一つの部分文書で検索結果を表そうとすると問題が起こる．ノード a を根とする完全部分木を解とすれば不適合部分を多分に含み，更に大きすぎる部分を検索結果とすることになる．しかし，ノード d もしくはノード h を根とする完全部分木のいずれかを抽出すれば他の適合文書を抽出できないという問題が起こる．従って本提案では，適切な文書サイズ以下に制限しつつ，一つの文書からクエリに適合する複数の部分文書を抽出することで上記の問題を解決しつつ検索精度の向上を目指す．

また，部分文書検索技術によるスコアリングでは，クエリキーワードが出現する部分文書に対してスコアが加算される．そのため，クエリキーワードがほとんど出現しない部分文書に高いスコアが付与されることはないが，クエリキーワードが出現しない部分文書が一概に不適合部分文書となるわけではない．つまり，高スコアが付与された部分文書のみがクエリに対する適合部分となり得るとは限らない．例えば，仮に低いスコアが付与された部分文書であったとしても，クエリキーワードが頻出する箇所の付近の部分文書が適合部分文書である可能性があると考えられる．このような低スコアが付与されたの適合部分は従来の部分文書検索技術では取りこぼされている可能性があったが，提案手法では最大限の適合部分の抽出を試みる．更に，回答部分文書をユーザに提示する際には抽出された部分文書から効果的に検索結果を作成しなければならないため，回答部分文書に挿入される部分文書を整形する必要がある．以上より，

要件 1　クエリに対する適合部分を抽出し，適切なテキストサ

文書は木構造において対応するノードを根とする完全部分木と一致する．

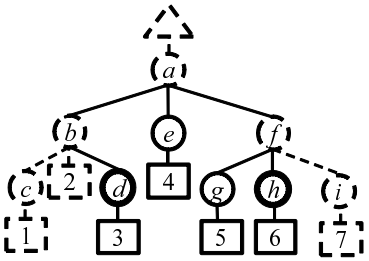


図 3　部分文書結合例

イズの回答部分文書を作成する

要件 2　スコアの低い適合部分である “隠れた正解” を抽出する

要件 3　部分文書群を集約し，ユーザに提示する検索結果に整形する

の三つの要件を考慮し回答部分文書を作成する．続いて，これらの要件を満たすための取り組みについて述べる．

3.2.1　部分文書抽出処理

要件 1 を満たすためには，回答部分文書を作成するために部分文書を抽出する際にそのテキストサイズを考慮する必要がある．以後，これらのために行う処理を **部分文書抽出** と定義する．部分文書抽出処理では，スコアリングされた部分文書群を文書番号ごとに纏めて取り出し，大きなスコアが付与された部分文書から順番に回答部分文書に対して挿入を行う．その際，部分文書の保持するテキストノードに分解して回答部分文書に挿入を行う．なお，部分文書抽出処理は，回答部分文書に含まれるテキストノードのテキストサイズが抽出閾値 EL を超えない限り繰り返される．なお，EL は以下の式で表される．

$$EL = \alpha \cdot textsize(root) \tag{1}$$

ただし，α は一つの文書中に含まれると考えられる適合部分の割合を表すパラメータであり，$textsize(root)$ は回答部分文書の作成対象の XML 文書のテキストサイズ，すなわち全索引語数である．

3.2.2　部分文書結合処理

要件 2 で抽出することを目標としている “隠れた正解” は適合部分付近に存在すると考えることができる．そのため，“隠れた正解” のうちの一つであると考えられる適合部分と適合部分の間の “隠れた正解” の抽出を試みる．なお，その際の処理を **部分文書結合** と定義する．部分文書結合処理は，回答部分文書に対して挿入がおこわなれた際に，挿入されたテキストノードと既に挿入されているテキストノードのうちもっとも距離の近い任意のテキストノードの結合コストが結合閾値 CL よりも小さければ部分文書結合を行い，それぞれの部分文書間に存在する部分文書を回答部分文書に挿入する．図 3 を用いて具体例を示す．仮に回答部分文書にノード d とノード h が挿入されていて，これらのノードが結合するとする．この場合，ノード d とノード h の間に存在するのはノード e とノード g であるため，これらのノードを回答部分文書へ挿入を行う．

ここで，部分文書が結合する条件を説明する．任意のテキストノード n, m の結合コスト $CCost(n,m)$ は以下の式で表される．

$$CCost(m,m) = |position(n) - position(m)| \tag{2}$$

ただし，$position(n)$ は，その文書中のテキストノード n が，文書順で辿ったときに何番目に出現するテキストノードかを示す．$CCost$ の具体例を図 4 に表す．ノード a と ノード b の持つテキストノードの最小値の絶対値は $|1-2| = 1$ なので，$CCost(a,b) = 1$ となる．同様に $CCost(c,d) = |1-3| = 2$, $CCost(e,f) = |1-2| = 1$ と求められる．

$CCost(a,b) = 1$
$CCost(c,d) = 2$
$CCost(e,f) = 1$

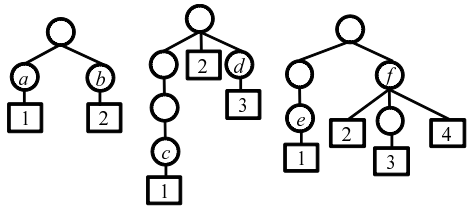


図 4　Ccost 計算例

3.2.3　部分文書統合処理

要件 1 と要件 2 を満たすために複数の部分文書の抽出，結合を行うが，それらの部分文書を最終的な検索結果として提示する形式に整形するためにはそれらの抽出された部分文書群を統合しなければならない．よって，要件 3 を満たすための処理，**部分文書統合** の定義を行う．なお，統合が必要となる場合とは即ち，

（1） 挿入される部分文書の子孫が既に回答部分文書に挿入されている場合

（2） 挿入される部分文書の祖先が既に回答部分文書に挿入されている場合

（3） ある部分文書に含まれる全ての子部分文書が回答部分文書中に存在する場合

である．前述の通り，従来の XML 検索では最もスコアの高い部分文書を最適解として扱った．そのため，複数の部分文書を抽出する際には部分文書の重複を許容せずに抽出するのが自然だと考えられる．しかし，提案手法ではより多くの適合部分を抽出することを目的としているため，要件 1 と要件 2 を満たす限りは (1) のような場合には先祖である部分文書を回答部分文書へ挿入を行い，(2) のような場合には，挿入を行ったところで回答部分文書への影響は全くないため，処理を行わない．

また，回答部分文書を部分文書で表現する際には極力少ない部分文書数で表現する必要がある．なぜなら，我々が評価実験の際に利用する評価ツールは，クエリごとにシステム対して渡すことができる部分文書数に制限があるため，より多くの適合部分を抽出するためには無駄なく部分文書を選択する必要があるためである．よって，(3) のような場合には部分文書を集約して一つの部分文書に纏める．図 5 を用いて具体例を示すと，ノード b とノード c が回答部分文書に挿入されているのであればこれらのノードを集約してノード a として表現する．

3.3　回答部分文書作成例

以降，図 6 を例に文書番号 1000 の回答部分文書作成手順の説明を行う．なお，図 6 の左のリストは，一つ目のステップに

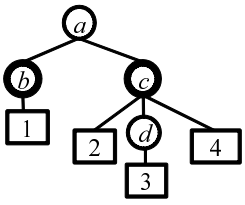


図 5　部分文書の集約

おいて得られた部分文書群から文書番号 1000 の部分文書を取り出したものであり，今回の回答部分文書作成においてはパラメータ $\alpha = \frac{1}{3}$, 抽出閾値 $EL = \frac{1}{3} \cdot 300 = 100$, 結合閾値 $CL = 3$ とする．まずは部分文書抽出処理を行う．リスト中でもっとも大きなスコアが付与されている部分文書はノード k を根とする完全部分木 (以降，任意のノード n を根とする完全部分木を単にノード n と表現する．) であり，ノード k の持つテキストサイズは 40 ($< EL$) であるため，回答部分文書にノード k に含まれるテキストノード 33 を挿入する (1)．回答部分文書に挿入が行われたので，続いて部分文書結合処理が行われるが，回答部分文書にはテキストノードが存在しないため部分文書結合処理は中止される．また，回答部分文書に含まれるテキストサイズは 40 ($< EL$) であるため，引き続き部分文書抽出処理が行われる．ノード k の次に大きなスコアが付与されているノート i を挿入した場合の回答部分文書のテキストサイズは 50 ($< EL$) であるため，ノード i のテキストノード 31 を新たに回答部分文書に挿入する (2)．なお，続いて行われる部分文書結合処理において $CCost(i,k) = |6-8| = 2 (< CL)$ であり，ノート i とノード k を結合した際の回答部分文書のテキストサイズは 70 ($< EL$) であるためノードは結合される．よって，ノード i と ノード k の間に存在するノード j のテキストノード 32 が新たに回答部分文書に挿入される．このとき，回答部分文書に含まれるすべてのノードが既に結合しているため，部分文書結合処理は中止される．同様に，部分文書抽出処理によってノード d の挿入に成功するが，$CCost(d,i) = |31-4| = 27 (> CL)$ のために結合処理は失敗する．その後のノード b, ノード c の挿入が試みられるがいずれも失敗し，最終的に回答部分文書はノード d と ノード h となる．

4. 評 価 実 験

4.1 テストコレクション

評価実験には，2008 年版の INEX (INitiative for the Evaluation of XML Retrieval) テストコレクションを用いた．INEX テストコレクションは，XML 部分文書検索のための国際プロジェクトである INEX Project によって 2002 年より構築作業が行われている XML 部分文書検索システム用の性能評価データセットである．検索対象となる約 66 万個の XML 文書で構成される 2008 Wikipedia document collection)，285 個のクエリの集合 INEX topics，それらクエリに対する解答部分文書集合及びその評価が記述されている INEX relevance assessments の三つで構成されている．なお，INEX topics には XPath とクエリキーワードを指定する CAS (Content-And-Structure) クエリと，クエリキーワードのみを指定する CO (Content-Only) クエリの二種類が存在する．

NodeID	Score	Test size
k	.887	40
i	.816	10
d	.702	25
j	.322	15
h	.256	70
b	.207	40
a	.194	300
c	.155	15

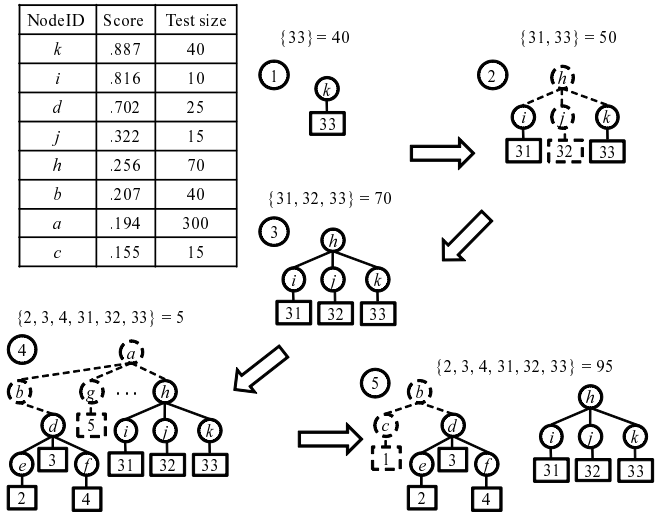


図 6　回答部分文書作成例

表 1　α の変動による精度への影響

α	0.1	0.2	0.3	0.4	0.5
MAiP	.0284	.0505	.0757	.0935	.107
α	0.6	0.7	0.8	0.9	1.0
MAiP	.122	.131	.137	.139	.236

表 2　CL の変動による精度への影響

CL	0	1	2	3	4	5	6	7
MAiP	.1793	.1795	.1795	.1795	.1795	.1795	.1795	.1795
CL	8	9	10	20	30	40	50	60
MAiP	.1796	.1796	.1796	.1798	.1800	.1800	.1801	.1802

4.2 予 備 実 験

提案手法の有効性を確認するための評価実験を行うにあたり，各種閾値を設定するための予備実験を行った．まずは，部分文書抽出の際の抽出閾値 EL を設定するため，パラメータ α を 0.1 から 1 までの 0.1 刻みでの精度評価実験を行った．その結果，$\alpha = 1$ において最高精度を示した (表 1 参照)．この結果から，一つの文書に含まれている適合部分の割合は一定以下の割合であると仮定するのではなく，よりスコアの高い部分文書を抽出することがより効果的であるということが判明した．

続いて部分文書結合の閾値 CL を求めるための実験を行った．“隠れた正解” は近接した部分文書同士の間に存在すると考えるため CL の値は比較的小さい値を採用することを想定しているが，テストコレクションにおける各文書の平均テキストノード数が 63.7 であることを考慮し，最大 60 の距離においても部分文書の結合を行うように CL のパラメータを設定して実験を行った．なお，部分文書結合単独での精度を調査することが目的であるため，この実験では部分文書統合処理は起こらないよう設定を行った．その結果，部分文書結合を行わなかった場合 ($CL = 0$) と部分文書結合を行った場合とでほとんど精度が向上しなかった．精度向上を達成できなかった原因として，

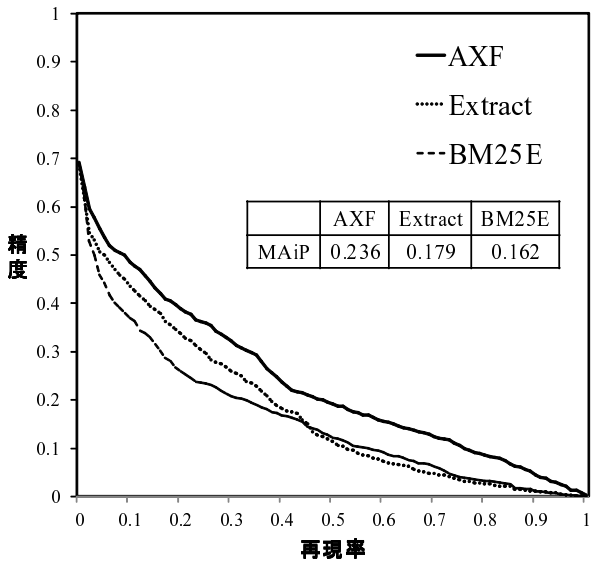


図 7　精度比較 (検索結果)

近接するスコアの高い部分文書があれば，その親部分文書も抽出されているため，結局のところ部分文書結合の如何に関わらず抽出を行うことができているからであると考えられる．また，CL が大きくなるにつれて精度が微妙に変化している理由としては，CL の値を大きくなるほど結合可能な部分文書の選択肢が増えるため，抽出制限の上限に少しでも近づくように部分文書を抽出できるからであるためであると考えられる．いずれにせよ，この僅かな精度向上が “隠れた正解” を発見した結果であるとは考えにくく，今後 “隠れた正解” を抽出する上では他のアプローチを考える必要がある．なお，今回の提案手法では，部分文書同士の結合を行うかどうかを判断する際の結合コストは文書順で辿った場合のテキストノードの番号を用いている．しかし，これでは XML 文書の持つ構造を考慮しているとはいえないという問題が存在する．よって，今後部分文書間の距離を計算する際には文書構造も加味して考慮する必要があると考えられる．

4.3 評価実験

4.2 を踏まえ提案手法と従来手法の検索精度の比較を行ったところ，提案手法 (AXF) は従来手法 (BM25E) と比較して全ての再現率点の精度において常により高い精度を保っており，全体では 46% 程度精度の上昇が見られた (図 7 参照)．なお，部分文書統合を行わずに重複しないように部分文書抽出を行った手法 (Extract) と比較すると 32% 程度精度が向上していることから，部分文書の上書きや集約などの操作が精度向上に大きく貢献していることが判明した．また，BM25E と Extract を比較すると Extract が 10% 程度高い精度を示していることより，一つの文書から複数の部分文書を検索結果として抽出することで効果的な検索に結びつけることが可能であるということが判明した．

5. おわりに

本稿では，従来の部分文書検索技術によって得点化された部分文書を再構築することで効果的な情報提示を目指す手法を提案した．その際，検索結果のテキストサイズを制限しつつ，従来の部分文書検索手法では発見できていない “隠れた正解” の抽出を行い，抽出された部分文書群を整形することでより多くの適合部分を抽出することを目指した．

今回提案した部分文書抽出制限や部分文書結合は今後更に精度向上を実現する上で再考の余地があるが，抽出された部分文書群を整形することで大きく検索精度を向上させることができた．これにより，情報検索技術によってスコアリングされた部分文書群に対して二次的に検索結果を再構成を行うことで精度を向上させることが可能であるという知見が得られた．

今後の課題として，部分文書同士が結合される際の条件について再考を行う必要がある．

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 若手研究 (B) (課題番号: 20700227) によるものである．ここに記して謝意を表す．

文　献

[1] Vagelis Hristidis and Nick Koudas. Keyword proximity search in xml trees. *IEEE Transaction on knowledge and data engineering*, Vol. 18, No. 4, pp. 525–539, April 2006.

[2] Yu Huang, Ziyang Liu, and Yi Chen. Query Biased Snippet Generation in XML Search. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, pp. 315–326, June/July 2008.

[3] Guoliang Li, Jianhua Feng, Jianyong Wang, and Lizhu Zhou. Effective keyword search for valuable lcas over xml documents. In *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pp. 31–40, New York, NY, USA, 2007. ACM.

[4] Ziyang Liu and Yi Chen. Identifying Meaningful Return Information for XML Keyword Search. In *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*, pp. 329–340, June 2007.

[5] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schutze. *Introduction to Information Retrieval*. Cambridge University Press, July 2008.

[6] Albrecht Schmidt, Martin Kersten, and Menzo Windhouwer. Querying xml documents made easy: Nearest concept queries. In *17th International Conference on Data Engineering (ICDE'01)*, p. 321, April 2001.

[7] Yu Xu and Yunnis Papakonstaninou. Efficient Keyword Search for Smallest LCAs in XML Databases. In *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*, pp. 527–538, June 2005.

[8] 高見真也, 田中克己. 類似性を考慮したスニペットの再生成による検索結果のパーソナライズ. 第 18 回データ工学ワークショップ (DEWS2007) 論文集, March 2007.