構造化文書の重要部分抽出のためのスコアリングアルゴリズム

欅　悼志† 波多野賢治† 宮崎　純††

† 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,hatano}@ilab.doshisha.ac.jp, ††miyazaki@is.naist.jp

あらまし　本稿では，構造化文書からの重要部分抽出の手法に関して述べる．XML 文書に対する従来の検索技術では，検索における計算コストを抑えることに主眼が置かれてきたため，XML 情報検索においては依然として精度における問題が残っている．実際，既存の構造化文書に対する検索システムの精度は，一般的に利用されている Web 検索エンジンの精度には到底及ばず，未だ実用的なレベルに達していない．そこで本稿では，重要部分抽出に関する研究から得られた知見を利用した三種類のスコアリング法と，それらのスコアを統合するためのスコア統合法を提案する．評価実験を行った結果，三種類のスコアリング手法のうち一種類のみその有用性が確認できたが，残る手法の改善を行う際に，部分文書が持つ索引語の統計量を用いたスコアリング方法を考案することで精度の向上が期待できるという知見が得られた．

キーワード　構造化文書，XML，重要部分抽出，スコアリングアルゴリズム

Algorithm for Extracting Important Parts from Structured Documents

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

† 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,hatano}@ilab.doshisha.ac.jp, ††miyazaki@is.naist.jp

Abstract　In this paper, we describe an approach for extracting important parts in fragment search for structured document. We have mainly tried to saving processing costs in the past, so we still have a lot of challenges to work on in this research area. In fact, compared with Web search engines, search engines for structured document do not be developed to a practical level in their information retrieval precision. Therefore, we propose three types of scoring algorithms and algorithm for integrating these scores to fulfill the requirements for extracting important parts from documents. In our experimental evaluation, we found only one score algorithm work well; we can expect to improve the accuracy by reforming the algorithms, which algorithms do not work before, with using the statistics values of nodes about index.

Key words　structured documents, XML, extraction of XML fragments, scoring algorithm

1. はじめに

従来の情報検索技術では，複数のファイルフォーマットが混在した文書群に対して一度にまとめて検索することはできず，それぞれのフォーマットごとにそれぞれの検索手法で検索を行う必要があった．これに対して，構造化文書の一つである XML (Extensible Markup Language) [1] はデータ交換のための標準フォーマットであるため，文書が XML に従っていれば Web 文書のフォーマットであっても，動画コンテンツのフォーマットであっても，もしくは企業の商品管理データのフォーマット

であっても，すべて一つの検索手法で検索を行うことができるという利点を持つ．そのため，XML 文書に対する情報検索に関する研究は，現在盛んに行われている．

XML 文書に対する情報検索では，部分文書単位を検索粒度とした検索を行うことができる．本稿における部分文書とは，XML 文書を木構造に変換した場合に，あるノード以下を一つの文書として見立てたものであり，それらの部分文書すべてを検索対象とすることで，文書単位よりも更に細かな粒度に細分化した単位での検索が可能になる．このように部分文書単位で文書検索を行うことで，従来の文書単位を検索粒度とした情報検索では見落とされる可能性があった，文書全体のうちのごく小さな部分に存在する，ユーザの情報要求を満たす部分，すなわち正解部分を発見することができるようになる．また，ユーザの求める正解部分は文書中のごく一部のみであると考えられるため，正解部分が含まれる文書，すなわち正解文書が正しく提示されたとしても，ファイルサイズの大きな文書が提示された場合にそこから正解部分を抽出することはユーザにとって大きな負担となるという問題が存在する．このような問題に対しても，部分文書単位に対する検索では，検索システムによって判断された，ユーザの求める粒度での文書中の重要部分を提示することで，文書から正解部分を抽出する際に大きな手助けになると考えられる．

一方，部分文書単位での情報検索はその性質上一つの文書から複数の部分文書が取り出せるため，検索対象となる部分文書数が膨大になってしまい，検索速度が低下するということが欠点である．構造化文書に対する情報検索には，検索速度の向上を目的とした効率的な検索を目指す研究と，検索精度の向上を目的とした効率的な検索を目指す研究が存在するが，これまでの構造化文書に対する研究では，主に前述の欠点を克服するために効率化を目指す研究が取り組まれてきた．効率化を目指す研究では，計算量の削減のために，特定の条件を満たすノードをクエリに対する最適部分と仮定している．これに対して，高精度検索を目指す研究においては，各部分文書に対してスコアリングを行い，高スコアと判定された部分文書をクエリに対する最適部分としているが，このスコアリング法は重要部分抽出を目的としていない．そのため，これら検索速度の向上を目指す研究や検索精度の向上を目指す研究いずれにおいても，それぞれの手法によって重要部分と判断された部分文書が実際にクエリに対して最適部分であるとは言い切れない．

そこで本稿では，情報検索の立場から精度重視のアプローチを目指すために，重要部分抽出に関する研究で得られた知見を利用した構造化文書に対する情報検索技術の実現を目的とする．その際，従来の検索精度の向上を目指す研究と同様に構造化文書の持つ構造的特長を利用しつつ，重要部分抽出の要件を満たすようにスコアリング法を設定する．本提案手法の実現によって，これまで考慮されていなかった重要部分の要件を考慮した構造化文書に対する情報検索が可能となり，ユーザのクエリに対し検索結果を返す際に，正解文書を提示するだけではなく，ユーザの情報要求に合致すると思われる部分を文書中から抽出してユーザに提示できるため，文書中から正解部分を抽出する際のユーザの負担を軽減することができる．

以下，2. では本論文の関連研究について，3. では提案手法を述べる．また，4. では評価実験を行った結果とそれについての考察を行い，5. ではまとめと今後の課題について述べる．

2. 関 連 研 究

2.1 構造化文書の情報検索における関連研究

XML 文書には，データ指向 (data centric) XML と呼ばれる一つのテキストノードに含まれる索引語数が少ない XML 文書と，文書指向 (document centric) XML と呼ばれる一つのテキストノードに含まれる索引語数が多い XML 文書の二種類存在する．データ指向 XML は各ノードに入っている値が一つのみであるため，クエリに対する正解部分を決定することが比較的容易である．そのため，データ指向 XML に対する検索では効率的な検索，つまり主に検索速度の高速化を目指している．これに対して，文書指向 XML はクエリに対する正解部分を決定することが比較的困難であり，効果的な検索，つまり検索精度を向上させる研究が進められている．このような背景から，XML 文書検索の研究は以下の節で述べるように，主として効率的な検索および効果的な検索の二種類が行われている．

2.1.1 効率的な検索を目指す研究

1. でも述べたように，部分文書に対する情報検索を行う場合，結果として検索速度が低下するという問題が生じている．そのため，これまでの XML 文書に対する情報検索の研究は，検索処理の効率化を目的とした研究が中心であった．その際，LCA (Lowest Common Ancestor) と呼ばれる概念が頻繁に用いられている．LCA と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．これらの研究では，LCA をクエリに対する最適部分であると仮定することで各部分文書のスコア計算を省略しており，その結果，検索速度の高速化を実現することができるようになっている．

LCA の具体例を図 1 で示すと，クエリキーワードが k_1，k_2，k_3 であった場合，LCA は点線で表されているノード 2 とノード 6 となる．ノード 1 を根ノードとする部分木には，k_1，k_2，k_3 が含まれるが，ノード 1 の子孫ノードのノード 3 やノード 6 を根ノードとする部分木にすべてのクエリキーワードが含まれているため，ノード 1 は LCA ではない．

一方，LCA を根とする部分木内で，クエリキーワードを含む部分のみから構成された部分木は MCT (Minimum Connection Tree) と呼ばれる [5]．図 2 の ノード 2, 6 を根ノードとする部分木のうち，点線部分を除く実線部分で構成される部分木がノード 2, 6 における MCT である．MCT は接続最小木と呼ばれ，LCA を根とする部分木からユーザのクエリに対する最適部分を抽出する際に最低限含むべき部分を表している．

この LCA は，XML 問合せ言語 XIRQL [3] を始め，構造化文書に対する情報検索に関する多くのシステムにおいて利用されており，また，LCA や MCT の概念を効率的に見つけ出すアルゴリズムも多く考案されている [11]．LCA を利用した研究の一つである XSeek では，構造化文書に対しての検索におい

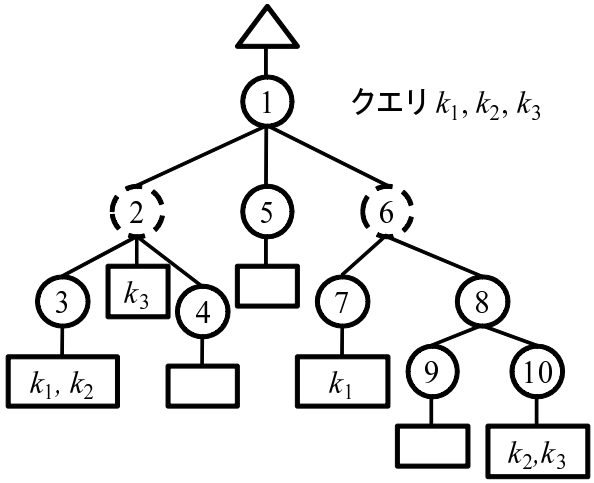

図 1　LCA

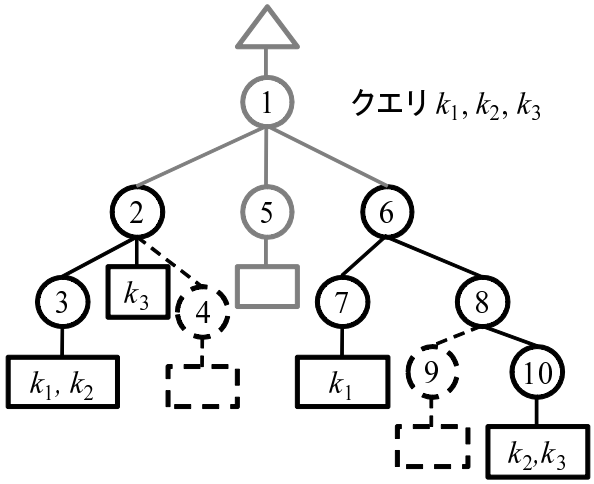

図 2　MCT

て効率化のみではなく高精度検索の実現をも目指している [6]. この XSeek では二つの目標を掲げており，検索対象部分を特定することと，クエリを判断して理想的な検索結果を出力することである．従来の効率化を目指す手法では，解として返ってきた結果の中にクエリに対して不必要な部分も含まれるため，XSeek では検索結果として最適な部分を特定している．一つ目の目標である検索対象部分の特定に LCA を利用しているが，二つ目の目標である理想的な検索結果の出力には，MCT を中心にクエリに対する解として相応しい部分を抽出している．さらに XSeek では，単に LCA もしくは MCT を抽出したものをクエリに対する最適部分とすることは高精度検索を目指す上では不十分であると考え，部分文書やクエリの構造情報を利用して重要部分を選択している．LCA を根とする部分木を，クエリに対する最適部分としてそのまま使うのではない点では従来の研究と XSeek は異なるが，LCA 以下の部分木中に重要部分となり得る部分が含まれていると仮定しているという点では他の研究と同様である．

2.1.2　効果的な検索を目指す研究

構造化文書に対する情報検索では，文書に対する情報検索技術として用いられてきた TF-IDF 法 [10] を拡張し，索引語が出現するノードへのパスを考慮した索引語の頻度情報から求める TF-IPF 法 [4] が用いられることが多い．TF-IPF 法 (Inverse Document Frequency) とは，部分文書内での索引語の重要度を表す TF (Term Frequency) 値と，あるパスにおいて索引語自体の重要度を表す IPF 値を掛け合わせることで得られる，それぞれの部分文書の索引語ごとの重要度を表す TF-IPF 値を利用することで，クエリに対して適切な部分文書を選択する手法である．TF-IPF 法を用いた構造化文書に対する情報検索は，高精度検索において大きな成果が上がっている [12] が，複雑な計算式を用いる必要があることや，検索を行う前に予め多くの処理を行っておく必要があるためデータの更新が困難であるという問題がある．また，先にも述べたように TF-IPF 法は，文書を抽出するための情報検索技術を構造化文書検索に適応させたものであるため，文書中から重要部分を抽出する際に必要な要件を考慮しているとは限らない．これは 2.1.1 で述べた LCA を利用して生成した部分を重要部分として取り扱っている研究も同様である．これらは重要部分抽出に関する研究を踏まえているとはいえず，構造化文書の重要部分抽出を目的とした情報検索を達成するためには重要部分抽出における研究を考慮する必要がある．なぜなら，文書中からもっともクエリに相応しい部分文書を抽出する際に，その部分文書は文書中の重要部分に相当すると考えられるため，重要部分抽出の際の要件を満たしている部分文書はクエリに対する最適部分となると考えられる．よって，部分文書単位を検索粒度とした検索ではそれらの要件を考慮する必要があると考えられるためである．

2.2　重要部分抽出に関する関連研究

情報検索に関するさまざまな研究が包括的に纏められている最近の書籍 Introduction to Information Retrieval (IIR) [8] によると，ユーザが情報検索を行う際には，検索エンジンによって返された検索結果すべてを確認するのではなく，検索結果の一覧から情報要求を満たすページを最終的にユーザ自身で選択することを望んでいるため，いずれのページがユーザにとって有用であるか判断するために必要十分な情報を検索エンジンによって要約し，提示する必要があると述べている．この提示される情報はスニペット (Results Snippet) と呼ばれるが，このスニペットの抽出には三つの要件があると IIR は述べている．具体的には，

- クエリキーワードに関する最大限の情報を盛り込む
- 短くまとめて要約する
- 文書中の用語を用いる

の三つである．ここで，スニペットがユーザの要求を満たす最適な部分，すなわち，重要部分であると言い換えられるとすると，それらの要件は重要部分抽出の際に考慮すべき要件とも言える．これらを考慮すると，索引語頻度から得られる TF-IPF 値が先ほどの三つの要件を必ずしも満たしているとは言い切れず，すべてのクエリキーワードを含むノードのうち最もテキストノードに近いということのみ保証されている LCA やその一部分である MCT も同様である．つまり，TF-IPF 値が高い判定された部分文書や LCA を根ノードとする部分文書，そして MCT は，ユーザのクエリに対する最適部分とは言い難く，高

精度検索を目指すためには，あらゆるノードが最適部分の根として返される可能性を踏まえて検討する必要があると考えられる．そのために，重要部分抽出の際の要件を考慮したスコアリング法を提案した．スコアリング法の詳細に関しては 3. で詳述する．

3. 提案手法

2.2 で述べたように，文書中から重要部分，すなわち，ユーザの情報要求を満たす部分文書を抽出するためには上述の三つの要件を考慮しなければならない．また，それらの要件を構造化文書検索において実現するためには，いくつかの課題が存在する．以下では，それぞれの課題について述べる(注1)．

（1） 情報検索技術では，クエリキーワードの出現率が高いほど，その部分文書のスコアが高くなるように設定されている．これは，すなわち先ほど出てきた TF の考え方であるが，重要部分抽出の際には，仮に部分文書中においてクエリキーワードが頻出していたとしても，それが特定のクエリキーワードのみが出現している場合には，その部分文書が重要部分とは言い切れない．なぜなら，クエリキーワードが複数の用途で用いられる単語であった場合に，クエリキーワードが文書中で意味している内容とユーザのクエリで意図された内容が等しいのか判断できないためである．なので，このような問題を解決しなければならないという課題が存在する．

なお，この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なければならないという課題は，要件「クエリキーワードに関する最大限の情報を盛り込む」を満たすために解決する必要がある．

（2） 部分文書中に不要な情報が大量に含まれていると，ユーザのクエリに対する最適部分とは言い難い．なぜなら，重要部分は，文書が有益であるのかユーザが判断する際に必要十分の情報を要約して提示することが目的であるため，クエリに無関係な部分が多く含まれる部分文書は重要部分に相応しいとはいえないからである．

なお，このクエリに無関係は部分の割合を考慮しなければならないという課題は，要件「クエリキーワードに関する最大限の情報を盛り込む」を満たすために解決する必要がある．

（3） 部分文書が大きさを考慮し，必要以上に大きすぎる部分がクエリに対する最適部分とならないようにしなければならないという課題が存在する．なぜなら，(2) でも述べたように，重要部分とは文書が有益であるのかユーザが判断する際に文書の一部分を掻い摘んで提示するためのものであるので，重要部分が大きいと本末転倒となるためである．

なお，この課題部分文書の大きさを考慮しなければならないという課題は，要件「短く纏めて要約する」を満たすために解決する必要がある．

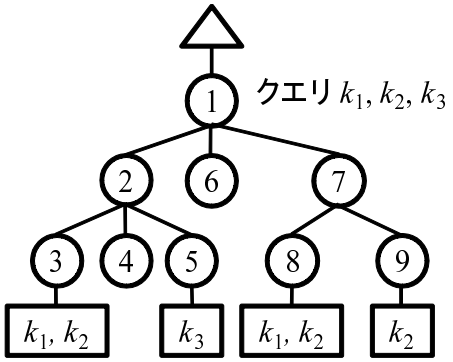

図 3 クエリキーワードの含有率の計算

以上，三つの課題を解決できれば重要部分抽出の要件を満たすことができると考える．そこで，本章では上記の三つの課題に対して，一つの課題につきそれを解決するためのスコアリング法を一つ提案し，計三種類のスコアリング手法を用いて部分文書検索におけるスコアリングに利用する．以下の節では，これらスコアリング法の詳細を述べる．

3.1 キーワード含有率スコア

まず，先ほど挙げた課題 1 の解決法について述べる．前述のとおり，ユーザが入力したクエリキーワードが頻出する部分文書は，一般的に検索結果として適切である可能性が高いと考えられる．しかしながら，ユーザによって検索エンジンに入力されたクエリキーワードのうち一部のクエリキーワードのみが頻出する部分文書がクエリに対して最適であるとはいえない．したがって，多くの種類のクエリキーワードを含んだ部分文書ほど大きなスコアを与える必要がある．

これは，ユーザによって指定されたクエリキーワードの種類数と部分文書中に含まれているクエリキーワードの種類数によって求められ，本稿では**キーワード含有率スコア**と呼ぶ．当然ながら，ユーザによって指定されたクエリキーワードの種類を多く含んでいる部分文書のキーワード含有率スコアがより大きくなるように設定している．

本稿では，構造化文書中のあるノードに着目し，それを根ノードとする部分木に含まれているクエリキーワードの種類と，それらを含んでいるテキストノードに着目してキーワード含有率スコアを以下のように定義する．

$$s_i(x) = \frac{v(x)}{n} \tag{1}$$

ただし，$s_i(x)$ をあるノード x のキーワード含有率スコア，$v(x)$ をノード x を根ノードとする部分木に含まれるクエリキーワードの種類の数，n をクエリに使用されたクエリキーワード数をとする．図 3 を用いて，部分文書のうちノード 1, 2, 7 を根ノードとする部分文書に対するスコアを例に挙げて説明すると，$s_i(1) = 3/3 = 1.00$，$s_i(2) = 3/3 = 1.00$，そして $s_i(7) = 2/3 = 0.667$ となる．

3.2 キーワード寄与率スコア

次に，先述した課題 2 の解決法について述べる．3.1，3.3 で述べる，キーワード含有率スコアやエッジスコアは，クエリ

(注1)：要件「文書中の用語を用いる」にまつわる課題は特に存在しない．なぜなら，重要部分抽出には自然言語処理による文書中の内容を自動要約する手法や，本文以外の情報を抽出する手法が存在するために，重要部分として本文を変形させたものや本文以外の部分が得られる可能性がある．しかし，提案手法ではテキスト自体へ自然言語処理などは行わず，重要部分の出力は本文をそのまま用いるため，今回の場合この要件を満たしているといえる．

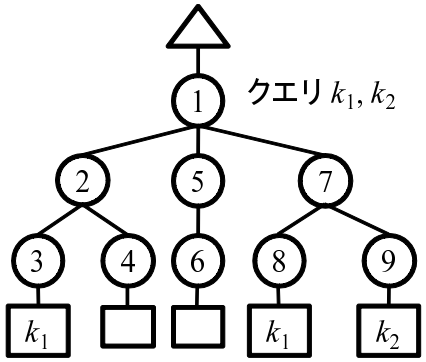

図 4　クエリキーワードの寄与率の計算

キーワードが出現した種類と位置を利用したスコアリング法であるが，これら二種類のスコアリング法では，部分木のうちクエリキーワードを含んでいないノードに関しては全く考慮されていない．つまり，部分文書中にクエリキーワードを含まないノードを多く含む場合でも，エッジスコアやキーワード含有率スコアが高くなる可能性があることを示唆している．課題 2 でも述べたように，このような不要な部分が多くを占める部分文書はユーザが入力したクエリキーワードに対する最適部分とは言い難いため，部分文書内にクエリキーワードを含んでいるテキストノードが多く含まれているほどスコアを大きくする必要がある．

このスコアリング法では，クエリキーワードが含まれている部分文書中のテキストノードの数が多いノードのスコアがより高くなるように設定した．本稿ではこのようにして計算される部分文書のスコアを**キーワード寄与率スコア**と呼ぶ．

本稿では，構造化文書中のあるノードに着目し，それを根ノードとする部分木に含まれているテキストノードの数と，そのテキストノードがクエリキーワードを含んでいるか否かに着目して，以下のようにキーワード寄与率スコアを定義する．

$$s_c(x) = \frac{k(x)}{l(x)} \tag{2}$$

ただし，$s_c(x)$ をあるノード x のキーワード寄与率スコア，$l(x)$ をノード x を根ノードとする部分木に含まれるテキストノードの数，また，それらテキストノードのうちクエリキーワードを含んでいるものの数を $k(x)$ とする．図 4 の例では，$s_c(1) = 3/5 = 0.600$，$s_c(2) = 1/2 = 0.500$，そして $s_c(7) = 2/2 = 1.00$ となる．

3.3　エッジスコア

最後に，先ほどの課題 3　の解決方法について述べる．1. でも述べたように，構造化文書の文書構造は，少なからず話題の切れ目を表現していると言える．つまり，ノード同士が離れていれば，その分だけノード間の結びつきが弱い．ノードの結びつきが弱い場合は，それらノードを一つの纏まり，すなわち一つの重要部分としてユーザに提示するのが必ずしもよいとはいえない．こうした問題を解決するためには，ノードのスコアを計算する際に，クエリキーワードを含んでいるノード間の距離を計算し，部分文書の大きさを順位付けに反映する必要がある．

ノード間の距離は，スコアを計算するノードとクエリキーワードを含んでいるノード間のエッジの数で表現することができる．このような概念から計算されるスコアを，本稿では**エッジスコア**と呼ぶ．エッジスコアが大きければ大きいほど，部分文書の大きさが小さくなるように設定している．

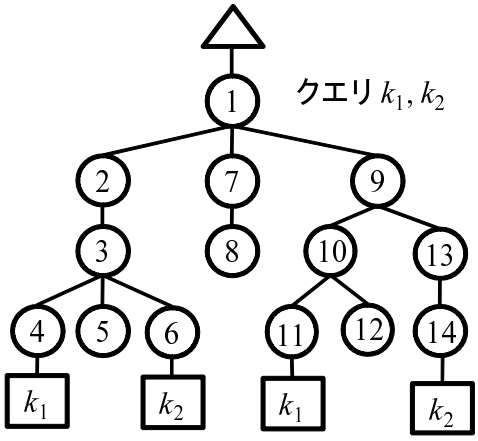

図 5　エッジスコアの計算

本稿では，構造化文書中のあるノードに着目し，そのノードの MCT に含まれるエッジの本数を利用して，エッジスコアを次のように定義する．

$$s_e(x) = \frac{1}{\log(1 + m(x))} \tag{3}$$

ただし，$s_e(x)$ をあるノード x のエッジスコア，$m(x)$ を x を根とした MCT に含まれるエッジの本数とする．図 5 の例では，$s_e(2) = 1/\log(5+1) = 0.387$，$s_e(3) = 1/\log(4+1) = 0.431$，そして $s_e(9) = 1/\log(6+1) = 0.356$ となる．

3.4　スコアの正規化と統合

ここまでで述べた三種類のスコアリング法を基に最終的な部分文書の評価を行うための統合方法に関して述べる．鈴木によると，複数のスコアリング法によって得られたスコアを統合したものを検索結果としてユーザに提示するメタ検索エンジンにおいて，スコアを統合する際に予め各スコアを正規化しておく必要があり，正規化を行うことは検索精度に大きな影響を及ぼす [13]．よって，本提案手法においても三種類のスコアリング法によって得られた各スコアの統合を行う前に，予め各スコアに対して正規化を行う．代表的なスコアの正規化手法には，

- **Standard:** スコアの最小値を 0，最大値を 1 に揃える
- **SUM:** スコアの最小値を 0，総和を 1 に揃える
- **ZMUV:** スコアの平均を 0，分散を 1 に揃える

が存在する [9] が，本提案手法で得られる各スコアはその分布に大きな偏りがあるため，以下の数式を用いて，ZMUV による正規化を行った．

$$s_n(x) = \frac{s(x) - \bar{s}}{\sigma_s} \tag{4}$$

ただし，スコアリング法によって得られた各部分文書に対するスコア群 s のうち，x 番目のスコア $s(x)$ の正規化後のスコアを $s_n(x)$，s の平均を $\bar{s}$，s の標準偏差を σ_s とする．

複数のスコアを統合するという点では提案手法はメタ検索エ

ンジンと同様であるが，これらメタ検索エンジンにおいて統合される際に用いられる各スコアリング法はそれぞれ単体でも検索エンジンとして利用可能であるのに対し，提案手法は解決すべき複数の課題に沿って，それぞれ異なる視点からスコアを求めているために，特定のあるスコアリング法でのみスコアが高い部分文書は要件すべてを満たしているとは考えにくい．よって，そのような部分文書が高スコアに順位付けされないようにしなければならない．従って，順位付けによって得られたスコアを統合する際に，メタ検索エンジンの各スコア統合において得られた知見を用いる際は慎重に行う必要がある．今回は，すべてのスコアの重みを均等に反映させるために，正規化されたスコアの和を各部分文書に対する最終的なスコアとした．

4. 評価実験

4.1 実験の目的と評価尺度

提案手法の妥当性を検証するため，後述するテストコレクションを用いて評価実験を行った．また，従来手法 [12] と検索精度を比較することで，提案手法の優位性を検証した．その際，評価尺度には MAiP (Mean Average interpolated Precision) [7] を用いる．本評価実験における MAiP とは，情報検索において従来から用いられてきた再現率と精度から求められる値であり，x 軸に再現率，y 軸に精度を用いて描かれるグラフ，つまり再現率–精度グラフにおける再現率が 0% から 100% までの一定の区間ごとの各点における精度の値を求め，すべての点での精度の値を平均した値である．今回の実験では，適合率を 1% 刻みに取る，計 101 点に対して MAiP を求めた．

4.2 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(注2) 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 [2]，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．

4.3 評価実験

INEX テストコレクションに含まれている 285 個のクエリに対して 3. で述べたスコアリング法により高スコアと判断された部分文書上位 1500 件と，比較対象となる従来手法によるスコアリング法により高スコアと判断された部分文書上位 1500 件を抽出して精度を求める．今回は予備実験であるため，2008 Topics から 25 個のクエリのみを用いた．INEX によって配布されている評価ツールを用いて提案手法と従来手法の MAiP を求めて比較した結果，図 6 に示すように，残念ながら従来手法の精度が圧倒的に高いという結果となった．各再現率においての精度を確認するためにすべてのクエリに対する再現率–精度グラフ (図 6 参照) を描画したところ，MAiP の比較からも予想されたが，提案手法は全体的に従来手法よりも適合率が小さい部分で特に精度が低くなっていることが分かった．また，提案手法は従来手法に比べて全体的に精度が低かったにも関わらず，再現率が大きくなった際の精度の減少が緩やかで，最終的に精度の逆転が起こった．

(注2)：http://www.inex.otago.ac.nz/

表 1　MAiP 比較表

	提案手法	従来手法
MAiP	0.00708	0.0388

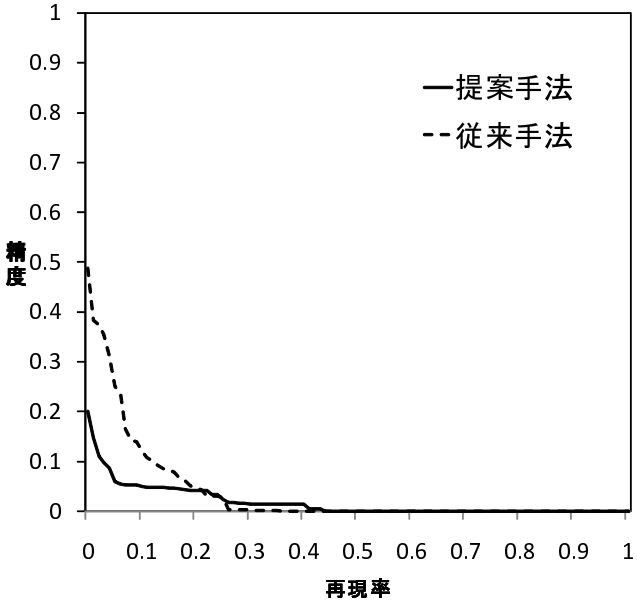

図 6　すべてのクエリに対する再現率-精度グラフ

表 2　スコア別 MAiP 比較表

	キーワード含有率スコア	キーワード寄与率スコア	エッジスコア
MAiP	0.0216	0.000175	0.000418

このような結果となった原因を究明するため，各スコアリング法それぞれに対して評価実験を行った．その結果の MAiP が表 2, 再現率–精度グラフを描画したものが図 7 である．図 7 から分かるように，キーワード含有率スコアはある程度の精度を表したが，エッジスコアとキーワード寄与率スコアは非常に低い精度を表した．これらのスコアリング法では課題を解決できていない可能性が高いため，エッジスコアとキーワード寄与率スコアそれぞれのスコアリング法を見直す必要がある．

エッジスコアの精度がこれほどまで低くなった原因を探るため，エッジスコアで解決を試みた，3. の課題 3 である重要部分の大きさの考慮することに関して適切にスコアリングされているか確かめた．そのために，エッジスコアは部分文書中の MCT を構成するエッジの本数から求められるため，x 軸にある部分文書の MCT に含まれるエッジの本数，y 軸にその部分文書が持つ索引語数の散布図を描いた．その結果，図 8 の通り，MCT のエッジの本数が同じデータ同士において索引語数が大きく異なる分布を示した上に，MCT が大きくなるからといっ

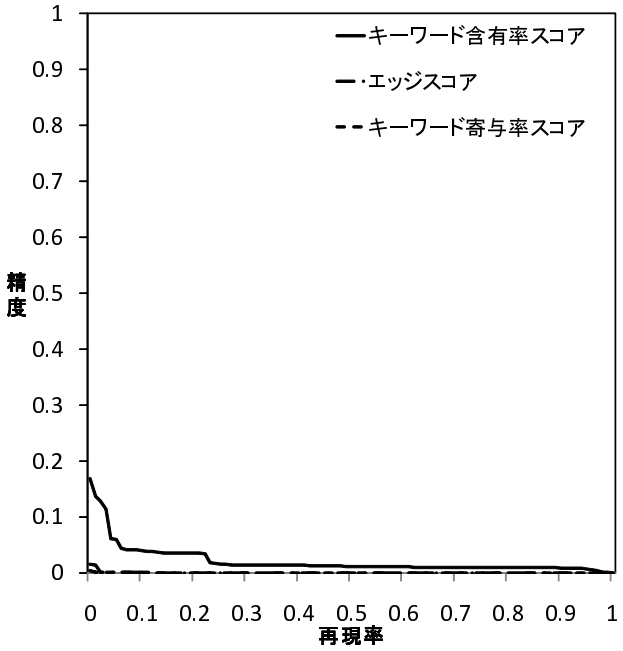

図 7　各スコアの検索精度の比較

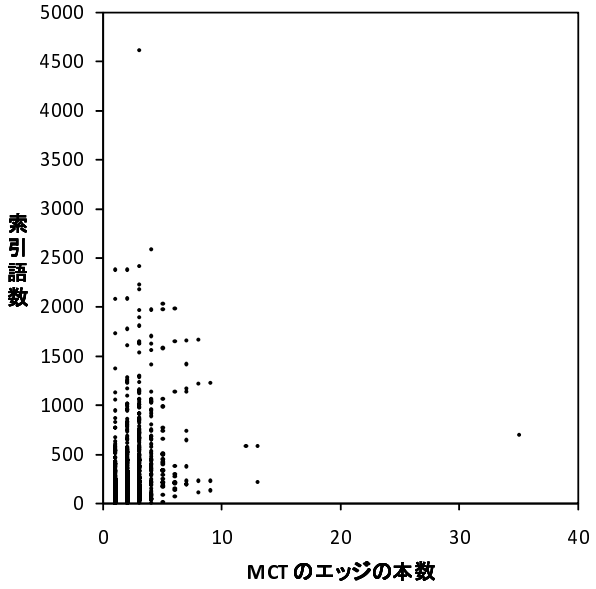

図 8　MCT のエッジの本数と索引語数

て必ずしも索引語数が大きくなるとは限らないことがわかった．このような結果となった原因として，それぞれの部分文書に含まれるノードのテキストサイズに大きく差があるからであると推測されるが，いずれにせよ MCT のエッジの本数では部分文書の大きさを判定できているとはいえない．

また，キーワード寄与率スコアにおいても同様に，スコアリングに利用しているクエリキーワードを含むテキストノード数と部分文書中すべてのテキストノード数の値を確かめることで精度が低くなった原因を調べた．x 軸にある部分文書に含まれるクエリキーワードを含むテキストノード数，y 軸にその部分文書が持つ全テキストノード数を取り，散布図を描いた．その結果，非常に多くの部分文書において，クエリキーワードを含

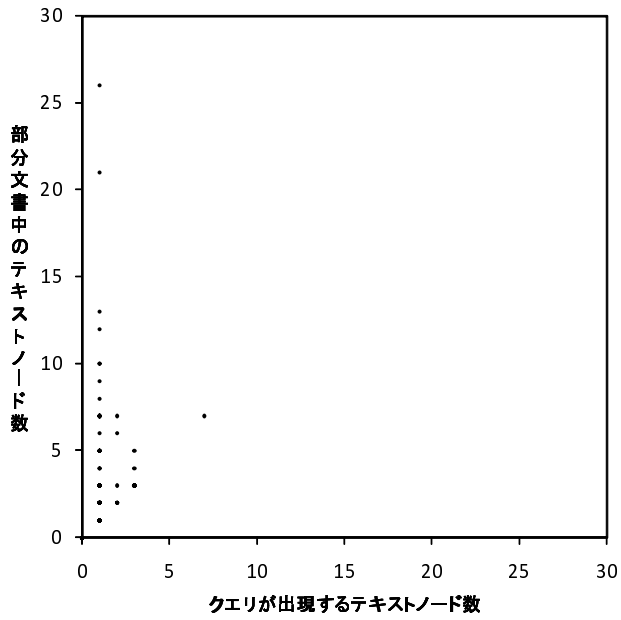

図 9　クエリキーワードと含むテキストノード数と全テキストノード数

むテキストノード数と部分文書中すべてのテキストノード数いずれも 0 を示す部分文書が非常に多く見られた．実際の各値がそのような分布になっているとは考えにくく，そのため，キーワード寄与率スコアのスコア計算に用いているデータを取得する方法に何かしら問題があると思われる．

一方，図 7 から見て取れるように，キーワード含有率は再現率が高くなった際においてもある程度高い精度を示したが，図 6 において再現率が大きくなった際の精度の減少が緩やかになったのはこれが理由であったと考えられる．これは，適合率が高くなるにつれて，スコア統合の際のキーワード含有率スコアの比重を重くすることで検索精度が向上する可能性があることを示唆している．

4.4　考察と今後の課題

今回のスコアリング法の提案では妥当な計算量でスコアリングを行うことを目的としていたために，なるべくシンプルな計算式を目指し，また，なるべく構造化文書の特徴を活かした統計量を用いることを目指した．しかし，4.3 で行った実験の結果，エッジスコアとキーワード寄与率スコアのスコアリング法に改善の必要性が認められた．

例えばエッジスコアのスコアリング法に関して，今回は部分文書中の MCT に含まれるエッジの本数をスコアリングに用いたが，図 8 の結果より，今回の手法では課題 3 を解決することができていないと考えられる．これは，MCT の本数で部分文書の大きさを測った場合に，ファイルサイズの大きなテキストノードへのエッジとファイルサイズの小さなテキストノードへのエッジの重みが等しくなっているため起こった問題であると考えられる．つまり，エッジスコアでは部分文書の構造的な大きさから部分文書の大きさを導くことができると考えたが，実際は構造的情報のみを用いただけでは目的を達成できなかった．そこで，今後エッジスコアで解決させようとした課題を他

のアプローチにより解決を試みる上で，部分文書のテキストサイズを用いたスコアリング法を考案することで，適切に部分文書の大きさを考慮することができるようになり，課題を解決することができると推測される．

次に，キーワード寄与率スコアの改善法に関して述べる．なお，今回行った XML 文書からのデータ取得の方法に問題があったのではないかと推測される．今回寄与率スコアを算出するにあたりテキストノードの数を用いたが，その際，テキストノードは主に葉ノードにおいてのみ出現するという前提でスコアリング法に則したシステム実装を行った．しかし実際は葉ノードではないノードにテキストノードが出現することが起こりえたために，適切にスコアリングできてなかったのではないかと考えられる．そのため，図 9 が示すように，クエリキーワードを含むテキストノード数，部分文書中すべてのテキストノード数ともに正確な値を取得できなかったと考えられる．ただし，先ほどのエッジスコアの問題点として浮き彫りになった，ノードによって索引語数がまちまちであるのであれば，仮に正確に上記の数を計算できたとしても，精度の向上はあまり期待できない．なぜなら，現状では，あるノードがほとんどがクエリに関係のない内容を表しているテキストノードであっても，そのうち一つの索引語でもクエリキーワードが含まれていれば，そのテキストノード全体が有用であると判断されてしまう．それでは，3. の課題 2 である，クエリに無関係な部分の割合を適切にカウントできないと思われるからである．そこで，今後はテキストノードの索引語の頻度などの統計量を効果的に活用し，テキストノード中のクエリに関係する索引語の重みを考慮することで精度の向上を目指す．また，それはすなわち従来の情報検索技術を当てはめることができるのではないかと考えている．

5. おわりに

本稿では，構造化文書に対する高精度検索を目指すためのスコアリング法を提案し，その評価を行った．本提案では，従来の手法では考慮されてこなかった重要部分抽出の際に考慮すべき要件を考慮し，それらを満たすために必要な課題の解決を目的とした三つのスコアリング法をキーワード含有率スコア，エッジスコア，そしてキーワード寄与率スコアとして算出しそれらの正規化および統合を行った．その結果，精度そのものは従来手法に対して低い結果となったが，今後精度を向上させるための取り組みを行う上での有益な知見，すなわち，部分文書に含まれるクエリキーワードの種類数を考慮することによって検索精度が変化し，特に再現率が高くなった場合にある程度の精度を維持することが判明した．

今後の課題として，エッジスコアとキーワード寄与率スコアで解決しようとした課題を満たす代替スコアリング法を考案する必要がある．今回のスコアリング法では，エッジスコアとキーワード寄与率スコアは，検索対象部分文書中のテキストノードにおいて，クエリキーワードが含まれているか否かによって求めた値を用いてスコアリングを行った．しかし，単にクエリキーワードを含むか否かの統計量を用いた場合，4.4 で述べたように実際の課題解決に対して提案手法は相応しくないということが判明した．そのため，今後はノードが持つ索引語の数や頻度などの統計量を用いたスコアリング法を考案することで精度の向上を目指す必要がある．

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 「情報爆発時代に向けた新しい IT 基盤技術の研究」(課題番号: 19024058) によるものである．ここに記して謝意を表す．

文　献

[1] Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, and Francois Yergeau. Extensible Markup Language (XML) 1.0 (Fifth Edition). `http://www.w3.org/TR/REC-xml/`, November 2008.

[2] Ludovic Denoyer and Patrick Gallinari. The Wikipedia XML Corpus. *SIGIR Forum*, 2006.

[3] Norbert Fuhr and Kai Großjohann. XIRQL: A Query Language for Information Retrieval in XML Documents. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 172–180, September 2001.

[4] Torsten Grabs and Hans-Jörg Schek. PowerDB-XML: A Platform for Data-Centric and Document-Centric XML Processing. In *Proceedings of the First International XML Database Symposium*, Vol. 2824 of *Lecture Notes on Computer Science*, pp. 100–117. Springer-Verlag, September 2003.

[5] 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.

[6] 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.

[7] Jaap Kamps, Jovan Pehcevski, Gabriella Kazai, Mounia Lalmas, and Stephen Robertson. Inex 2007 evaluation measures. In *INEX*, pp. 24–33, 2007.

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

[9] Mark Montague and Javed A. Aslam. Relevance Score Normalization for Metasearch. In *Proceedings of the Tenth International Conference on Information and Knowledge Management*, pp. 427–433, October 2001.

[10] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management: an International Journal*, Vol. 24, No. 5, pp. 513–523, 1988.

[11] 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.

[12] 清水敏之, 寺田憲正, 吉川正俊. 関係データベースを用いた XML 情報検索システムの開発. 情報処理学会論文誌: データベース, Vol. 48, No. SIG11, pp. 224–234, June 2007.

[13] 鈴木優. メタ検索エンジンにおけるスコアの統合手法に関する研究. PhD thesis, 奈良先端科学技術大学院大学, Febrary 2004.