特徴ベクトルの要素間の影響を考慮したXML文書検索手法の提案

森　康弘† 吉川　正俊†† 波多野賢治†††

† 名古屋大学 情報科学研究科
†† 名古屋大学 情報連携基盤センター
††† 奈良先端科学技術大学院大学 情報科学研究科
E-mail: †mori@dl.itc.nagoya-u.ac.jp, ††yosikawa@itc.nagoya-u.ac.jp, †††hatano@is.aist-nara.ac.jp

あらまし　本研究では，XML 文書構造の中で文書全体のタイトルや章，節などの見出し，強調語のような文書の主題を表す単語を「特徴語」と定義し．特徴語を含む要素の影響により，ある要素の特徴量ベクトルの改善させる手法を提案する．本手法により，ある要素と特徴語を含む要素との構造的な近さに応じて，特徴語を含む要素の影響の大きさを変えることができるため，XML 文書検索システムの精度の向上を期待できる．
キーワード　XML，文書検索

Proposal of XML Documents Retirieval Method Reflecting Relationship among Element Feature Vectors

Yasuhiro MORI†, Masatoshi YOSHIKAWA††, and Kenji HATANO†††

† Graduate School of Information Science, Nagoya University
†† Information Technology Center, Nagoya University
††† Graduate School of Information Science, Nara Institute of Science and Technology
E-mail: †mori@dl.itc.nagoya-u.ac.jp, ††yosikawa@itc.nagoya-u.ac.jp, †††hatano@is.aist-nara.ac.jp

Abstract　In this research, we first define feature words that are words such as titles of whole documents, heading of chapters and sections, or emphasized words. Then, we propose a method for making use of feature words using context relationship among documents' nodes. Using our method, we can enhance overall performance of XML document retrieval system because feature vectors of retrieved XML subdocuments are improved.
Key words　XML, Document Retrieval

1. はじめに

W3C(World Wide Web Consortium) から仕様が勧告された XML(Extensible Markup Language) [5] は，インターネット上の文書やデータを表現するメタ言語であり，SGML(Standard Generalized Markup Language) の設計思想を受け継ぐ形で設計された．XML 形式の文書の記述は，文書の可読性を上げたり，目的に応じたデータ構造の表現できるため，近年になってさまざまな用途で使用され始めている．たとえば、OASIS [1] や BizTalk [2] のようなポータルサイトには多数の業界別標準 XML アプリケーションが登録されている．多くの企業や団体は、これらの登録済みの標準語彙を利用することによって一つのデータ形式を共有することが可能になった．また，次世代の Web 記述言語としての XHTML (Extensible HyperText Markup Languge) や MS Word の doc ファイルの内部記述言語などのデータフォーマットも XML に基づいている．

以上のような背景から XML で記述された文書が多くなることが予想されるため，HTML(HyperText Markup Language) [8] の普及で Web 検索エンジンが開発されてきたように，XML 文書検索システムに対する性能向上への期待は大きい．

従来の Web 検索システムでは，特定のタグ中の HTML 文書中の主題に関する情報がうまく活用されてきた．ロボット型検索システムでは，Web ページのタイトル表示をする title タグや見出し表示をする hi タグ，strong タグや b タグ等の中にある論理的，視覚的に重要な単語の重みを大きくすることによって，HTML 文書中の主題に関する情報を活用することができる．また，英語の文書では結論から書き始めることが多いため，特定のタグ中で最初に出てきた単語ほど重みを大きくするアプローチも採られている．一方，XML 文書検索システムの場合，主題に関わる情報を利用している例は見当たらない．

本研究では，XML 文書検索システムで上記の主題に関わる情報を利用する際，XML 文書の文書型定義 (Document Type

Definition; DTD) で規定されているようなノードの位置関係を考慮することによって，XML 文書検索システムの精度向上をねらう．文書全体のタイトルや章，節など見出し，論理的，視覚的に重要なイタリック体や太字などの単語を表す強調語を，「特徴語」と定義して本稿を進める．

2. 関 連 研 究

XML 文書を検索するための手法では，現在，XPath (XML Path Language) [6] や XQuery [4] のような XML 問合せ言語を用いた方法が主流である．これらの方法では，問合せを行うための専門的知識や検索したい XML 文書の構造を利用者があらかじめ把握し，検索の際に文書構造を指定する必要があるため，利用者にとって使いやすいものとはいえない．

一方，我々は利用者に対する使い易さを考慮し，問合せキーワードを入力するだけで利用者が求める文書を検索するシステムを開発してきた [9]〜[12]．このシステムは，利用者がキーワードを入力するだけで問合せに適合した XML 文書の一部分，すなわち XML 部分文書を検索することができ，問合せに対する適合度を基にランキング化して利用者に提示する．

しかし，どちらの問合せ方法の XML 文書検索システムも文書の一要素を検索単位としているため，特徴語を含まない要素の場合は主題に関わる主要な情報を含むことができない．

そこで我々は，構造化文書を念頭に置いた単語の重みづけのため，従来の全文検索システムのような特徴語を重要視した重みづけを XML 文書検索システムにも適応することを考える．全文検索システムでは，特徴語の影響が文書全体に及ぶが，XML 文書は構造を持つため，ノードの構造上の位置関係によって特徴語の影響の大きさが異なると予想される．たとえば，ある章に対する章の見出しの影響は，文書全体のタイトルの影響よりも大きいと考えられる．

以上の問題点を解決するため，本稿では特徴語の重みを文書構造について意味的に関係のあるノードに対して反映させるための手法を提案する．

3. XML 部分文書

本研究は，XPath 1.0 [6] で定義されているデータモデルに基づいているため，我々は，XPath データモデルに準拠した用語を使用して議論を進める．

3.1 XPath データモデル

XPath データモデルは，XML 文書を図 1 に示されるような木構造で表現する．それぞれの節点には，document order が割り当てられており，主に element node，attribute node，text node の 3 種類のノードに区別される．

- **element node**

子ノード として，element，attribute，text のいずれかのタイプを複数もつことができる．ラベルは要素型名を示す．

- **attribute node**

子ノード をもたない．ラベルは属性名，属性値を示す．また，複数の属性が存在するとき，属性の順序は区別しない．

- **text node**

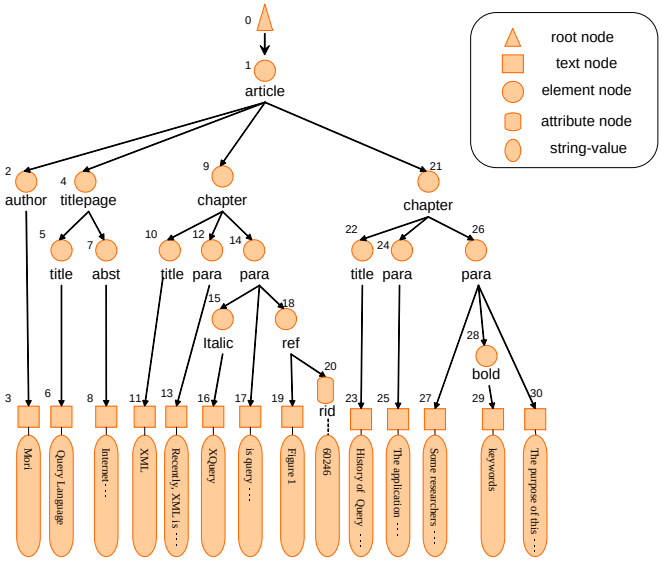

図 1　XML 文書の木構造表現

子ノード をもたない．ラベルは連続した文字データを示す．

3.2 XML 部分文書

XPath データモデルに基づいた XML 文書のための検索モデルとして提案されたものに non-overlapping [7] リストモデルと proximal node [14] モデルがある．本稿で提案する検索モデルは，proximal node モデルに近いので，XML 部分文書を以下のように定義する．

［定義 1］ XML 文書中に出現するすべての要素について，開始タグと終了タグで囲まれた部分，すなわち element node を根とする木全体を **XML 部分文書**と呼ぶ．本稿では，このような XML 部分文書をその根につけられている ID *n* を利用して，**XML 部分文書 #n** と呼ぶ．

4. Retrieved Partial Document(RPD)

4.1 RPD の定義

従来のパッセージ検索 [15] では，情報検索結果を文書を要素単位に分割した単位，もしくは文書全体としていた．また，文献 [9] で提案した XML 文書検索システムでは，XML 文書中の element node を根とする全ての XML 部分文書を検索対象としていた．しかし，このシステムは検索コストが高く，利用者にとって不必要な検索結果が多い．本稿では，これらの XML 部分文書の中から利用者にとって有益な内容を含んでいる XML 部分文書を **Retrieved Partial Document(RPD)** [11] とする．RPD とは利用者が本当に検索したい比較的小さな部分，つまり問合せキーワードを含み，文書構造について意味的にまとまっている部分文書のことである．

4.2 RPD の 例

図 1 の XML 文書に対して，問合せキーワード「XML」をパッセージ検索システムに与えた場合，その検索結果として XML 部分文書，<title >XML </title >が返される．この XML 部分文書の中に利用者が必要としているキーワードが含まれているが，XML のどんな内容なのかが示されていないため，検索結果として不適切である．一方，従来の全文検索システムの

```
<! ELEMENT   article   (author, titlepage, chapter*)>
<! ELEMENT   author  (#PCDTA)>
<! ELEMENT   titlepage   (title, abst)>
<! ELEMENT   title (#PCDTA)>
<! ELEMENT   abst(#PCDTA)>
<! ELEMENT   chapter   (title, (para*))>
<! ELEMENT   para (#PCDTA | italic | bold | ref)>
<! ELEMENT   italic(#PCDTA)>
<! ELEMENT   bold(#PCDTA)>
<! ELEMENT   ref(#PCDTA)>
<! ATTLIST   ref   rid #IMPLIED)>
```

図 2　DTD の 例

ように XML 文書全体を検索結果としても，XML に関する情報が全く書かれていない 2 番目の chapter の情報も含むので，不適切な検索結果と考えられる．

図 1 の XML 文書の中に含まれる XML 部分文書のうち，例に挙げた検索要求にふさわしいと思われる部分文書，つまり RPD は XML 部分文書#9，#12，#14 である．なぜなら，これらの XML 文書には 問合わせキーワード「XML」に関して記述されていて，XML に関して全く書かれていない 2 番目の chapter の情報を含まないからである．

5.　Selected Partial Documents(SPD)

5.1　SPD の特定方法

すべての XML 部分文書を検索対象とすると，検索対象 XML 部分文書数が膨大となり，検索コストが非常にかかる．また，利用者にとって不必要な検索結果が多い．そのため，選択ノードアプローチ [12] によって決定された XML 部分文書を **Selected Partial Document (SPD)** と定義し，SPD のみを検索対象とする．SPD の中から問合せキーワードを含む検索されるべき RPD を検索する．

選択ノードアプローチとは，検索システム設計者が DTD を手がかりに文書構造を把握して文書構造について意味的にまとまっている部分文書の最上位ノードを指定する方法のことである．以下に SPD の特定方法の例を挙げる．

5.2　SPD の 例

ある XML 文書の木構造 (図 1) の DTD を，図 2 に示す．検索システム設計者は，この DTD を利用して文書中に出現する要素型の名前，要素の出現順序，入れ子関係を把握することができる．図 2 の中で，　+，　*によって複数回の出現を許されている要素は chapter や para であり，文書構造について意味的にまとまっている要素ノードとみなすことができる．また，検索システム設計者は要素の名前からも文書構造上の境界を識別でき，さらにタイトルや強調語など特定のタグで囲まれた XML 部分文書は短すぎたり見かけだけの場合が多いので，SPD として適切でないと判断できる．

図 1 の XML 文書インスタンスの木構造の中で意味的にまとまっている要素ノードは chapter と para であるため，SPD は#9，#12，#14，#21，#24，#26 である．

5.3　想定している XML 文書検索システム

図 3 に，我々が開発してきたシステムの概略を示す [9]〜[12].

我々の提案システムは XML 文書を XML プロセッサを用いて DOM 木を構築する部分，構築された DOM 木から element node を探索する部分，探索された element node を根とする XML 部分文書から RPD の候補となる SPD を抽出する部分，SPD から索引ファイルを構築する部分，そして利用者の問合せに対し SPD と問合せキーワードとの類似度を計算し，それを基にランキング付きの SPD を提示する部分から構成されている．

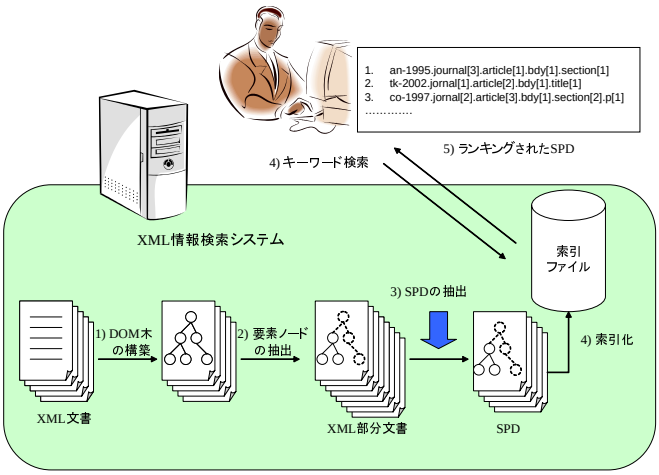

図 3　XML 文書検索システムの概略図 [9]〜[12]

6.　特徴語の影響を考慮した SPD の特徴量の計算手法

本章では，SPD の重みを特徴語の影響を反映させて計算する手法について述べる．

6.1　前 提 条 件

本稿で想定している特徴語とは，XML 文書全体のタイトル，章，節の見出しやイタリック体，太字などで修飾された強調語である．この指定方法は，SPD を抽出した場合と同様に検索システム設計者が DTD をもとに指定する．たとえば図 2 では，title，abst，italic，bold が特徴語である．

6.2　アプローチ

一般的に，XML 文書構造の要素が深い位置にあるほど特定の話題について書かれてある要素になる．図 1 を例に説明すると，XML 文書中で深い位置にある para 要素は隣の para 要素や親の chapter 要素の内容と同じ話題が書かれていることが多いが，chapter 要素は隣の chapter 要素の内容とは別の話題が書かれていることが多い．

したがって，特徴語を含む要素と XML 文書構造の深い位置で近いときに特徴語の SPD に対する影響を大きくする手法を採る．

6.3　文書構造に関する近似度を利用した特徴量の反映手法

特徴語の影響を文書構造的に近いノードに大きく与えるためには，要素間の文書構造における近さを表すパラメータの設定が必要である．検索システム設計者が DTD を見てパラメータを設定することは難しく手間がかかるので，本稿では文書構造の近さを表す式を導入する．

最初に文書構造での深さを考慮したノード間の距離を表す近似度 *Near_Depth* を以下の式で与える．

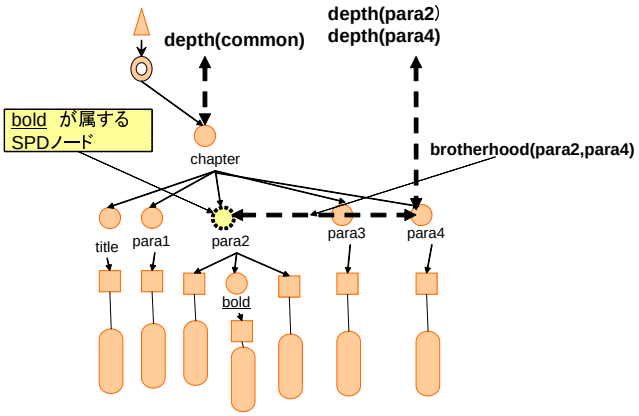

図 4　近似度の関数

$$Near\_Depth(n_1, n_2) = \frac{2 * depth(common(n_1, n_2))}{depth(n_1) + depth(n_2)} \qquad (1)$$

ここで，*depth(n)* は根ノードからノード n までの距離を表す関数である．*common*(n_1,n_2) は，ノード n_1 とノード n_2 の最小共通祖先を表す．上記の式は，ノードが深い位置で近くにあるほど，大きな値をとる関数であり，シソーラス内の語の類似度 [13, pp. 237] を参考にしている．

次に，文書構造での先祖の順序を反映した近似度 *Near_Width* を以下の式で与える．距離を測る対象を n_1,n_2 とする．n_1 と n_2 の *common*(n_1,n_2) の子ノードの中で，n_1 自身または n_1 の先祖であるノードを c_1，n_2 自身または n_2 の先祖であるノードを c_2 とする．そこで，c_1 と c_2 の同じ要素名をもつ兄弟としての順序の差を *brotherhood*(n_1,n_2) とする．ただし，n_1 が *common*(n_1,n_2) と等しい場合もしくは n_2 が *common*(n_1,n_2) と等しい場合は，*brotherhood*(n_1,n_2) は 0 とする．α は，深さを考慮するための定数であり，α が大きくなるほど，XML 文書の深い位置における左右の影響力が大きい．

$$Near\_Width(n_1, n_2) = \frac{(depth(common(n_1, n_2)))^{\alpha}}{1 + brotherhood(n_1, n_2)} \qquad (2)$$

上記の考えを合わせて反映するために，XML 文書の構造間の近似度を以下の式で与える．β は，*Near_Depth* と *Near_Width* のどちらを重視するかを決定する 0 以上 1 以下の定数である．

$$Near(n_1, n_2) = (1-\beta)Near\_Depth(n_1, n_2) + \beta \times Near\_Width(n_1, n_2) \qquad (3)$$

特徴語の影響を考慮する時，この近似度の値を特徴語の特徴量ベクトルにかけた後，対象とした SPD の特徴量ベクトルに加えることによって，文書構造的に近い特徴語の影響をより大きくすることができる．ただし，本稿において，*Near_Depth* と *Near_Width* の両方を一括して計算するため，特徴語はある SPD の属性として扱う．そのため，図 4 で示すように特徴語が所属するノードの祖先の中で特徴語に一番近い SPD ノードに，特徴語があるものとして扱うことにする．以下に，XML 文書の構造間の近似度の例を挙げる．

［例 1］ 図 1 の中の文書全体のタイトルを表す要素 ID 5 のノードと，章の見出しを表す要素 ID の 10 のノードのそれぞれに対し，要素 ID12 の SPD のノードとの *Near* を計算する．*Near_Depth* と *Near_Width* の両方を一括して計算するため，要素 ID 5 のノードと要素 ID 10 のノードは，それぞれ，要素 ID 4 のノードと要素 ID 9 のノードとして扱うことにする．次に，式 (1),(2) に代入して計算すると，*Near_Depth*，それぞれ $\frac{2\times1}{2+3} = \frac{2}{5}$ $\frac{2\times2}{2+3} = \frac{4}{5}$ となる．また両方とも *brotherhood* が 0 なので，α を 1 とすると *Near_Width* はそれぞれ 1 と 2 である．

β を $\frac{2}{3}$ とすると *Near* は，それぞれ $\frac{3}{5}$，$\frac{6}{5}$ となる．

要素 ID 12 の SPD ノードへの影響は，文章のタイトルを表す要素 ID 5 のノードより，章の見出しを表す要素 ID 10 のノードの方が大きいことを，計算値は示している．

6.4　重み計算式

以上をまとめて，SPD の特徴量ベクトルを以下のように表現する．

$$e'_{ij} = (1-\gamma)e_{ij} + \gamma \times \sum_{s=1}^{m} Near(e_i, t_s) * t_{sj} \qquad (4)$$

ただし，特徴語の影響を考慮しない場合の SPD の特徴量ベクトルは $\boldsymbol{e_i} = (e_{i1}, e_{i2} \ldots e_{iN})$，特徴語の影響を考慮した後の SPD の特徴量ベクトルは $\boldsymbol{e'_i} = (e'_{i1}, e'_{i2} \ldots e'_{iN})$ である．ある SPD とある近似度の範囲内にある特徴語を含む要素群を $(t_1, t_2, \ldots t_m)$，その一つの特徴語の特徴量ベクトルを $\boldsymbol{t_s} = (t_{s1}, t_{s2} \ldots t_{sN})$ と表現する．$\boldsymbol{t_s}$ は，例えば特徴語の IDF(inverse document frequency) をもとに決定することが考えられる．γ は，特徴語の重みをどれだけ考慮するかを決定する定数であり，0 以上 1 以下の値である．γ が大きいほど特徴語の情報が影響力をもつことになる．

6.5　特徴語の影響を考慮した SPD を返す手順

（1） それぞれの SPD に対して，*Near* がある閾値以内の特徴語を選択する

（2） 上記の式を利用して，特徴語の影響を与えた SPD の特徴量ベクトルによって，SPD を索引化する

（3） 問合わせの特徴ベクトルと SPD の特徴ベクトルの類似度を基に，ランキング化された SPD を表示する．

7.　お わ り に

本稿では，XML 文書検索システムに構造化文書を念頭に置いた単語の重みづけを行うため，XML 文書の構造に関する意味的な関係を考慮して特徴語を利用することによって，XML 文書検索の精度を向上させる手法を提案した．この手法によって SPD の特徴ベクトルを文書構造的に近い特徴語を使用して改善するため，XML 文書検索システムの精度向上が期待できる．これを実証するため，我々は INEX Project [3] において作成された INEX テストコレクションを使用して性能評価する予定である．

今後の課題として以下の点が残されている．

- 章や節の見出しには，“ Introduction ”, “ Related Work ”, “ Conclusion ”などの単語が文書集合全体を通して頻繁に存在することが分かる．これらの単語は，文書構成上の役割を表現

しているにすぎない．よって，文書集合全体を通して頻繁に存在する見出しによる影響は小さくする．

- 強調語では，数式や物理量などで使われることが多い．この場合，強調語が主題を表しているとは言い難いので，影響は小さくする．

- 章や節の見出しには，SPD の内容を抽象的に表現する単語の存在が多い．たとえば見出しが“ 要求 ”場合，その属する SPD の中で“ 要求 ”という単語が使用されることは少なく，別の話題たとえば“ 要求 ”されている内容が書かれてあることが一般的である．逆に，見出しの単語がその属する SPD でそのまま使われる場合，見出しの内容に関する具体的な記述が書かれていることが多い．よって，見出しの単語がその属する SPD でそのまま使われる場合の方が，見出しの単語が SPD で使われない場合よりも，重みの影響を大きくする手法を採用する必要がある．

文　　献

[1] http://www.oasys-open.org.

[2] http://www.biztalk.org.

[3] http://qmir.dcs.qmw.ac.uk/INEX/.

[4] S. Boag, D. Chamberlin, M.F. Fernandez, D. Florescu, J. Robie, J. Siméon. *XQuery: A Query Language for XML*. W3C Working Draft, November 2003. http://www.w3.org/TR/xquery.

[5] T. Bray, J. Paoli, C.M. Sperberg-McQueen, and E. Maler. *Extensible Markup Language (XML) 1.0 (Second Edition)*. W3C Recommendation, 2000. http://www.w3.org/TR/REC-xml.

[6] J. Clark and S. DeRose. *XML Path Language (XPath) Version 1.0*, Vol. 16. W3C Recommendation, November 1999. http://www.w3.org/TR/xpath.

[7] C. Clarke, G. Cormack, and F. Burkowski. An Algebra for Structured Text Search and A Framework for its Implementation. *The Computer Journal*, Vol. 38, No. 1, pp. 43–56, 1995.

[8] D.Raggett. *HTML 3.2 References Specification.* W3C Recommendation 14-Jan-1997, January 1997. http://www.w3.org/TR/REC-html32/.

[9] 波多野賢治, 渡邉正裕, 吉川正俊, 植村俊亮. 情報検索技術を用いた XML 部分文書の検索手法. 情報処理学会論文誌:データベース, Vol. 42, No. SIG8(TOD10), pp. 36–46, 2001.

[10] 波多野賢治, 絹谷弘子, 吉川正俊, 植村俊亮. XML 文書検索のための検索結果粒度決定. *DEWS2003*, 2003.

[11] 波多野賢治, 絹谷弘子, 吉川正俊, 植村俊亮. 統計量を用いた XML 部分文書検索システムの実装. *DEWS2004*, 2004.

[12] 絹谷弘子, 波多野賢治, 吉川正俊, 植村俊亮. XML 文書の文書構造と内容を用いた部分文書の抽出方法. 情報処理学会論文誌:データベース, Vol. 43, p. 80, March 2002.

[13] 長尾真. 岩波講座ソフトウェア科学 15 自然言語処理. 岩波書店, 1996.

[14] G. Navarro and R. Baeza-Yates. A Model to Query Document Databases by Content and Structure. *ACM Transactions on Information Systems*, Vol. 15, No. 4, pp. 400–435, October 1997.

[15] G. Salton, J. Allan, and C. Buckley. Approaches to Passage Retrieval in Full Text Information Systems. *Proc. of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 49–58, 1993.