DEWS2003 8-A-05

XML 文書検索のための検索結果粒度決定

波多野賢治† 絹谷 弘子†† 吉川 正俊††† 植村 俊亮†

† 奈良先端科学技術大学院大学 情報科学研究科
†† 科学技術振興事業団 戦略的基礎研究推進事業
††† 名古屋大学 情報連携基盤センター
E-mail: †{hatano,uemura}@is.aist-nara.ac.jp, ††kinutani@dblab.is.ocha.ac.jp, †††yosikawa@itc.nagoya-u.ac.jp

あらまし XML 文書の検索方式は，キーワードと要素名などの文書構造の組を問合せとして入力する SQL のような形式をとっている．しかし，利用者の利便性を考えた場合，そのような問合せの方法は利用者にとって使いやすいものとは言えない．したがって，XML 文書の検索も Web 検索エンジンのように問合せとしてキーワードを入力するだけで，利用者が欲している XML の一部分を検索できるようにすべきである．本稿では，キーワードを利用した XML 文書検索システムの実現のために，あらかじめ XML 文書を分割し検索結果とする際の検索結果の粒度決定法について提案する．本稿における提案によって，検索結果候補の数を削減することが可能であるため，XML 文書検索の高速化およびその高精度化を図ることが可能となる．

キーワード XML 文書検索，検索結果の粒度決定

A Unit of Retrieval Results for XML Documents

Kenji HATANO†, Hiroko KINUTANI††, Masatoshi YOSHIKAWA†††, and Shunsuke UEMURA†

† Graduate School of Information Science, Nara Institute of Science and Technology
†† CREST Program, Japan Science and Technology Corporation
††† Information Technology Center, Nagoya University
E-mail: †{hatano,uemura}@is.aist-nara.ac.jp, ††kinutani@dblab.is.ocha.ac.jp, †††yosikawa@itc.nagoya-u.ac.jp

Abstract XML query languages adopt a SQL-style query to retrieve portions of XML documents. However, the query style of XML query languages is not suitable for XML document retrieval because it is difficult for users to specify both keywords and document structures of XML documents in order to retrieve portions of XML documents. In this paper, we propose a method for defining a unit of retrieval results to develop a keyword-based XML document retrieval system. Using our method, we can reduce the number of targeted retrieval results for XML documents, so that we can speed up retrieving retrieval results and enhance overall performance of XML document retrieval system.

Key words XML Document Retrieval, A Unit of Retrieval Results

1. はじめに

XML (Extensible Markup Language) [3] が，情報化社会に与えた影響は非常に大きく，世間では WWW (World Wide Web) に次ぐ大きな提案であったとまで言われている．特にデータ交換については，これまでアプリケーション間で決められた専用のファイルフォーマットを利用するしか術がなかったが，XML の標準化によって電子商取引，流通チェーンの統合，データ管理，出版などが，容易に行えるようになってきている．このような背景から，計算機上に存在するあらゆるデータが，近い将来，XML 形式で記述されると予想される．そうなると WWW の発展に伴って WWW 検索システム (Web 検索エンジン) が開発されたように，XML の標準化が進むにつれ XML 文書検索システムへの期待は大きくなると考えられる．

XML 文書を検索するための手法としてこれまで広く知られている方法は，XML 問合せ言語 [2] であり，これらは市販の XML 対応を謳ったデータベースの検索機能に盛り込まれたり，W3C (World Wide Web Consortium) からワーキングドラフトが公開されたり [1] と，盛んに研究が行われている．しかし，これら XML 問合せ言語は，データベース問合せ言語の SQL と同様，問合せを行うための専門的知識や，検索したい XML 文書の文書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，利用者の利便性を考えると Web 検索エンジンのように使いやすいものとはいえないのが現状で

ある．

このような利用者に対する利便性に関する問題点を克服するために，我々はこれまで，問合せキーワードを入力するだけで利用者が求めている XML 文書を検索できるシステムを開発してきた [7]．開発したシステムでは，Web 検索エンジンのように，利用者は検索キーワードを入力するだけで求めている情報，すなわち，問合せに相応しい XML 文書中の一部分を検索することができ，さらにそれらは問合せ内容に対する相応しさを基にランキングされている．しかし，キーワード入力による問合せの実現のために，あらかじめ検索対象となる XML 文書をその文書構造を利用して XML 部分文書に分割しており，XML 文書を分割した結果，検索対象 XML 部分文書数が膨大となり，検索に非常に時間がかかるという問題点を持っていた．

そこで本稿では，XML 検索システムにおける高速検索の実現のために，検索対象となる XML 部分文書の粒度を決定し，その数を削減する手法を提案する．我々は，検索対象となる XML 部分文書には 2 種類，すなわち利用者にとって有益な内容を含んでいる部分文書 (以下 CPD (Coherent Partial Document) と表記する) と不要な部分文書(注1)があると考えており，不要な部分文書を取り除くことが，検索対象 XML 部分文書の数を削減，検索の高速化および検索精度の向上につながると考えている．つまり，質のよい CPD を検索対象の部分文書と定義することが XML 文書検索システムの利便性と高速化を実現できる要素であると考えている．

2. Coherent Partial Document

2.1 XML 部分文書

本研究は，XPath 1.0 [5] で定義されているデータモデルに基づいているため，本稿で用いられている用語は XPath データモデルに合わせている．

XPath データモデルでは，XML 文書は図 1 に示されているように，階層構造をもった木構造で表現され，それぞれの節点は document order を利用して ID が振られている．XML 木の leaf node は図のように text node もしくは attribute node であり，XML 木の root node の子は document node と呼ばれている．また，document node と leaf node 間にある中間ノードは element node と呼ばれている．

この XPath データモデルに基づいた XML 文書のための検索モデルは，これまでに 2 種類，non-overlapping リストモデル [6] と proximal node モデル [12] が提案されている．本稿で提案する検索モデルは後者の検索モデルに近いため，その検索モデルを利用して，XML 部分文書の定義を以下のように定めている．［定義 1］（XML 部分文書） 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，document node または 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．

(注1)：このような部分文書のことを，文献 [8] では stop-contexts と呼んでいる．この文献においても，検索システムの scalability の確保には stop-contexts の除去が必要であると述べられている．

2.2 提案している XML 文書検索システム

1 章でも述べたように，我々は利用者に対する利便性を考慮し，問合せキーワードを入力するだけで利用者が XML 部分文書を検索できるシステムを開発してきた [7]．図 2 に構築したシステムの概略を示す．図に示したように，我々の提案システムは XML 文書を XML パーザー Xerces(注2) 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 部分文書からその inverted file を構築する部分，そして利用者の問合せに対し各 XML 部分文書と問合せとの類似度を計算しそれを基にランキング付きの検索結果を提示する部分の 4 つから構成されている．

我々が文献 [7] で提案していた XML 文書検索システムでは，XML 文書木中の 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が，本稿で提案する手法は，これら XML 部分文書から有益な内容を含んでいる XML 部分文書だけを検索対象にするように改善する．1 章で述べたように，このような有益な内容を含んでいる XML 部分文書を，本稿では CPD と定義している．

2.3 CPD の概念

利用者にとって意味のある XML 部分文書，すなわち CPD とは，文書構造および文書内容について意味的にまとまりのある部分文書であり，従来の情報検索技術，例えばパッセージ検索 [13] で行われているような検索要求として入力されたキーワードを含んだ XML 文書の一部分とは異なる．つまり，入力キーワードを含み，かつ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を指す．精度のよい，しかも利便性の高い検索システムを構築するためには，このような XML 部分文書を検索対象とすべきである．

このような意味のある XML 部分文書を具体例を挙げて説明する．例えば，入力キーワードとして `Hatano` を従来型のパッセージ検索システムに与えた場合，図 1 の XML 文書からその検索結果として，XML 部分文書

```
<author>Hatano</author>
```

が返される．この XML 部分文書は，利用者が必要としているキーワードを含んでいるが，`Hatano` が何の `author` であるか示されていないため，利用者にとっては情報量が不足しており，検索結果としては不適切である．また，従来型の全文検索システムのように，図 1 が示す XML 文書全体が先の問合せの検索結果として返されても，利用者にとって問合せの解として不必要な 1 番目の `chapter` の情報まで含まれているため，これも情報過多な検索結果であるため不適切だと考えられる．

図 1 が示す XML 文書の中に含まれる XML 部分文書のうち，先に例として挙げた検索要求に最も相応しいと思われる部分文書，すなわち意味のある XML 部分文書は，要素 ID #20 を root node とする XML 部分文書 #20 である．なぜなら，この XML 文書には 2 つの `chapter` が存在し，`Hatano` は 2 番目の `chapter` の `author` だからである．利用者が情報検索を行う場

(注2)：`http://xml.apache.org/xerces-j/index.html`

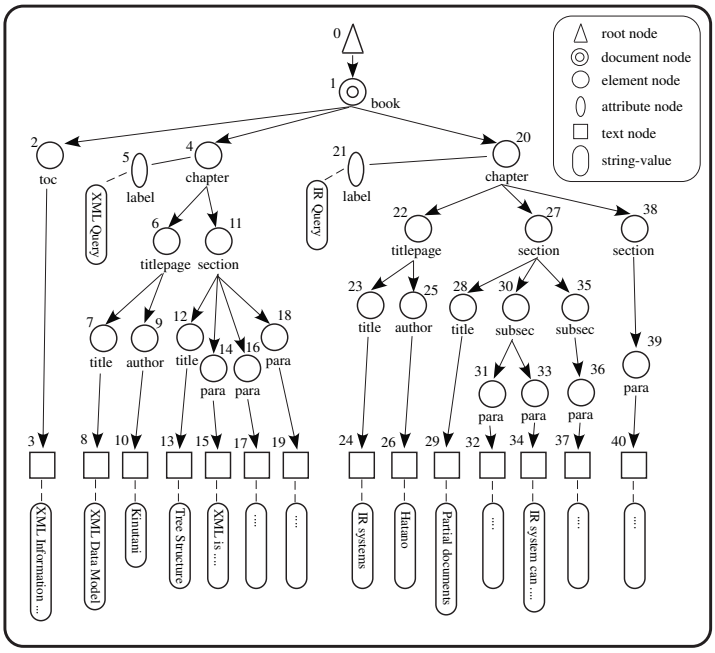

図 1　XML 文書の木構造表現

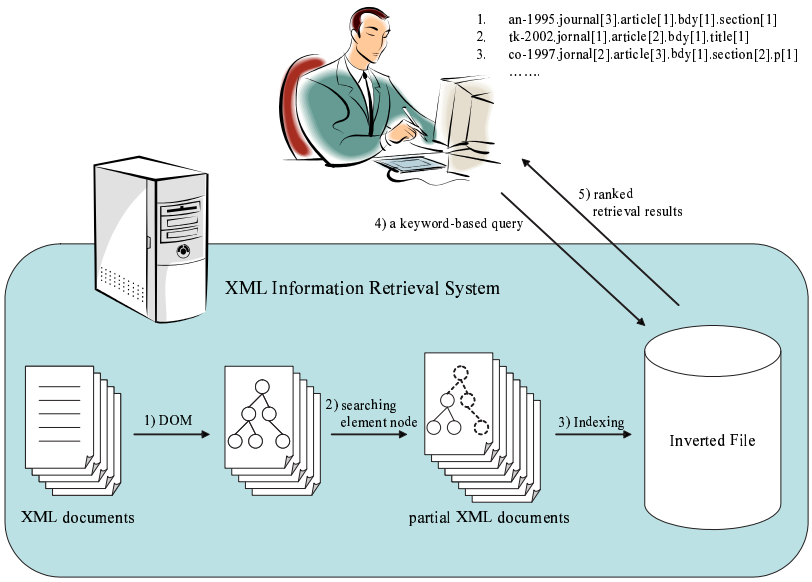

図 2　XML 文書検索システムの概略図

合は，入力キーワードを含んでいる最小の部分文書ではなく，XML 部分文書 #20 のような意味のある XML 部分文書群を検索対象とすべきであり，そのことが検索精度を向上させ，また検索システムの利便性の向上にも結びつく．我々はこの意味のある XML 部分文書のことを CPD と呼んでいるが，XML 文書中から分割されるすべての XML 部分文書が CPD に該当するわけではない．そのため，利用者が文書構造を意識せずに従来の文書検索と同様に，検索要求としていくつかのキーワードを与えるだけでこれら CPD を検索結果の候補として得るためには，検索システムに CPD を決定する仕組みが必要となる．

2.4　CPD の定義

文書検索の研究分野において，検索要求に対してそれに類似した文書の一部分だけを検索するというテーマは，先に述べたパッセージ検索 [13] が提案されてから非常に注目されている．パッセージ検索では，文書の章や節，文節を自然言語処理技術を利用して抽出することでパッセージ長の決定を行っている(注3)が，利用者に検索結果を提示する際にそのパッセージを含む文書全体を提示することが多く，利用者はその文書のどの部分が検索要求に対して類似しているのかを把握することが困難であった．また，検索結果としてパッセージ自体を提示する検索システムであっても，パッセージ長が固定であることが，2.3 節で述べたようにパッセージ長が利用者の問合せに不適切な場合があり，検索システムの利便性が損なわれる点では変わりはなかった．

(注3)：文書の章や節，文節などのうち，どれをパッセージとして利用するかは，検索システムの設計者が決めることが多い．

そうした問題点に対処するため，我々は先行研究においてXML 文書中から意味のあるまとまり (以下文脈と表記する) をXML 文書の構造を利用して発見する「文脈検索」という手法を提案した (図 2 中の 2) の部分に該当) [10]．この手法の基本的な考え方は，文脈は文書の論理構造によって決定されるものであり，文脈を表現する XML 部分文書の root node は，元の XML 文書中に同じ要素名を持つ兄弟ノードを持つことが多いという事実を利用している．この手法を用いることで，先のパッセージ検索の問題点であったパッセージ長が固定されていることによって生じる問題が解消され，また，検索結果として文脈を表現する XML 部分文書を返すような検索システムを実装したので，利用者にとって利便性の高いシステムとなった [9].

ここで問題となるのは，文脈検索によって決定された文脈 (以後，文脈と区別するために CS (Coherent Subdocument) と表記する) が文書の論理構造である章や節などを反映しているかどうかである．実際，XML 文書内には，文書の論理構造だけではなく語の強調やリンクのアンカーなどに使用される要素も多数存在するため，そのような要素を根とする XML 部分文書が CS を構成する可能性が高くなる．さらに，例えば図 1 中の XML 部分文書 #11 は XML 文書内で章を表した部分であるが，文脈検索ではこの XML 部分文書を文脈として抽出することができない．したがって，文献 [10] で提案した文脈検索を XML 文書検索に利用する方法は，(単語数として 2, 3 語程度の) それ自体では意味を持たない XML 部分文書が CS として抽出されたり，本来，文脈として抽出されるべき XML 部分文書が CS として抽出されないなどの問題点があり，提案した文脈検索をあらゆる XML 文書に適用することはできないことが想像される．実際，多くの XML 文書について，文書の論理構造を文脈として抽出できるかどうかの予備実験を行ったところ，XML 文書から論理構造を表現した文脈だけを忠実に抽出することができず，それが原因で検索精度の低下が生じることが判明した．

本稿では，文脈検索のように検索対象となる XML 文書の文書構造の性質を利用するのではなく，XML 文書中の持つ文書構造にしたがって分割した XML 部分文書群から，部分文書自身に含まれる単語数や異なり語数などの統計量を利用して明らかに文脈とはなりえない XML 部分文書を除去し，残った XML 部分文書が文書の論理構造を反映した部分文書であるとする新しい手法を提案する．

こうして得られた XML 部分文書を，2.3 節で述べた CPD と定義することで，CPD と文脈が一致しないという問題点は解消される．

3. CPD 決定のための評価実験

実際にどのような統計量をどのように CPD の決定に利用すればよいのかは，評価実験を行った上で決定する必要がある．

実験に使用した XML 文書は，IEEE Computer Society から 1995 ～ 2002 年に発行された雑誌および論文誌に含まれているすべての記事および論文であり，含まれている論文数は 12,107 文書である．この XML 文書群は，INEX test collection と呼ばれており，2002 年 4 月に発足した INEX Project(注4) によって DTD が制定され，すべての記事，論文がその DTD に基づいて 1 つの XML 文書として表現されている．DTD 中で定義されている文書要素は 192 種類であり，その XML 文書サイズは 496 GBytes にのぼる．

3.1 統計量の決定

評価実験では XML 文書から抽出することが可能な統計量として，XML 文書が持つ文書構造にしたがって分割した XML 部分文書に含まれる単語数，異なり語数，そして単語数と異なり語数から計算される異なり語率を利用した．この 3 種類とした理由には，XML 部分文書は単語で構成されており，また，その内容は文章である場合や，数値や単語，熟語などのデータである場合など多彩であるため，XML 部分文書に含まれる文数など単語に関係ない統計量を利用することが難しいからである．以下に，異なり語率の定義を示す．

［定義 2］（異なり語率） XML 部分文書中に出現する単語数を n^w，異なり語数を n^k とすると，異なり語率 R は以下のように表現される．

$$R = \frac{n^k}{n^w} \tag{1}$$

異なり語率を定義する理由は，XML 部分文書に含まれている単語数はさまざまであるため，検索を行う際に XML 部分文書と問合せとの類似度をベクトル空間モデルで評価するのに適しているかどうかを判定するためである．一般にベクトル空間モデルで評価可能な文書には，同じ単語が何度も含まれており，異なり語率は 100% とはならない．その一方，カタログのデータ一つ一つを表す文書には，同じ単語が複数出現することはほとんど考えられず，異なり語率は 100% に近いといえる．すなわち，異なり語率が 100% に近い XML 部分文書はデータ指向が強いと考えられ，ベクトル空間モデルで CPD として検索されるべきではないと考えられる．

本実験では，以下の手順で統計量から意味のある XML 部分文書，CPD を決定する．

（1） 図 2 に示したように，INEX test collection を表現する XML 文書を Apache Xerces を利用して DOM 木に展開し，さらに，その element node を探索しておく．探索された element node には document order にしたがって ID が付けられる．

（2） 抽出した element node を根とする XML 部分文書を XML 文書から切り出す．2.1 節で述べた XML 部分文書の定義から，XML 文書から XML 文書中の element node の数と同数の XML 部分文書が抽出されることになる．

（3） 各 XML 部分文書に含まれる単語数 n^w，異なり語数 n^k，そしてそれらの比を表す異なり語率 R を利用して，CPD として相応しい XML 部分文書を決定する．具体的には n^w，n^k，そして R においてそれぞれある閾値を設定し，その閾値を利用して XML 部分文書が CPD として相応しいかどうかを決定する．また，そうして決定した CPD の文書数 N が，検索対象 XML 部分文書数となるので，N を利用することで XML 検索

(注4)：Initiative for the Evaluation of XML Retrieval (INEX): http://qmir.dcs.qmul.ac.uk/INEX/.

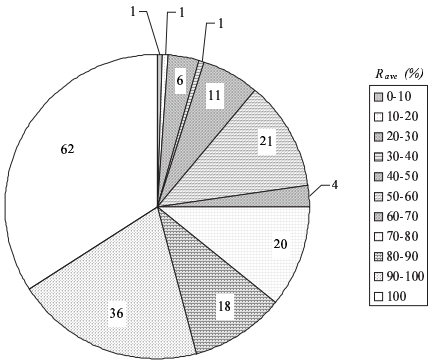

図 3　平均異なり率 R_{ave} による XML 部分文書の分類 (前処理有)

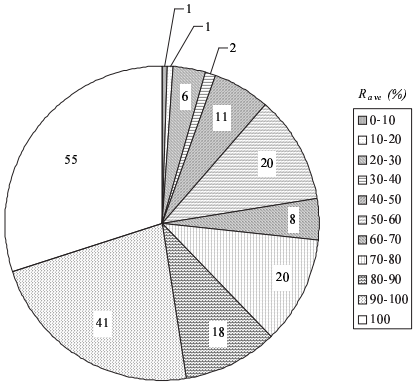

図 4　平均異なり率 R_{ave} による XML 部分文書の分類 (前処理無)

システムの高速化の指標とすることが可能となる．

3.2　統計量の解析および考察

解析のために使用する統計量の一部を表 1 〜4 に示す．統計量はストップワード処理や接辞処理などの前処理の影響もあると考え，前処理を行った場合と行わなかった場合双方について集計を行った．また，表中の平均異なり語率 R_{ave} は，XML 部分文書の root node 名が表中の要素名であるような XML 部分文書 d_i が持つ単語数を n_i^w，異なり語数を n_i^k としたとき，

$$R_{ave} = \frac{\sum_i n_i^k}{\sum_i n_i^w} \qquad (2)$$

で計算される値を表している．

表 1，2 からわかるように，INEX test collection 全体を表現する XML 文書の root node に近い要素 (例えば，`books`，`journal`，`article` など) を根とする XML 部分文書には，多くの単語，異なり語が含まれている．また，それらの XML 部分文書の多くは，その平均異なり語率 R_{ave} が小さく，これらの傾向はストップワード処理や接辞処理などの前処理の有無に関わらずほぼ同じである．さらに，平均異なり語数 n_{ave}^k が 100 語以上である XML 部分文書の種類は，192 種類ある XML 部分文書のうち 20 種類以内であり，サイズの大きな XML 部分文書の種類は少ないこともわかる．一方，抽出された部分文書数 N が多い XML 部分文書の種類に注目してみると (表 3，4 参照)，平均単語数 n_{ave}^w が少ない XML 部分文書は，その平均異なり語率 R_{ave} が 100% に近い値となっており，この傾向もまた前処理の有無に関わらずほぼ同じである．また，抽出された XML 部分文書の種類を比較しても前処理が行われた場合が 181 種類に対し，前処理が行われなかった場合は 183 種類とほとんど差がなかった(注5)．以上の点を考慮すると，それぞれの XML 部分文書によって，単語数および異なり語数に大きなばらつきが見られるため，これら 3 種類の統計量を単独で利用して CPD を決定する手法は非常に難しく，それぞれの統計量の相関関係を利用する必要があることが判明した．

そこで，解析に利用している統計量をさらに詳細に分析するために，XML 部分文書をその平均異なり語率 R_{ave} の比率によって 11 のグループに分類した結果を，図 3，4 に示す．これら円グラフが示すように，全体の約 3 割 (前処理ありの場合は 62 種類，前処理なしの場合は 55 種類) の XML 部分文書の平均異なり語率 R_{ave} が 100%，また全体の約 2 割 (前処理ありの場合は 36 種類，前処理なしの場合は 41 種類) の XML 部分文書の平均異なり語率が 90% 以上 100% 未満であった．これらの XML 部分文書の多くは，INEX test collection の XML 文書木において，leaf node にあたる XML 部分文書であり，また，表 3，4 を参照してもわかるように，それらに含まれている単語数は非常に少ない．さらに，これら 2 つの円グラフを比較してみても，平均異なり語率 R_{ave} の分類傾向は，前処理の有無とは関係がないと考えても差し支えない程度の差違しか表れていない．これらの結果は，3.1 節を踏まえると，少なくとも平均異なり語率が 100% の XML 部分文書はデータ指向が強く，ベクトル空間モデルによって CPD として検索されるべきではない．すなわち，XML 部分文書の持つ異なり語率 R_{ave} を変化させることで検索対象 XML 部分文書数の調整が可能であるため，キーワードを利用した XML 文書検索システムの課題であった高速検索の実現を，異なり語率 R_{ave} を利用して実現することが可能であることが分かった．

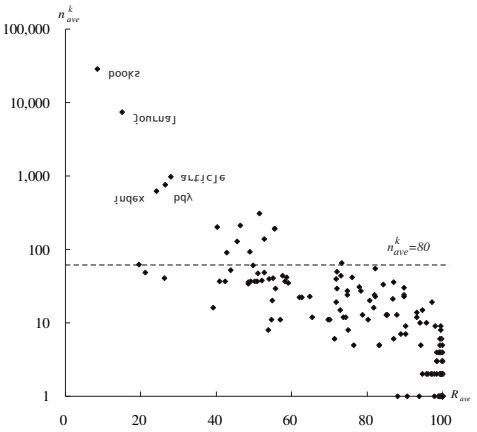

図 5　平均異なり率 R_{ave} と平均異なり語数 n_{ave}^k の関係 (前処理有)

一方，図 5，6 は，各 XML 部分文書の種類に対する異なり語

(注5)：DTD 中では，192 種類の XML 部分文書が抽出されるはずだが，本実験では単語数が 0 である XML 部分文書は部分文書として扱っていないため，抽出される XML 部分文書の種類に差が生じたものと思われる

表 1　統計量の解析結果 (平均単語数 n^w_{ave} 上位 20，前処理有)

要素名	部分文書数	単語数 n^w			異なり語数 n^k			平均異なり語率
	N	平均 (n^w_{ave})	最大 (n^w_{max})	最小 (n^w_{min})	平均 (n^k_{ave})	最大 (n^k_{max})	最小 (n^k_{min})	R_{ave} (%)
books	125	337,099	894,853	42,734	28,897	64,181	6,341	8.57
journal	860	48,997	129,417	17,192	7,342	14,903	3,982	14.99
article	12,107	3,478	28,824	32	974	4,727	29	28.02
bdy	12,107	2,884	28,276	13	765	3,943	11	26.55
index	117	2,585	10,728	381	623	1,593	230	24.13
bm	10,060	604	10,074	2	310	2,863	2	51.40
sec	69,733	501	16,089	1	201	2,613	1	40.24
dialog	194	458	2,424	21	212	906	19	46.45
bib	8,543	350	5,690	8	194	1,959	8	55.48
bibl	8,551	350	5,690	8	194	1,959	8	55.48
tgroup	5,822	318	3,961	2	62	401	2	19.58
ss1	61,490	280	11,857	1	127	2,109	1	45.61
app	5,863	262	7,698	2	138	1,353	2	52.72
tbody	5,820	233	3,851	2	49	390	2	21.23
ss3	127	213	1,361	9	91	325	9	42.88
ss2	16,288	189	11,640	1	92	1,261	1	48.90
tbl	12,740	159	3,965	6	41	414	6	26.17
proof	3,765	122	3,815	5	60	801	5	49.71
dl	353	120	1,562	11	52	745	5	43.90
l4	117	92	794	6	37	231	6	40.83
181 種	6,802,061	2,222	894,853	1	234	64,181	1	38.85

表 2　統計量の解析結果 (平均単語数 n^w_{ave} 上位 20，前処理無)

要素名	部分文書数	単語数 n^w			異なり語数 n^k			平均異なり語率
	N	平均 (n^w_{ave})	最大 (n^w_{max})	最小 (n^w_{min})	平均 (n^k_{ave})	最大 (n^k_{max})	最小 (n^k_{min})	R_{ave} (%)
books	125	568,089	1,530,446	73,483	48,007	93,216	10,930	8.45
journal	860	82,571	223,661	28,896	12,665	24,655	7,179	15.34
article	12,107	5,862	38,269	43	1,604	8,128	39	27.36
bdy	12,107	5,024	33,850	15	1,321	7,197	15	26.30
index	117	3,133	13,354	467	817	1,998	300	26.30
dialog	194	969	5,661	33	413	1,698	32	42.66
sec	69,733	876	17,619	1	344	4,700	1	39.34
bm	10,060	853	18,347	2	412	5,175	2	48.29
ss1	61,490	486	14,142	2	219	3,798	2	45.03
bib	8,543	434	7,016	9	233	2,215	9	53.80
bibl	8,543	434	7,016	9	233	2,215	9	53.80
app	5,863	418	11,810	2	213	1,712	2	51.03
ss3	127	380	2,471	22	164	560	20	43.33
ss2	16,288	332	11,959	2	162	1,579	2	48.74
tgroup	5,822	327	3,962	2	66	409	2	20.30
tbody	5,820	242	3,904	2	53	399	2	22.01
proof	3,765	208	6,555	3	100	1,186	3	48.46
dl	353	182	2,777	18	80	1,226	12	44.36
l4	117	182	1,769	9	71	431	9	39.36
lb	54	169	1,172	31	67	157	16	39.52
183 種	8,224,053	3,693	1,530,446	1	384	93,216	1	37.03

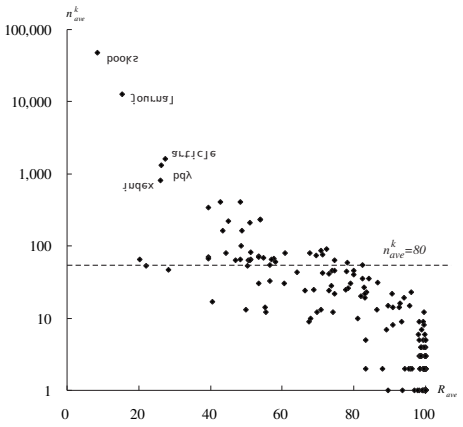

図 6　平均異なり率 R_{ave} と平均異なり語数 n^k_{ave} の関係 (前処理無)

数の平均 n^k_{ave} と平均異なり語率 R_{ave} の相関を散布図として示したものである．この図からわかるように，平均異なり語率 R_{ave} の値が 90% 以上の XML 部分文書に含まれている平均異なり語数 n^k_{ave} は約 80 語以下であり，先に説明したように平均異なり語率 R_{ave} の値が大きい XML 部分文書ほどその単語数，すなわち XML 部分文書のサイズは小さいという関係がある．

以上の点から，CPD は XML 部分文書の平均異なり語数 n^k_{ave} と平均異なり語率 R_{ave} によってある程度絞り込むことが可能であり，CPD の条件として以下の点を考慮することが有効であると思われる．

- 本実験では，ストップワード処理や接辞処理などの前処理の有無によって，利用した統計量の相違を調査したが，いずれの分析においてもその差異を見出すことはできず，前処理は統計量に影響を与えないことが判明した．
- 平均異なり語率 R_{ave} が 90% 以下の XML 部分文書のほとんどは，その部分文書中に 1,000 語以下の異なり語を含んでいる．つまり，CPD に相応しい XML 部分文書の条件として，その異なり語数はたかだか 1,000 語程度であることがわかる．
- 2.4 節で述べたように，我々のこれまでの研究成果である

表 3　統計量の解析結果 (XML 部分文書数 N 上位 20，前処理有)

要素名	部分文書数	単語数 n^w			異なり語数 n^k			平均異なり語率
	N	平均 (n^w_{ave})	最大 (n^w_{max})	最小 (n^w_{min})	平均 (n^k_{ave})	最大 (n^k_{max})	最小 (n^k_{min})	R_{ave} (%)
p	762,223	35	3,272	4	27	313	4	78.43
tmath	574,395	2	288	1	2	60	1	96.09
ref	395,933	5	15	3	5	15	3	100.00
it	394,549	2	149	1	2	96	1	97.21
au	317,457	2	28	1	2	26	1	99.96
entry	317,384	4	167	2	4	50	2	99.19
snm	311,257	1	15	1	1	15	1	100.00
ip1	178,788	32	1,529	1	24	400	1	74.69
obi	164,908	3	226	1	3	142	1	98.52
ti	159,565	4	65	1	4	48	1	99.13
pdt	154,978	4	7	1	1	7	1	100.00
yr	154,943	1	7	1	1	7	1	100.00
sub	154,324	1	18	1	1	15	1	99.82
bb	149,168	20	237	2	19	164	2	97.33
st	136,935	1	36	1	2	27	1	99.56
fnm	135,192	1	9	1	1	9	1	100.00
atl	134,247	5	70	1	5	54	1	99.35
b	123,463	2	273	1	2	86	1	98.54
pp	108,134	1	10	1	1	10	1	99.99
scp	107,544	1	18	1	1	14	1	99.99
181 種	6,802,061	2,222	894,853	1	234	64,181	1	38.85

表 4　統計量の解析結果 (XML 部分文書数 N 上位 20，前処理無)

要素名	部分文書数	単語数 n^w			異なり語数 n^k			平均異なり語率
	N	平均 (n^w_{ave})	最大 (n^w_{max})	最小 (n^w_{min})	平均 (n^k_{ave})	最大 (n^k_{max})	最小 (n^k_{min})	R_{ave} (%)
it	1,292,929	1	354	1	1	158	1	96.87
p	762,223	63	3,340	4	46	532	4	73.91
tmath	574,421	2	289	1	2	61	1	96.30
ref	395,933	5	20	3	5	20	3	100.00
au	317,749	2	46	1	2	40	1	99.96
entry	317,384	4	109	2	4	72	2	99.00
fnm	315,953	1	15	1	1	15	1	99.99
snm	312,017	1	28	1	1	25	1	99.99
sub	300,439	1	19	1	1	16	1	99.68
obi	252,145	3	383	1	3	219	1	98.13
ip1	183,567	59	1,574	1	42	412	1	71.30
b	162,582	2	308	1	2	125	1	98.54
ti	159,583	5	113	1	5	74	1	98.14
pdt	154,985	1	10	1	1	9	1	100.00
yr	154,949	1	10	1	1	9	1	100.00
bb	149,168	24	401	3	23	229	3	96.09
st	139,061	3	49	1	3	35	1	98.69
atl	134,293	7	94	1	7	62	1	98.89
scp	117,834	1	31	1	1	26	1	99.94
pp	108,135	1	12	1	1	12	1	99.99
art	81,544	9	10	2	9	10	2	98.17
183 種	8,224,053	3,693	1,530,446	1	384	93,216	1	37.03

「文脈検索」では，文脈を表現する XML 部分文書の root node に対し同名の兄弟ノードを持つことが多いという事実を利用して文脈を発見していた．この文脈の持つ特長は，本稿で定義する CPD も持っていると考えられるため，XML 部分文書の出現数 N の値が大きく，またその平均異なり語率 R_{ave} が小さな部分文書は，CPD として相応しいと考えられる．

- 3.1 節で述べたように，平均異なり語率 R_{ave} が 100% の XML 部分文書は少なくともデータ指向が強く，そのため，ベクトル空間モデルによって正確に検索できない，すなわち，CPD として定義されるべきではないと考えられる．同様に，平均異なり語率 R_{ave} が 90% 以上 100% 未満の XML 部分文書も同じ傾向を持っていると考えられるため，この XML 部分文書もまた CPD として定義されるべきではないと考えられる．

こうした事実を考慮し，例えば，平均異なり語率 R_{ave} が 90% 未満の XML 部分文書を CPD とすれば，CPD として定義される XML 部分文書数は INEX test collection を表現する XML 文書木から抽出される XML 部分文書数の約 3 割 (前処理ありの場合は 26%，前処理なしの場合は 23%) に減少し (図 7，8 参照)，XML 文書検索システムの高速検索が実現可能となる．また，INEX test collection の query/answer セットが決定されれば，さらに CPD として定義されるべきではない XML 部分文書の特長が明らかになり，XML 文書検索の更なる高速化が期待できる．

4. 関 連 研 究

文書検索の研究分野において，検索要求に対してそれに類似した文書の一部分だけを検索するという研究テーマは，2.3 節で述べたパッセージ検索 [13] が提案されてから非常に注目されている．これらの研究の主眼は，文書の一部分を検索することに置かれているが，見方を変えれば検索対象の文書の粒度 (単位) をどのように決定するかについて提案しているとも言え，単に検索精度を向上させるためだけではなく，検索システムのパ

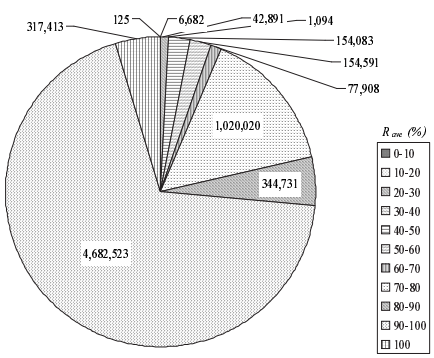

図 7　平均異なり率 R_{ave} による XML 部分文書数 N (前処理有)

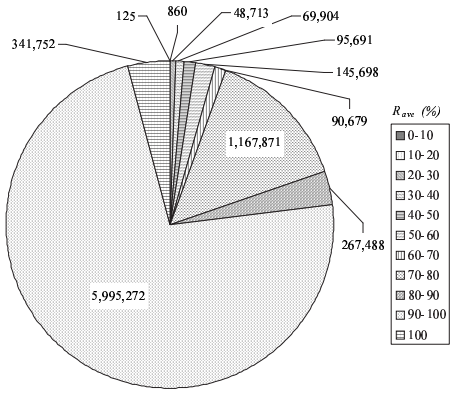

図 8　平均異なり率 R_{ave} による XML 部分文書数 N (前処理無)

フォーマンスの確保などにも利用されている．

近年，特にこれらの研究テーマが盛んに行われているのは，Web 文書検索の分野であり，文献 [11] や [14] では Web 文書間に張られているリンクを利用して文書間の関連度を計算し，それを基に Web 文書検索における検索対象文書粒度 (Information Unit) を決定しようとしている．また，半構造データにおいても同様の研究が始められており，Fine-grained Semi-structured Data などと呼ばれている [4]．もちろん，XML 文書検索の分野においても，検索対象となる文書粒度の決定は，検索精度の向上だけではなく，検索システムの scalability の確保やパフォーマンスの向上などさまざまな効果が期待され，今後議論されるべき研究テーマであると思われる．

5. おわりに

本稿では，問合せキーワードを利用した XML 文書検索システムを構築する際に生じる，検索対象 XML 部分文書数が膨大となることによる検索コストが増加するという問題に対して，XML 部分文書から抽出される単語数などの統計量を利用した検索対象 XML 部分文書の粒度決定法を提案した．また提案した手法を利用すれば，抽出可能な XML 部分文書の 3 割程度に文書数を抑えることができ，より高速な検索が実現可能であることが確認できた．本稿で提案した CPD の概念は，検索対象となる XML 文書が大きくなればなるほど検索システムの高速化を図るために必要であり，さらに検索精度を高めるために有効な手法であると考えている．

今後の課題としては，本稿で判明した CPD の条件を，さらに INEX test collection の query/answer セットを利用してより詳細に決定し，それを適用することによる，XML 文書検索システムの検索時間短縮の効果および検索精度の向上の確認，および，CPD の決定条件に利用した統計量について，計量情報学における理論的な裏づけをとることが挙げられる．

謝　辞

本研究の一部は，文部科学省科学研究費基盤研究 (課題番号はそれぞれ 11480088，14019064，14780325)，および科学技術振興事業団の戦略的基礎研究推進事業「高度メディア社会の生活情報技術」プログラムの支援によるものである．ここに記して謝意を表す．

文　献

[1] S. Boag, D. Chamberlin, M.F. Fernandez, D. Florescu, J. Robie, and J. Siméon. XQuery: A Query Language for XML. `http://www.w3.org/TR/xquery`, Nov. 2002. W3C Working Draft 15 November 2002.

[2] A. Bonifati and S. Ceri. Comparative Analysis of Five XML Query Languages. *ACM SIGMOD Record*, Vol. 29, No. 1, pp. 68–79, Mar. 2000.

[3] T. Bray, J. Paoli, C.M. Sperberg-McQueen, and E. Maler. Extensible Markup Language (XML) 1.0 (Second Edition). `http://www.w3.org/TR/REC-xml`, Oct. 2000. W3C Recommendation 6 October 2000.

[4] S. Chakrabarti. Text Search for Fine-grained Semi-structured Data. In *Tutorial Notes of the 28th International Conference on Very Large Data Bases*, pp. 115–135, Aug. 2002.

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

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

[7] K. Hatano, H. Kinutani, M. Yoshikawa, and S. Uemura. Information Retrieval System for XML Documents. In *Proc. of the 13th International Conference on Database and Expert Systems Applications*, Vol. 2453 of *LNCS*, pp. 758–767. Springer-Verlag, Sep. 2002.

[8] G. Kazai and T. Rölleke. A Scalable Architecture for XML Retrieval. In *Proc. of the First Workshop of the Initiative for the Evaluation of XML Retrieval*. ERCIM, Mar. 2003. (to appear).

[9] 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手法. 情報処理学会論文誌: データベース, Vol. 43, No. SIG2(TOD13), pp. 80–93, Mar. 2002.

[10] H. Kinutani, M. Yoshikawa, and S. Uemura. Identifying Result Subdocuments of XML Search Conditions. In *Proc. of the 2000 Kyoto International Conference on Digital Libraries: Research and Practice*, pp. 232–239, Nov. 2000.

[11] W.-S. Li, K.S. Candan, Q. Vu, and D. Agrawal. Retrieving and Organizing Web Pages by “Information Unit”. In *Proc. of the 10th International World Wide Web Conference*, pp. 230–244, May 2001.

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

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

[14] K. Tajima, K. Hatano, T. Matsukura, R. Sano, and K. Tanaka. Discovery and Retrieval of Logical Information Units in Web. In *Proc. of the 1999 ACM Digital Library Workshop on Organizing Web Space*, pp. 13–23, Aug. 1999.