# XMLデータベースの全文検索技術

## Full-text Search Engine for XML Database

宮澤　隆幸　　國分　智晴　　金輪　拓也
■ MIYAZAWA Takayuki　■ KOKUBU Tomoharu　■ KANAWA Takuya

**東芝は，XML（Extensible Markup Language）データをそのまま格納でき，かつ大容量データに対しても高速に検索できるXMLデータベースを研究・開発している。XMLデータベースをコンテンツ管理システムに適用する場合，非定型データを検索対象とするために，より高度な全文検索機能が要求される。**

**このような背景から，XMLデータベースの全文検索技術の開発を行い，構造化文書に対応したスコアリング（スコア付け）や同義語検索などの機能を実現した。また，全文検索用の索引としてNグラム索引に加えて形態素索引を利用できるようにし，索引のサイズを抑えながら高速な全文検索を実現した。**

Toshiba is carrying out research and development for new full-text retrieval software for XML database. The database stores large volumes of XML documents without modification, and the search engine works in it at high speed. An advanced full-text search engine is essential because XML data is becoming more and more common, and these circumstances compel us to develop full-text retrieval technology for its database. The functions of the search engine were extended to include synonym retrieval and a new scoring method for structured documents. Moreover, we have introduced a morphological index in addition to an N-gram index in order to reduce the index size for full-text retrieval.

## 1　まえがき

XMLは，インターネット上の交換及び蓄積のフォーマットとして標準化された新たなデータ記述言語として期待されている。東芝は，XMLの持つ柔軟性や拡張性を生かすため，構造を事前定義することなくXML文書を格納できるXMLデータベースの研究・開発を進めている[1), 2)]。

このXMLデータベースの適用分野の一つとして，コンテンツ管理システムが重要な位置を占めている。コンテンツ管理システムにおいては，日付や分類などの定型データのほかに，コンテンツの内容についての記述などの非定型データを検索対象とする必要がある。大量の非定型データの検索では，厳密な一致検索だけで必要なデータを検索することはユーザーにとって困難であり，検索結果のランキング表示や，絞込みのためのキーワード出現順序などの条件指定，表記揺れへの対応など，高度な全文検索機能が要求される。

ここでは，これらの要求に対応するために開発を進めてきたXMLデータベースの全文検索技術について述べる。

## 2　技術的な背景

### 2.1　XQuery

現在，XMLデータベースに対する問合せ言語としては，W3C（World Wide Web Consortium）で策定されているXQuery（XQuery 1.0：An XML Query Language）[3)]が業界標準になりつつある。

XQueryでは，階層構造に関する条件とノードの値の条件を記述できる。ノードの値の条件としては，数値の比較条件のほか，テキストの完全一致，部分一致（前方，中間，及び後方）の条件を指定することができる。

### 2.2　全文検索技術

一方，インターネットのサーチエンジンなどに代表される検索エンジンでは，文書の全テキストを検索対象とする全文検索機能を持つ。

これには，一般的に次のような機能がある。

(1)　検索結果をスコアリングして，上位のものからリストアップするランキング機能

(2)　アルファベットの大文字と小文字や，全角文字と半角文字，漢字の新字体と旧字体のような表記揺れを吸収する同一視検索機能

(3)　同義語辞書を用い，キーワードを展開して検索する同義語検索機能

(4)　キーワードの出現順序や出現位置（キーワードどうしの出現位置が指定した文字数以内など）に関する条件を指定する近傍検索機能

(5)　英語の語尾変化や，日本語の送りがなの変化，外来語表記などの揺れを吸収する検索機能

(6)　検索でヒットした文書を種文書として，それに類似する文書を検索する類似文書検索機能

このように，検索エンジンでは，全文検索での検索条件と

して様々な条件を指定できる。しかしその一方，文書単位の検索しか行うことができず，また，あらかじめ指定した属性による絞込みしか行うことができなかった。

## 3　XMLデータベースの全文検索技術

### 3.1　XQueryに対する全文検索機能の拡張

現在のXQueryでは，テキストに関する検索条件としては先に述べたとおり，完全一致検索と前方，中間，及び後方の各部分一致検索だけが存在する。XQueryの全文検索機能の強化に関してはW3Cで議論がなされており，XQuery 1.0 and XPath 2.0 Full-text[4]（以下，XQFTと略記）として策定が進められている。

XQFTの特長は，次のとおりである。

⑴　半構造化文書の検索　　従来の検索エンジンにおける全文検索では文書単位で検索し，検索結果も文書単位で返す。それに対し，XQFTではXML文書の特長を生かし，検索及び結果取得を文書の部分構造に対して行うことができる。

⑵　表現力及び拡張性　　AND/ORなどの論理式，近傍条件，同一視の有無，同義語検索指定を組み合わせて指定できる。例を**図1**に示す。ⓐの“ftcontains”がXQFTの文であることを示し，その前の“./title/text ()”が検索対象，ⓑ～ⓔが検索条件となる。意味としては，<book>タグの下の<title>タグの文字列の中に，ⓑ“web”と“site”と“usability”の三つの単語がⓒ6単語以内に出現するものを検索，このときⓓ大・小文字は同一視し，ⓔ同義語辞書として“user-dictionary”を使用した同義語検索を行う，ということになる。

```
db("bib")/book[./title/text() ftcontains ····· ⓐ
    "web" && "site" && "usability" ····· ⓑ
    window 6 words ····· ⓒ
    case insensitive ····· ⓓ
    with thesaurus at "user-dictionary"] ····· ⓔ
```

**図1．XQFTクエリの例** ― 複数の検索条件をまとめて記述した例である。従来の条件式の代わりに，このように複数の全文検索条件を組み合わせて記述することができる。

Example of XQuery 1.0 and XPath 2.0 Full-text (XQFT) query

⑶　スコアリングによるランキング機能　　ヒットした文書すべてではなく，スコアリングの結果に基づくランキングを行い，上位n件を取得できる。

⑷　XQueryへの統合　　図1で示したように，従来の条件式の代わりに全文検索の条件文を書くことが可能であり，XQuery式の中に埋め込むことができる。

このように，XQFTはXQueryと高い親和性を保ちつつ，高度な全文検索機能を記述することができる。そこで今回の全文検索技術の開発でも，XQFT準拠の形で実現することにした。**表1**に検索エンジン，XQuery，XQFTそれぞれの特性についてまとめた。

**表1．検索システムの特性**

Characteristics of retrieval systems

| | 構　造 | 検索条件 | ランキング |
|---|---|---|---|
| 検索エンジン | 単純なパス表現 | 単語，フレーズ，近傍条件 | ○ |
| XQuery | 強力なツリー構造指定 | 部分文字列マッチだけ＋文字列操作関数 | × |
| XQFT | 強力なツリー構造指定 | 同一視条件や近傍条件を組み合わせて指定可能 | ○ |

### 3.2　実装上のポイント

この節では，全文検索機能の実装で重要なポイントである，スコアリング，形態素解析を利用した索引，及びXQFTの日本語対応について説明する。

**3.2.1　スコアリング**　　従来の文書に対するスコアリング アルゴリズムをXML文書に拡張することにより，様々な部分構造単位でのランキングを可能にした。このため，より容易に所望の情報にアクセスできるようになった。

文書に対するスコアリング アルゴリズムには様々な手法が存在するが，代表的なものがtf-idf（term frequency - inverse document frequency）法[5]を用いたものである。文書$d$における，キーワード$k_i$（$i=0,...,n$）の出現回数$tf$（term frequency）を$tf(k_i, d)$，$k_i$が出現する文書数$df$（document frequency）を$df(k_i)$，全文書数を$N$とする。$k_i$が全文書中のどれくらいの文書に出現するかを表す尺度である$idf$（inverse document frequency）を$idf(k_i) = \log(N/df(k_i)) + 1$として，文書$d$のスコアscore$(d)$は式⑴で与えられる。

$$\mathrm{score}(d)=\sum_{i=0}^{n}(tf(k_i,\ d)\ \cdot idf(k_i)) \qquad (1)$$

すなわち，出現文書数が少ないキーワードが多く出現するほどスコアの値は大きくなる。

XML文書に対するスコアリングでも，基本的にはこのtf-idf法を用いることができるが，XQFTではスコアを付ける対象と検索対象とを構造上の別の部分に指定することができるため，その場合のスコアリング アルゴリズムを定める必要がある。そこで，今回新たに開発したアルゴリズムでは，まず検索条件ごとに検索対象ノードそれぞれについてtf-idf法によりスコアを算出し，その和を検索条件ごとのスコア値とした。更に，検索条件に応じたマージ（併合）を行って，スコア対象ノードのスコア値とした。

今回開発のスコアリング機能を使用したランキングの例を**図2**に示す。for文の“score $s”という記述により，for文の対象となる要素ごとにスコアが計算され，スコア値が変数

一般論文

```
クエリ
for $patent score $s in                                          検索条件
  db("test")//jp-official-gazette[./description//text()[. ftcontains "データベース" && "XML"
                          case insensitive width insensitive with thesaurus at "SystemDic"]‥‥(1)
    and ./bibliographic-data/classification-ipc/main-clsf/text() [starts-with(.,"G06F")] ‥‥‥‥‥‥(2)
    and ./bibliographic-data/publication-reference/document-id/date/text()
                                                    [. > 19990100 and . < 20040700]]‥‥(3)
order by $s descending
return <result score="|$s|"><invention-title>|$patent//invention-title/text()|</invention-title></result>
```

```
検索結果
ヒット件数：186件
---------- [1/186] ----------
<result score="384"><invention-title>XML文書検索装置
</invention-title></result>
---------- [2/186] ----------
<result score="367"><invention-title>様々な形式の内容を出版するためのリポジトリ
</invention-title></result>
---------- [3/186] ----------
<result score="247"><invention-title>データベース管理方法及びシステム
</invention-title></result>
```

**図2. スコア計算を使用した検索の例** — スコア値を計算し，その値でソートして上位から結果を取得する例である。検索結果のscore = "〜"の部分がスコア値である。スコア値の降順に結果が表示されている。

Example of retrieval with scoring

$sに格納される。“order by $s descending”という記述により，スコア値$sの降順に結果がソート（整列）される。これにより，スコア値が大きい検索結果から順に結果を取得することができる。検索条件は，次の三つである。

（1） 発明の詳細な説明を表す<description>タグ以下に，“データベース”と“XML”のキーワードを含み，アルファベットの大・小文字は区別せず（case insensitive），英数字の全・半角は区別せず（width insensitive），同義語辞書名“SystemDic”で同義語検索を使用する（with thesaurus at "SystemDic"）。

（2） 国際特許分類（IPC）が“G06F”で始まる。

（3） 公報の公開日が1999/1/1〜2004/6/30の範囲にある。

このようにXQFTでは，構造の任意の部分に対して，それぞれ別々の検索条件を記述できることが特徴の一つである。また将来，文書に別のタグが追加になった場合でも，文書を再登録することなく，追加されたタグに関する条件をクエリに追記することで対応することができる。

**3.2.2 形態素解析を利用した索引** データベースで全文検索を高速に処理するために，何らかの索引を使用する方法が多く用いられている。当社の従来のXMLデータベースでは，このための索引としてNグラム索引を付加できるようになっていた。Nグラム索引とは，文字列をN文字単位で分割した部分文字列を索引語として，その索引語の出現位置情報を記録した転置索引である。このNグラム索引は，すべての部分文字列を再現できるため漏れのない検索が可能である。その反面，文字数とほぼ同数の位置情報を管理する必要があるため，索引のサイズが増大するという問題がある。

そのため，形態素解析(注1)(6)を利用した形態素索引を新たに導入した。これは，文字列を形態素解析した結果を索引語として用いるというものである。

（注1） 自然言語処理技術の一つで，文を解析して，意味を持つ最小単位（形態素）の列に分割し，品詞を割り当てる。

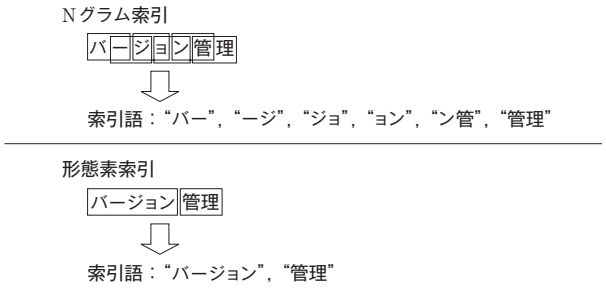


**図3. Nグラム索引と形態素索引** — 同じ文字列を与えた場合に，どのような索引語が作られるかを示す。Nグラム索引と比較して，形態素索引では作られる索引語が少ない。

N-gram index and morphological index

同じ文字列を与えた場合に，Nグラム索引と形態素索引でそれぞれどのような索引語が切り出されるかを**図3**に示す。この例では，“バージョン管理”という文字列に対してはNグラム索引では6個の索引語が切り出されるのに対し，形態素索引では2個の索引語しか切り出されない。

このように，特に自然文に近い文字列の場合，形態素索引を使用することにより大幅に索引語を減らすことができる。反面，形態素索引は“バー”や“ージ”などを索引語としないため，検索漏れが生じる場合がある。Nグラム索引と形態素索引の比較を**表2**に示す。

**表2. Nグラム索引と形態素索引の比較**

Comparison of N-gram index against morphological index

| | 索引サイズ | 検索漏れ |
|---|---|---|
| Nグラム索引 | 大 | なし |
| 形態素索引 | 小 | あり |

Nグラム索引と形態素索引は相補的な関係にあるため，XML文書の構造単位でこれらの索引を付けるか付けないかを設定できるようにした。これにより，検索時の条件や文書構造の特徴に応じた索引の設定が可能になる。例えば，商品コードのような漏れのない検索が必要なものには“Nグラム索引を付ける”，本の概要のような自然文に近いものには“形態素索引を付ける”，と設定する。このようにすると，両方ともNグラム索引を付けた場合と比較して，索引のサイズを抑えつつ，商品コードと本の概要の両方に関して高速な全文検索が可能になる。

**3.2.3 XQFTの日本語対応** 現在策定が進められているXQFTでは，英語をはじめとする欧米の言語を前提とした全文検索機能については言及されているが，日本語特有の機能については言及されていない。言語固有の機能に

関しては，使用言語に応じて実装の際に拡張する必要がある。そこで，XQFTのほかの機能に合わせた形で次のように拡張を行った。

⑴　近傍検索における距離の指定を文字単位とする（letters）。

⑵　全角文字と半角文字を同一視するかどうかの指定を追加する（width sensitive / insensitive）。

この拡張の具体例を次に示す。

db("bib")//書籍[./書名/text() ftcontains "web" &&“検索” case insensitive width insensitive window 20 letters]

これは，データベース“bib”内のすべての＜書籍＞タグのうち，＜書名＞タグに“web”と“検索”を含むものを検索する。このとき検索条件として，次の⑴と⑵を満たすものを検索する。

⑴　英数字の大・小文字，及び全・半角は区別しない。

⑵　二つのキーワードが20文字以内に出現する。

この拡張により，検索対象の文書中に英数字の大・小文字及び全・半角が混在している場合にも，表記揺れを吸収した全文検索を行うことができる。

## 4　あとがき

コンテンツ管理システムをはじめとする様々なシステムへのXMLデータベースの適用を可能にするために，XQFTに準拠した形でスコアリングや近傍検索，同一視検索，同義語検索などの全文検索技術を開発した。また，Nグラム索引に加えて形態素索引を利用できるようにしたことにより，データベースのサイズを抑えながら高速な全文検索を実行することができるようになった。

今回開発した技術は，2006年11月28日に東芝ソリューション（株）より製品化されたTX1™ V2に組み込まれている。

今後もXMLデータベースの適用分野を拡大するために，機能面及び性能面の強化を図っていく。

## 文　献


⑴　服部雅一，ほか．高速性と信頼性を両立したコンテンツ管理向けネイティブXMLデータベース．東芝レビュー．59，2，2004，p.54－57.

⑵　谷川　均，ほか．大規模でも高速な検索を実現するXMLデータベースTX1．東芝レビュー．60，7，2005，p.71－75.

⑶　W3C. "XQuery 1.0: An XML Query Language." <http://www.w3.org/TR/xquery/>,（参照2006-12-19）.

⑷　W3C. "XQuery 1.0 and XPath 2.0 Full-Text". <http://www.w3.org/TR/xquery-full-text/>,（参照 2006-12-19）.

⑸　徳永健伸．情報検索と言語処理．東京，東京大学出版会，1999，234p.

⑹　住田一男．新世紀を拓く東芝の技術 5．自然言語処理．東芝レビュー．56，8，2001，p.68－69.



宮澤　隆幸　MIYAZAWA Takayuki

研究開発センター 知識メディアラボラトリー研究主務。XML データベース及びナレッジマネジメントの研究・開発に従事。IEEE，情報処理学会会員。

Knowledge Media Lab.

國分　智晴　KOKUBU Tomoharu

研究開発センター 知識メディアラボラトリー。XML データベース及び情報検索の研究・開発に従事。

Knowledge Media Lab.

金輪　拓也　KANAWA Takuya

研究開発センター 知識メディアラボラトリー研究主務。XML データベース及びナレッジマネジメントの研究・開発に従事。

Knowledge Media Lab.


一般論文