文書間の類似度の概念を含む構造化文書操作記述方式

1P－2

品川 徳秀 †　北川 博之 ‡　石川 佳治 ‡
† 筑波大学 工学研究科　‡ 筑波大学 電子・情報工学系

1 はじめに

近年、構造化文書が多く利用されるようになってきている。それに伴い、大量の構造化文書に対する検索、再構築、要約などの各種処理が必要とされている。

このような背景の下に、構造化文書に対する問合せ記述方式が幾つか提案されている。しかし、その多くは半構造化データとしての構造化文書の処理に向いたものとなっている [1]。これらは、文書要素の種類や位置関係といった構造に関する条件や、その記述内容と文字列との包含関係や数値としての大小関係といった「値」に関する条件に基づいた記述を行なう。しかし、文書要素間の類似度のような記述内容の扱いは不十分である。

一方、内容処理としては文書検索や自動的な要約生成、話題抽出等の研究が行なわれている [2] [3]。これらは、その記述内容である文章を統計的、言語学的な手法等によって解析して得られる特徴量に基づいて処理を行なう。文書検索ではベクトル空間モデルが広く知られており、類似度を用いた文書の類似性判定が実現されている。しかし、文書は語の集合とみなされ、文書構造はほとんど扱われない。また、要約生成等は様々な特徴量を利用した文書変形の処理であり、一般にその方式は処理に依存して特化したプログラムとして実現されている。これらの内容処理では、問合せ記述に見られるような汎用的な記述方式が確立していないため、文書構造の変換等を汎用的に扱う能力を持たない。

上記のようなアプローチによって実現される機能は相補的な関係にあると考えられる。本研究ではこれらの記述を一つの枠組に融合する事を目的とする。これにより、文書の構造的な側面と記述内容の意味的な側面を持つ処理を統一的に記述する事が可能となる。

以下では、提案する枠組によって可能となる処理を例示し、その記述方式について説明する。

2 構造化文書処理例

本稿で扱う構造化文書として、XML 文書を想定する。複数の記事を含む新聞に対する以下のような処理を考える。個々の記事には、記事の見出しと、どの国に関する記事であるかが記述されており、本文は段落毎に区切られているものとする。

```
<!ELEMENT newspaper article+>
<!ELEMENT article  (title country paragraph+)>
<!ELEMENT title     CDATA>
<!ELEMENT country   CDATA>
<!ELEMENT paragraph CDATA>
```

これに対し、記事を国別にグループ化し、キーワード群「EC, XML」との類似度に基づいて各グループ中の記事のランキングを行ない、各々から上位 3 件ずつ選択し、出力を作成する。その際、各記事中でタイトルとの類似度が最も高かった段落について、強調した表示を行なうような指定を行なうものとする。

この例において、グループ化や選択等の構造変換は問合せ言語において対象となる処理であり、類似度判定やそれに基づくランキング等は文書検索で行なわれてきた処理である。また、重要部分を特定してそれを抜き出す処理は自動要約の基礎でもある。このような、従来は個別に扱われていた処理を統合的に扱うための記述方式を次に説明する。

3 統合記述方式

本稿で述べる記述方式を検討するに当たって次を考慮した。

1. 様々な文書構造に柔軟に対応できる事
2. 低水準で繁雑な記述を避ける事
3. 文書の構造と内容処理の直交性を高める事

本研究で提案する記述では、構造化文書に対する操作を変換規則に基づいて記述する。変換規則にはパターン駆動型の記述方式を採用した。

変換規則は、“pattern” で始まり XML-QL をベースとした変換式の集合として与える。即ち、変換式の基本文法は次の通りである。

```
pattern
where パターン式 (及び変数束縛)
      [in ソース文書もしくは文書要素]
[order-by 順序評価基準式]
construct 出力生成式
```

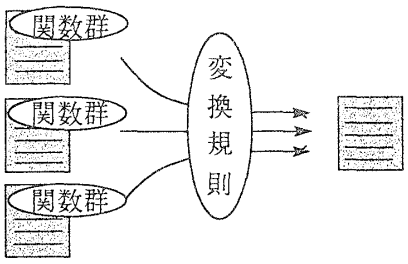
与えられた変換規則に対して、文書要素がいずれかの変換式で処理されるか末端に到達するまで、順次パターン照合が行なわれる。特に、出力生成式中で生成される文書要素を再帰的に処理する必要がある場合には、その文書要素の直前に “#” を記述しておく* 。これらは、パターン式に適合する全ての文書要素の処理が完了した後に、同様に処理される。これにより、多段階のステップを踏むような変換を単一の処理過程に織り込む事ができる。

内容処理に必要な特徴量はユーザ定義関数を通して取得する。ここで、操作系は複数の DTD に対応できねばならないが、各 DTD の設計は処理系の目的に適するとは限らず、更にそれの定める語彙はまちまちである。このような多様性を吸収するため、ユーザ定義関数群は DTD 毎に定義する。文書要素の特徴量の取得方法を抽象化する事で同一の方法で

*但し、一つの出力生成式中で “#” で指定される文書要素を入れ子にはできない。

An Operations Scheme of Structured Documents Considering Similarities of Document Elements
Norihide Shinagawa †, Hiroyuki Kitagawa ‡ and Yoshiharu Ishikawa ‡
† Doctoral Program in Eng., Univ. of Tsukuba,
‡ Institute of info. Sci. and Elec., Univ. of Tsukuba

利用が可能になり、文書の構造と内容処理との直交性を高める事も容易になる。その関数群はアダプタとしての役割を果たしているとみなせる。



ユーザ定義関数は Java 等の言語で外部プログラムとして定義可能とする。関数の引数及び返値として有効なデータ型は、数値、文字列、文書要素である。ここで、文書要素の種類に応じて個別の処理が必要とされる事が想定される。この点から、通常の関数とは別に、"文書要素名. 関数名 ()" という形でメソッドとして関数を定義し、"文書要素変数. 関数名 ()" という形式で利用できるものとする。

4 記述例

図 1, 図 2 に 2 節で示した操作の実際の記述例を示す。ここで、"position(a)" は文書要素 a の母集団中での出現位置番号を、"rank(a,b)" は文書要素 a の母集団における b の値での降順順位を返す組み込み関数とする。

図 1 は、ここで用いられる関数群の定義である。一般関数と title 文書要素のメソッドがユーザ定義関数として与えられている。

図 2 中の一つ目の変換式は、newspaper 内の article を country の値別に localnews にグループ化する。localnews は country と articles を含み、articles 内の article は "EC, XML" との類似度でソートされている。また、各 article は再帰的パターン照合の対象となる。これが生成する文書要素は次の形式に沿う。

```
<!ELEMENT newspaper localnews+>
<!ELEMENT localnews (country articles)>
<!ELEMENT articles  article+ )>
```

二つ目の変換式は、articles 中で 3 番目までに出現する article を選出し、そのタイトルと最も類似した paragraph のみに emphasis タグを付加している。ここでは、article は title と paragraph のみを含み、結果的に次の形式となる。

```
<!ELEMENT article (title (paragraph|emphasis)+>
<!ELEMENT emphasis paragraph>
```

5 まとめと今後の課題

従来の構造化文書を対象とした問合せ言語では述内容に対する文字列パターンマッチング処理と構造変換が主な操作としているものが大部分であった。文書検索においてはベクトル空間モデル等の利用によって類似検索を扱えるものの、文書構造の再構成などはほとんど考慮されていない。一方、要約生成や話題抽出などの内容処理は目的毎の個別アプリケーションプログラムで独立に処理されている場合がほとんどである。本稿では、これらの処理を、パターン駆動型の変換規則と DTD 毎に対して定義

```
function    similarity( element $t1, string $t2 )
defined-by  "http://..."

function    title.similarity( element $text )
defined-by  "http://..."
```

図 1: 上記の文書構造に対する関数定義

```
pattern
where     <newspaper> </> in $n
construct <newspaper>
  where     <article> <country> $c </> </> in $n,
  construct <localnews>
              <country> $c </>
              <articles>
    where       <article>
                  <country> $x </> content_as $a,
                </> in $n,
                $x = $c
    order-by    similarity($a,"EC XML") descending
    construct   #<article> $a </>
          </> </>

pattern
where     <article> </> element_as $a,
          position($a) <= 3
construct <article>
  where     <title> </> $t in $a,
            <paragraph> </> element_as $p in $a,
  construct $t
    where     rank($p,$t.similarity($p)) = 1
    construct <emphasis> $p </>

    where     rank($p,$t.similarity($p)) > 1
    construct $p
          </>
```

図 2: 処理例を実現する変換規則定義

された関数群を利用する事で、入力となる文書の構造と内容の処理の直交性を保ちながら統合する記述方式について説明した。

今後、記述方式のより詳細な検討を行ない、内容処理と問合せ処理を柔軟に組み合わせた記述を可能とする必要がある。例えば、[4] [5] 等の手法の記述を試みる予定である。また、既存の記述方式との親和性についても考慮する必要があると考える。

参考文献

[1] 田島 敬史. "半構造データのためのデータモデルと操作言語", 情報処理学会論文誌 データベース, Vol.40, No.SIG 3 (TOD 1), 1999.2.

[2] G. Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*, Addison-Wesley, Reading, MA, 1989.

[3] 奥村 学, 難波 英嗣. "テキスト自動要約に関する研究動向". 自然言語処理,「テキスト要約のための言語処理」特集号, Vol.6, No.6, 1999.7.

[4] 品川徳秀, 北川博之. "内容解析に基づく文書構造の自動抽出", 情報処理学会研究報告, Vol.98, No.58, 1998.7.

[5] 品川徳秀, 北川博之. "ユーザ視点に即した仮想 WWW ページの動的生成による閲覧支援" 情報処理学会研究報告, Vol.99, No.61, 1998.7.