November 28, 2011

BaseX における日本語全文検索の実装について

平井俊夫 <toshio.hirai@gmail.com>

この文書は、XML データベースシステム BaseX[1]の拡張機能である XQuery and XPath Full Text 1.0 (W3C Recommendation 17 March 2011)[2] に関する実装（以下、XQuery Full-Text と記述する）において、日本語で記述された文書をこれに適合させるための方法を、極めて簡潔に説明したものである。

(1) イントロダクション

日本語で記述された文書の字句解析は、形態素解析によって行われる。BaseX では、形態素解析を行うための外部ライブラリとして「Igo - Java 形態素解析器 (ver 0.4.3)[3]」を使用している。

Igo を使う利点として、

- 著名な形態素解析器「MeCab」と解析結果が互換であること
- MeCab プロジェクトが配布する辞書を使用できること
- Java で実装された形態素解析器のなかでは、比較的高速であること

などがあげられる。

BaseX において、日本語の全文検索機能を有効にするためには、クラスパス上に igo-0.4.3.jar パッケージを配置しなければならない。また、Igo が使用する辞書ファイルを、BaseX ホームディレクトリ[4] の以下の場所に、適切に展開する必要がある。

```
├ [HOME_DIR]
    └ etc
        └ ja
            ├ char.category
            ├ code2category
            ├ matrix.bin
            ├ word.ary.idx
            ├ word.dat
            ├ word.inf
            └ word2id
```

なお、辞書ファイルは以下のいずれかの場所から入手することが可能である（コーパスの編纂元が異なる 2 種類の辞書を用意している）。

IPA Dictionary: http://files.basex.org/etc/ipadic.zip

NAIST Dictionary: http://files.basex.org/etc/naistdic.zip

(2) 字句解析

例えば、「私は本を書きました。」という文は、形態素解析器によって、以下のように分解される。

```
私は本を書きました。
私　　名詞,代名詞,一般,*,*,*,私,ワタシ,ワタシ
は　　助詞,係助詞,*,*,*,*,は,ハ,ワ
本　　名詞,一般,*,*,*,*,本,ホン,ホン
を　　助詞,格助詞,一般,*,*,*,を,ヲ,ヲ
書き　動詞,自立,*,*,五段・カ行イ音便,連用形,書く,カキ,カキ
まし　助動詞,*,*,*,特殊・マス,連用形,ます,マシ,マシ
た　　助動詞,*,*,*,特殊・タ,基本形,た,タ,タ
。　　記号,句点,*,*,*,*,。,。,。
EOS
```

分解された個々の要素のうち、単語部分は「表層形(Surface)」、解析内容は「形態素(Morpheme)」と呼ばれる。形態素の構成要素は以下の通りである。

```
品詞,品詞細分類 1,品詞細分類 2,品詞細分類 3,活用形,活用型,原形,読み,発音
```

このうち、表層形はそのままトークンとして使用される。また、形態素の解析内容は、索引化や、ステミング（語幹抽出処理）において使用される。

(3) 構文解析と索引化

解析結果のトークンを１単位とし、これを BaseX の全文検索インデクサに順次渡すが、この時、インデックスのサイズを小さくするため、また、全体の文脈における検索結果に影響を与えないものとして、以下の品詞を意図的に除外している。

- 記号
- フィラー
- 助詞
- 助動詞

従って、先の例では、「私」「本」「書き」のそれぞれのトークンがインデクサに渡されることになる。

(4) トークンの扱いについて

Unicode における EastAsianWidth[5]付則に定義される、Fullwidth と Halfwidth に対しては両者を区別しないように考慮されている（いわゆる、全角／半角問題）。例えば、「ＸＭＬ」と「XML」は同一の語句として処理される。

また、多言語で記述されたハイブリッドな文書を考慮し、XQuery Full-Text で規定されている以下のオプションは、ISO_8859_1(Latin-1)文字セットで構成された単語に対して有効である。

- Case Option[6]
- Diacritics Option[7]

(5) ステミング[8]

日本語におけるステミングは、形態素解析の結果において、動詞と形容詞の原形を使用することによって実現している。

例えば、「私は本を書いた」と「私は本を書く」という二つの文において、動詞部分の解析結果に着目すると、原形は同一であることがわかる。

```
書く　　動詞, 自立, *, *, 五段・カ行イ音便, 基本形, 書く, カク, カク
```

```
書い　　動詞, 自立, *, *, 五段・カ行イ音便, 連用タ接続, 書く, カイ, カイ
た　　助動詞, *, *, *, 特殊・タ, 基本形, た, タ, タ
```

ステミングオプションが有効な場合、表層形でなく、原形を処理する。なお、助動詞はトークンから常に除外されるので、活用を考慮する必要はない。従って、以下に示す 2 種類のクエリは、同じ結果(true)を返す。

```
'私は本を書いた' contains text '書く' using stemming using language 'ja'
```

```
'私は本を書く' contains text '書いた' using stemming using language 'ja'
```

(6) ワイルドカード[9]

XQuery Full-Text におけるワイルドカードオプションは、日本語についても有効である。一般に、「芥川龍之介」の名は「竜之介」とも表記されるが、このようなケースにおいて以下のクエリは同一の結果(true)を返す。

```
'芥川龍之介' contains text '.之介' using wildcards using language 'ja'
```

```
'芥川竜之介' contains text '.之介' using wildcards using language 'ja'
```

しかしながら、注意を要する点として、このクエリを以下のようにすると、結果は false となる。

```
'芥川龍之介' contains text '芥川.之介' using wildcards using language 'ja'
```

これは、クエリにおけるメタキャラクタの前後の単語境界を判別できないためである。

この場合、単語境界に対して、意識的に空白を挿入するか、

```
'芥川龍之介' contains text '芥川　.之介' using wildcards using language 'ja'
```

以下のようにクエリを変更する必要がある。

```
'芥川龍之介' contains text '芥川' ftand '.之介' using wildcards using language 'ja'
```

以上

参考リンク

[1] BaseX | The XML Database: http://basex.org/

[2] XQuery and XPath Full Text 1.0: http://www.w3.org/TR/xpath-full-text-10/

[3] Igo - a morphological analyzer: http://igo.sourceforge.jp/

[4] Configuration: http://docs.basex.org/wiki/Configuration#Home_Directory

[5] http://unicode.org/Public/UNIDATA/EastAsianWidth.txt

[6] XQuery and XPath Full Text 1.0: http://www.w3.org/TR/xpath-full-text-10/#ftcaseoption

[7] XQuery and XPath Full Text 1.0: http://www.w3.org/TR/xpath-full-text-10/#ftdiacriticsoption

[8] XQuery and XPath Full Text 1.0: http://www.w3.org/TR/xpath-full-text-10/#ftstemoption

[9] XQuery and XPath Full Text 1.0: http://www.w3.org/TR/xpath-full-text-10/#ftwildcardoption