形態素解析での数詞の未知語処理

Unknown Numeral Word Processing in Morphological Analysis

青木和夫、中山章弘、松崎剛士
Kazuo Aoki, Akihiro Nakayama, Tsuyoshi Matsuzaki
日本アイ・ビー・エム（株） ソフトウェア開発研究所
Software Development Laboratory – Yamato (YSL), IBM Japan, Ltd.

1. はじめに

文を品詞付き単語に分かち書きする形態素解析は、検索エンジンやテキスト・マイニングに代表される自然言語処理アプリケーションの前処理として、必ず必要とされる機能である。形態素解析は、品詞などの形態素の特性を定義している単語辞書と、文法ルールやコストを定義している文法辞書を用いる［1]。形態素解析中に辞書に未登録なトークンは、接続コスト最小法で文全体のコストが高くなるように「未知語」の品詞を定義する。

形態素解析で未知語と推定される単語は、辞書に登録されてなかったものと、辞書に登録できなかったものがある。前者は、新語、専門語、方言、口語などで、これらは事前にサンプル文を用いてできるだけ未知語を特定して[2]、それらを辞書に登録することにより解決できる。しかし後者は、無数にバリエーションのある数詞(150 億光年, 2003 年, 第 50 回, 摂氏 28 度, 42.195 キロ, 146 億光年, 99 パーセント、など)で、これらは全てを辞書に登録することができない。

筆者らは、この数詞の未知語を解決するために、弊社の形態素解析に正規表現ルールを用いて実験し、その有効性を確認した。以下に、その手法と日本語だけでなく中国語と韓国語への適用を説明する。

2. 未知語処理

2.1. 従来の文字種チェック

図１に弊社の形態素解析のブロック図を示す。今までの未知語処理は、単語候補抽出処理の辞書引きで見つからなかったトークンに対して文字種をチェックして、数字列、カタカナ列、英字・記号列などでひと塊としていた。その際、数字列には「数字」の品詞を、カタカナ列には「未知語-固有名詞」の品詞を、その他の文字列には「未知語」の品詞を付けていた。

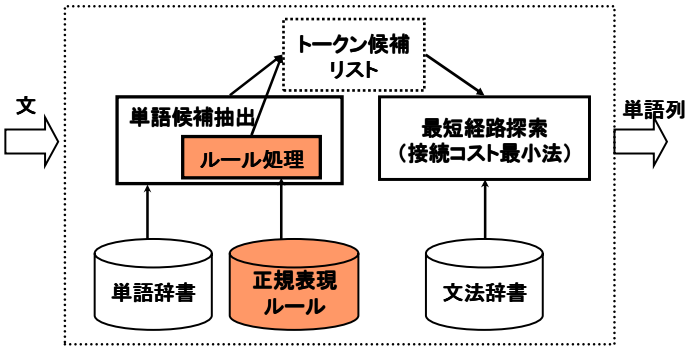

図 1 形態素解析のブロック図

2.2. 新しい手法

従来の文字種チェックの代わりに、図１のオレンジ色で示したように、正規表現ルールを用いて入力文からそのルールにマッチしたトークンをトークン候補リストに追加する処理を追加した。この結果、数字列、カタカナ列、数詞、 URL 、 eMail などの無数にバリエーションのあるトークンを正しく認識することができるようになった。

以下にその正規表現ルールと処理ロジックを説明する。

2.2.1. 正規表現ルール

弊社の形態素解析は、オープンなクラスライブラリーの ICU を利用して Unicode をサポートしている［3]。ICU の正規表現ルールは Perl の正規表現に準拠しており、ルールをフラットテキストで定義した後で ICU のツール genbrk.exe を用いてバイナリー形式に変換して実行時に使用する。

筆者らが定義した正規表現ルールは、基本ルールと数詞ルールである。前者の基本ルールは言語に依存しないものや ASCII 文字だけで定義できるもので、数字、URL、eMail、ファイルパス、電話番号、同一文字列などである。これらは、 全ての言語で同一のルールを定義できる。一方、後者の数詞は言語や文化によって表現が異なり、今回は、日付、時間、通貨に対して日本語、中国語、韓国語で共通に使えるルールを定義した。図２は日付のルールの例である。言語によって異なる「年」、「月」、「日」などの数詞の接辞は Unicode エスケープ記号を用いて１６進値で定義した。ルールの識別は、最後の行の{}の中の整数によって行う。この例の日付ルールの識別番号は４０１である。

```
$CJKYearPrefix = (¥u5e73 ¥u6210)                    //明治
       | (¥u662d ¥u548c)                            //大正
       | (¥u5927 ¥u6b63)                            //昭和
       | (¥u660e ¥u6cbb);                           //平成
$CJKYearSym    = ¥u5e74 | ¥ub144;                   //年 | 년
$CJKMonthSym   = ¥u6708 | ¥uc6d4;                   //月 | 월
$CJKDaySym     = ¥u65e5 | ¥u53f7 | ¥uc77c;          //日 | 号 | 일
$CJKYear  = (((($CJKYearPrefix $Whitespace?)? $CJKWholeNumber $Whitespace
       | ($CJKYearPrefix? ¥u5143)) $CJKYearSym;              //元
$CJKMonth = $CJKWholeNumber $Whitespace? $CJKMonthSym;
$CJKDay   = ($CJKWholeNumber $Whitespace? $CJKDaySym)
       | (¥u5409 ¥u65e5)                            //吉日
       | (¥u5143 ¥u65e5)                            //元日
       | (¥u5143 ¥u65e6);                           //元旦
$CJKDate = (($CJKYear $Whitespace?)? $CJKMonth ($Whitespace? $CJKDay)?)
       | $CJKYear | $CJKDay;
$CJKDate    {401};
```

図 2 正規表現ルールの例

2.2.2. 正規表現ルールの処理ロジック

正規表現ルールを処理するロジックには、ICU の

RuleBasedBreakIterator クラスの次のメソッドを利用した。

- following(offset)：offset から始まる文字列がルールにマッチした場合、マッチした文字列の終端のオフセット値を返す。
- getRuleStatus()： マッチしたルールの識別番号を返す。図２の例では、最後の行で指定した４０１が返る。

図３に、形態素解析の中で実装したロジック例を示す。これは入力文のオフセットからトークンを見つけ出す処理の一部である。最初に getRuleStatus() を使用してマッチしたトークンのルール番号を入手して、それぞれに対応したトークンに適切な品詞を付けてトークン候補リストに追加する。例えば、もしルール番号が４０１であれば日付のトークンなので、「数詞」の品詞を付けてトークン候補リストに入れる。また２０４であれば、カタカナ列なので、「未知語-固有名詞」の品詞を付けてトークン候補リストに入れる。

```
$CJKYearPrefix = (¥u5e73 ¥u6210)              //明治
       | (¥u662d ¥u548c)                      //大正
       | (¥u5927 ¥u6b63)                      //昭和
       | (¥u660e ¥u6cbb);                     //平成
$CJKYearSym    = ¥u5e74 | ¥ub144;             //年 | 년
$CJKMonthSym   = ¥u6708 | ¥uc6d4;             //月 | 월
$CJKDaySym     = ¥u65e5 | ¥u53f7 | ¥uc77c;    //日 | 号 | 일
$CJKYear  = (((($CJKYearPrefix $Whitespace?)? $CJKWholeNumber $Whitespace?)
       | ($CJKYearPrefix? ¥u5143)) $CJKYearSym;          //元
$CJKMonth = $CJKWholeNumber $Whitespace? $CJKMonthSym;
$CJKDay   = ($CJKWholeNumber $Whitespace? $CJKDaySym)
       | (¥u5409 ¥u65e5)                      //吉日
       | (¥u5143 ¥u65e5)                      //元日
       | (¥u5143 ¥u65e6);                     //元旦
$CJKDate = ((($CJKYear $Whitespace?)? $CJKMonth ($Whitespace? $CJKDay)?)
       | $CJKYear | $CJKDay;
$CJKDate    {401};
```

図 3 処理ロジックの例

3. 解析結果

図４は、日付、時間、通貨の数詞を正規表現ルールで定義しない場合と定義した場合の解析結果である。

日本語（ルール無し）

今は	副詞
2004	NUMBER
年	接辞
1	NUMBER
月	接辞
15	NUMBER
日	接辞
です	助動詞
今は	副詞
15	NUMBER
時	接辞
45	NUMBER
分	接辞
です	助動詞
今は	副詞
3000	NUMBER
円	接辞
です	助動詞

日本語（ルール有り）

今は	副詞
2004年1月15日	**DATE**
です	助動詞
今は	副詞
15時45分	**TIME**
です	助動詞
今は	副詞
3000円	**CURRENCY**
です	助動詞

図 4 日本語の例

図５は、日付、時間、通貨の同じ正規表現ルールを使用して、中国文と韓国文を解析した結果である。

中国語

是	
临终	
2004年1月15号	**DATE**
是	
临终	
15点45分	**TIME**
是	
临终	
3000元	**CURRENCY**

韓国語

지금은	
2004년 1월 15일	**DATE**
입니다	
지금은	
15시 45분	**TIME**
입니다	
지금은	
3000원	**CURRENCY**
입니다	

図 5 中国語と韓国語の例

ルール有りの場合に、日本文、中国文、韓国文で数詞が一つのトークンで解析され、またそのトークンの意味情報（日付、時間、通貨）が正しく取れていることが解る。韓国文の場合、日付は年月日が、時間は時分がそれぞれスペースで分かち書きされているが、それらが一つのトークンに解析されている。

郵便番号や割合などの他の数詞も日本語で正規表現ルールを定義して、一つのトークンに形態素解析できることを確認した。

4. まとめ

辞書に登録できない無数にバリエーションのある日付、時間、通貨の数詞に対して、形態素解析で正規表現ルールを用いてトークン処理が可能であることを実証した。その際、正規表現ルールに Unicode を用いる事により、日本語、中国語、韓国語で共通のルールを定義できることを実証した。

今後の課題として、日本語、中国語、韓国語の日付、時間、通貨などの同一の正規表現ルールを欧米語へ拡張し、この手法が特定の言語に依存することなく複数の言語に動的に適応できることを検証したい。

参考文献

[1] 田中穂積：自然言語処理-基礎と応用-、電子情報通信学会発行（コロナ社販売）、平成11年3月25日

[2] 浅原正幸、松本裕治　形態素解析とチャンキングの組み合わせによる日本語テキスト中の未知語出現箇所同定、情報処理研究会報告（自然言語処理研究会)、No. 2003-NL-154-8.

[3] ICU: the International Components for Unicode libraries,
http://oss.software.ibm.com/icu
http://oss.software.ibm.com/icu/userguide/regexp.html