ー 研究ノート ー

MeCab で利用可能なロシア語辞書について

上　原　順　一

Данная статья посвящена разработке электронного словаря русского языка для системы MeCab. Хотя MeCab является морфологическим анализатором, созданным для японского языка, однако для него можно создать «лингво-независимый» словарь. Автор статьи предлагает такой словарь и методы его использования. В статье даётся краткий обзор формата словаря и некоторые результаты аланиза текста на его основе.

(キーワード:ロシア語，辞書，テキスト分析)

はじめに

本稿のタイトルは「MeCabで利用可能なロシア語辞書について」である。MeCabとロシア語との両方に関わる研究者はそれほど多くないと考えられる。MeCabのことを知る研究者はロシア語の事情にそれほど精通しておらず，ロシア語の研究者はMeCabについてはあまり知らないのが現状であると思われる。本稿の目的はMeCabやその周辺的な知識をロシア語研究に利用する方法の可能性を提案することである。従って，MeCabを知る諸氏にとっては本稿の記述がやや冗長であると考えられるかもしれない。この点はご容赦願いたい。

まず，Rという統計解析ソフトがある[1)]。次にMeCabという形態素解析エンジンがある[2)]。MeCabは言語や辞書に依存しない設計になっているので，ロシア語の辞書を別途用意すればロシア語テキストの形態素解析も可能になると考えられる[3)]。この2つを効率的に利用すれば，ロシア語のテキストを分析して統計的な処理を行うことができる。研究者の立場からは，双方のソフトを行き来することなく，どちらかを主体にして分析ができれば便利である。幸いなことに，RからMeCabを操作するパッケージが作成されている。これがRMeCabである[4)]。

本稿では，MeCabで利用できるロシア語辞書（主として語形変化などの形態情報が含まれるデータ）について述べた後，実際にRMeCabでテキスト分析を行う様子を提示する[5)]。

ロシア語のテキスト分析

ロシア語のテキストを分析する方法は様々であるが，ロシア語が形態論の豊かな言語であり，

語形変化が多いことが分析を簡単に行う際の障害となることがある。たとえば，任意のテキストから говорить“話す”なる語の使用頻度を求めるという単純な作業を考える。もっとも原始的な方法は，エディタなどを用いてテキストから "говорить" という文字列を検索することである。しかし，この結果からは，この語の変化形である говорю などが脱落してしまう。このような意図しない結果を避けるためには，正規表現を用いることがひとつの方策である[6]。たとえば，"говор(ю|ишь)" といった正規表現を検索することで，たしかに，говорить の特定の語形を抽出することは可能になる[7]。しかし，使用頻度を求めたい語の変化形を正規表現で書く必要があり，このような語が複数あればなお煩雑な作業が要求されるので，あまり現実的な方法とは言えない。では，語形変化の情報は，どこでどのように提供されるのが良いのだろうか。

この方法には次の2つが考えられる。まず，ひとつはテキスト中に形態論的な情報などを埋め込んでしまう方法である。たとえば，テキストにある語形 говорю に対して，「この動詞の不定形は говорить であること，テキストの該当箇所に登場している形式は1人称単数形であること」などといった情報を記載しておけば，この情報をもとにこのテキスト中にある говорить の使用頻度を求めることができる。この情報は現実にはタグという特別な形式によって実現されることが多い。ロシア語最大のコーパスである Russian National Corpus もこの方法を採用している[8]。このコーパスでは，タグに若干の意味的情報も含まれている。テキスト中にタグを書き込んでおくこの方法の利点は，当然のことながら，利用者が語の形態的な情報をもたなくても良く，これをコーパスにあるテキストに委ねられることである。一方で，利用者自身が分析したいテキストを自由に設定しにくいという難点もある。

この難点を克服するためには，利用者の方で語形変化の情報を含む辞書を用意する方法がある。これがふたつめの方法である。これが用意できれば，たとえば CasualConc などのコンコーダンサー（コーパス分析ソフト）でロシア語のテキスト，しかも自分で分析したいテキストを自由に設定することができる[9]。このソフトでは統計的な分析は可能であるが，さらなる応用を試みる際には，R を用いる方が現実的である。前述のように MeCab に対応したロシア語辞書をもっていれば，RMeCab を使用することで R を離れることなく統計処理を行える。MeCab の辞書は比較的自由な設計になっていて，利用者が必要だと考える情報，たとえば，ある語がどの語からの派生語かなども書き加えることができる。

MeCab の辞書

MeCab で使用する辞書では変化形ごとにレコードを作る必要がある[10]。говорить ならば，

говорю, говорить, 1s
говорят, говорить, 3p

といったデータをあらかじめ用意する必要がある。上の例は説明用の単純な形式であるが，左から表層形（テキストに登場する実際の語形），不定形(原形)，文法情報（1sは1人称単数形，3pは3人称複数形）である。筆者は先行研究で似たような辞書を作成したことがあるので[5]，これをもとにMeCab用の辞書を用意した。

辞書の作成方法は多様であると想定される。なんらかのデータを変換することもあろうし，いちからMeCab用の辞書を作成することもあろう。前者の場合は，元になるデータも一様ではない。筆者がすでに作成していたデータをMeCab用の辞書に変換した際には，Rの関数を利用した。作成した辞書は次のようである[11)]。

```
"表層形","左文脈ID","右文脈ID","コスト","品詞","品詞細分類1","品詞細分類2","品詞細分類","活用形","活用型","原形","読み","発音","user"
"дело",0,0,10,"noun","N1","*","*","*","*","дело","*","*","user"
"дела",0,0,10,"noun","N2","*","*","*","*","дело","*","*","user"
"делу",0,0,10,"noun","N5","*","*","*","*","дело","*","*","user"
（中略）
"делах",0,0,10,"noun","N12","*","*","*","*","дело","*","*","user"
```

コストは小さいほど出現しやすいという数であり，ここでは仮に10を入れることにした。左文脈IDと右文脈IDは，それぞれ左から，ないし右から語を見たときの内部IDとのことである。これには-1をつけて，自動的にIDがふられるようにした。

この辞書は作成した段階ではCSVファイルである。これをバイナリ化して，ユーザ辞書として登録すると一連の作業が完了する。

上記のプロセスで作成された辞書が正しいかどうかは，ただちにMeCabを実行することで確認できる。ターミナル内で，

```
> mecab
> работает на почте
работает　動詞,V6,*,*,*,*,работать,*,*,user
на　prep,,*,*,*,*,на,*,*,user登録
почте　noun","N5","*","*","*","*","почта","*","*","user"
EOS
```

などとすれば良い[12)]。ただし，本稿執筆時点で語形の同音異義を区別するメカニズムは実現できなかった。上記の例では，почта が N5，すなわち単数与格と分析されているが，文脈的には

単数前置格と分析されるべきである。この難点はさらなる課題として残された。

なお，作成された辞書には，228401の形容詞語形，205505の名詞語形，1417417の動詞語形が含まれている[13]。

RMeCabでの利用

MeCabで利用できるロシア語辞書が整備された時点で，RMeCabを用いたテキストの分析がある程度可能になる[14]。もっとも単純な分析例として，テキストに提示された表層形から原形を導いて，語彙の頻度などを求めることが挙げられる。

まず，テキストより小さな文字列の分析例を述べる。RMeCabC()なる関数は，与えられた文字列を形態素分析するためにある。ロシア語の場合は，すでに分かち書きにより決定している語彙の分析に利用できる。第1の引数は文字列で，第2の引数は0の場合に表層形を返し，1の場合に原形を返す。

```
> res <- RMeCabC("работает на почте",0)
> res
[[1]]
"verb"
"работает"
[[2]]
"prep"
"на"
[[3]]
"noun"
"почте"
```

これで得られたそれぞれはベクトル形式になっている。Rでいうベクトル変数とは，複数のデータがまとめられたひとつの変数のことである。[[1]]は，"verb"と"работает"の2つが格納されたベクトルである。結果全体はリスト形式で返される。この結果はRのunlist()関数を利用することで，形態素別，上例では語彙別のデータを求めることできる。

```
> unlist(res)
"verb"          "prep"    "noun"
"работает"      "на"      "почте"
```

第2引数に1を指定した場合には，表層形が原形に変換される。

```
> res <- RMeCabC("работает на почте",1)
[[1]]
"verb"
"работать"
[[2]]
"prep"
"на"
[[3]]
"noun"
"почта"
> unlist(res)
"verb"        "prep"     "noun"
"работать"  "на"        “почта"
```

テキスト中の語彙頻度を求めるにはRMeCabFreq()なる関数が使用できる。次に挙げるのは，ガルシンの短編小説「信号」[15]を読み込んだところである。

```
> res <- RMeCabFreq("signal.txt")
file = signal.txt
length = 1473
```

読み込まれたテキスト名と分析された形態素数（ここでは語数）が表示される。結果は変数 res に格納されている。これを表示したものの一部を次に挙げる。

```
> res
Term      Info1     Info2     Freq
2          "баба"   “noun"   "N1"        2
3          "благородие"       “noun"     "N1"      7
4          "бог"     “noun"   "N1"        3
5          "бой"     “noun"   "N1"        1
6          "бок"     “noun"   "N1         "1
7          "болото"             “noun"     "N1       "1
```

```
8       "боль"  “noun"  "N1     "1
9       "брат"  “noun"  "N1     "7
10      "братец"        “noun"  "N1"    1
```

Termは形態素（語），Info1はMeCab辞書の品詞，Info2は同じく品詞細分類，Freqが頻度である。RMeCabFreq()で返される結果はデータフレームであり，これはRで加工することが可能である。たとえば，次のようにRの関数 subset() を用いて，頻度が1の語を抽出することができる。その一部を提示する。

```
> temp <- subset(res,Freq==1)
> temp
Term    Info1   Info2   Freq
5       "бой"   “noun"  "N1"    1
6       "бок"   “noun"  "N1     "1
7       "болото"        “noun"  "N1"    1
8       "боль"  “noun"  "N1"    1
10      "братец"        “noun"  "N1"    1
12      "вагон" “noun"  "N1"    1
13      "вал"   “noun"  "N1"    1
```

これとは別に，RMeCabの別の関数である collocate() を実行してみる。この関数では，特定の語と現れる別の語を求めることができる。下の例では，特定の語すなわち node に "новый» を指定した。これはMeCabの辞書によって変化形も検索の対象になる。span は特定の語の前後の語数である。ここでは1を指定した。

```
> res <- collocate("signal.txt",node = "новый",span=1)
file = signal.txt
length = 5
> res
        Term            Span    Total
1       "место"         1       5
2       ,               1       342
3       "Будка"         1       1
4       "на"            1       100
```

5	"новый"	2	2
6	[[MORPHEMS]]	5	1153
7	[[TOKENS]]	6	4078

「信号」の該当箇所を確認すると，上の1は "на новые места"，2は "Будка новая, теплая"，3は "Будка новая, теплая"，4は "на новые места" など，5はnode自体である。上の Total は各語彙の頻度である。検索語であるnodeもそれと共起する語も辞書の働きにより原形が返されている。

今後の展望と課題

本稿で試行したRとRMeCabにおけるMeCabの辞書使用は，語彙の頻度を求めるという単純な事例である。しかし，RMeCabにはこの他にもいくつもの関数が用意されており，いわゆるテキストマイニングに最適の環境が整っている。これを応用することにより，ロシア文学の分野では，たとえば作家の個人的文体を探ることも可能であると考えられる。

今後の第１の課題は辞書の整備である。MeCabで利用可能なロシア語辞書は，研究者が個人レベルで作成するにはやや荷が重い。しかし，特定のテキストをタグ付けし，それに特化した分析が多く行われている現状がある一方で，MeCab用の辞書を一度用意してしまえば，研究者がテキストを自由に選んで分析できるメリットは大きいと考えられる。

次の課題は，同音異義形の区別や形態素の正しい切り分けである。これはMeCabシステムをさらに理解することである程度可能になるかと思われる。これが実現できれば，テキストにおける統語構造の分析などがより簡単に行える可能性がある。

このような課題を念頭に，研究を進めることにしたい。

本稿は平成19年度-22年度の科学研究費補助金基盤研究（B）「LCTLを含む多言語平行マルチメディア資源の構築と構造化方式の研究」（課題番号:19300047，研究代表者:堀一成）の成果の一部である。

参考資料

・文献

［1］熊谷悦生，舟尾暢男，『「R」で学ぶデータマイニング（1.データ解析編）』，オーム社，2008.

［2］熊谷悦生，舟尾暢男，『「R」で学ぶデータマイニング（2.シミュレーション編）』，オーム社，2008.

［3］石田基広，『Rによるテキストマイニング入門』，森北出版，2008.

［4］金明哲，『テキストデータの統計科学入門』，岩波書店，2009.

［5］上原順一,「XMLを用いたロシア語の語形成電子教材の可能性について」『大阪大学世界言語研究センター論集』第1号，pp.63-73, 2009.

・Webサイト（2010年9月16日 20:29 現在）

1. http://www.r-project.org/（Rのサイト）

2. http://mecab.sourceforge.net/ (MeCab: Yet Another Part-of-Speech and Morphological Analyzer（MeCabのサイト）

3. http://rmecab.jp/wiki/index.php?RMeCab（石田基広氏によるRMeCabサイト）

---

[1] http://www.r-project.org/

[2] http://mecab.sourceforge.net/

[3] 形態素解析とは，主として文を語に分解するプロセスのことである。分解された語には辞書形や文法情報が付記されることもある。MeCabは日本語を形態素解析するための辞書とともに提供されている。

[4] http://rmecab.jp/wiki/index.php?RMeCab

[5] 筆者が利用したOSはMac OS X（10.6.4）である。しかし，本稿で紹介するソフト類は他のOS，たとえばWindowsなどでも動作する。

[6] 正規表現とは文字のパターンを表現する表記法である。

[7] この正規表現で，говорю と говоришь の両方を検索できる。

[8] http://www.ruscorpora.ru/

[9] http://sites.google.com/site/casualconcj/

[10] 辞書の詳細については http://mecab.sourceforge.net/dic.html で知ることができる。

[11] RMeCabの関数を実際に利用する際は品詞名を「名詞」などの日本語にするほうが便利である。

[12] > の右側が実際に打ち込む内容である。

[13] 見出し語に相当する語の概数は，形容詞が5700語，名詞が12800語，動詞が7700語である。なお，これらの語数と語形数は辞書作成直後に計算した結果である。

[14] RMeCabのインストールやその関数については，http://rmecab.jp/wiki/index.php?RMeCab で知ることができる。

[15] http://az.lib.ru/g/garshin_w_m/text_0190.shtml