

## 統計的テキスト解析(2)

### ～ データのクリーニングと関連ツール ～

同志社大学文化情報学部教授

金 明哲 (Jin Mingzhe)

■中国生まれ。総合研究大学院大学数物研究科統計科学専攻博士後期課程修了。博士(学術)。1995年札幌学院大学社会情報学部、助教授、教授を経て、2005年4月より現職。E-mail: mjn@mail.doshisha.ac.jp

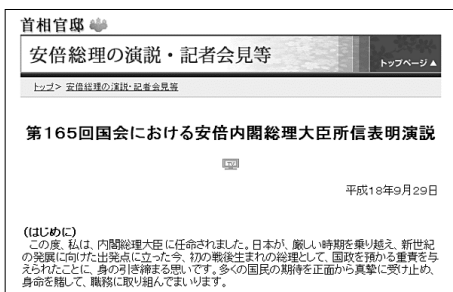


#### 1. データの作成とクリーニング

コンピュータを用いて、テキストを計量分析するためには、分析する対象を電子化し、データの形式を揃えたり、不必要なものを削除したりする作業が必要である。

例えば、安倍元総理と福田総理の所信表明演説文について比較分析を行う場合は、まずその演説文を電子化しなければならない。幸い電子化されているものがインターネット上に公開されているので、コピーして用いることが可能である。図1に安倍元総理所信表明演説のウェブページの画面コピーを示す。

図1 ウェブページ画面



出典) <http://www.kantei.go.jp/jp/abespeech/2006/09/29syosin.html>

これをコピーしてテキストエディタに貼り付けると、図2のようになる。テキストの中の日付「平成18年9月29日」に関心がなければ削除し、図3のように空行も削除した方がよい。このような作業をデータのクリーニングという。

図2 テキストの編集画面1

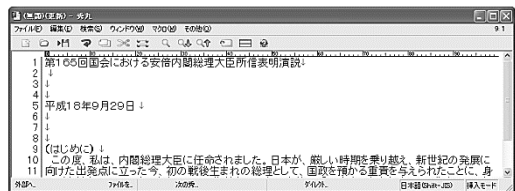
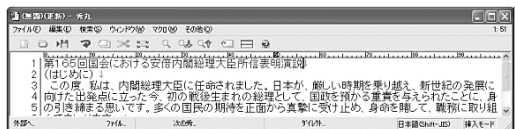


図3 テキストの編集画面2



#### 2. テキストのクリーニングと正規表現

テキストのクリーニングを行うためには、テキストエディタを用いる必要がある。テキストエディタとは、テキストを編集するためのソフト

トである。マイクロソフト社のWindows系列の場合、「メモ帳」、「ワードパット」がフリーインストールされている。これらは一種のテキストエディタである。テキストデータのクリーニングの作業に広く用いられているのは、日本では「秀丸」である。秀丸は有料であるが、期間限定で、無料で試用することができる。

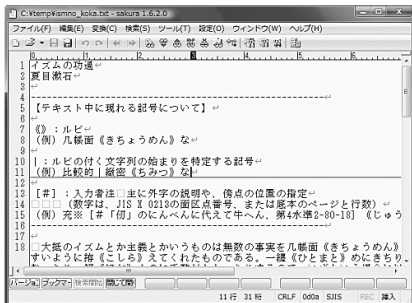
フリーのテキストエディタは多数ある。比較的に評判が良いのは「サクラエディタ」である。サクラエディタは、1998年頃から公開され、2000年頃からオープンソースで開発が行われるようになってきている。これは、次のサイトのリンクからダウンロードできる。

<http://sakura-editor.sourceforge.net/>

安倍元総理や福田総理の所信表明演説のようなデータは、クリーニングにあまり手間がかからないが、データによっては非常に手間がかかるものもある。

ここでは、著作権切れの小説などをネット上で公開している「青空文庫」(<http://www.aozora.gr.jp/>) からテキストファイルをダウンロードして用いるケースを例とする。まず、青空文庫から夏目漱石の「イズムの功過」のzipファイルをダウンロード・解凍し、Cドライブの中のtempというフォルダの中にismno\_koka.txtとして保存する。次に、これをテキストエディタで開く。その画面コピーを図4に示す。

図4 「イズムの功過」のテキスト画面



本文の上に破線で囲まれた部分は、テキストの本文ではなく、本文を電子化する際に付け加えられたルビや記号に関する説明文である。本文を計量分析するときは、本文以外の説明、付け加えられたルビや記号などを削除することが必要である。ルビの削除は単純であるが、外字(現時点の文字コードにない文字)をどう処理するかは難しい問題である。

ルビの削除はマウスやキーボードの操作で一つひとつ行うと、非常に労力がかかる。比較的に効率良く作業を行う方法は、テキストエディタにおける正規表現を用いることである。正規表現とは、文字列の検索・置換を行うため、文字列のパターンを記号の組み合わせで表現する表記法である。

正規表現には、アルファベットとメタキャラクタと呼ばれる特別な意味を持っている記号が用いられている。サクラエディタの正規表現に用いられている主なメタキャラクタとエスケープシーケンスを表1に示す。

表1 主なメタキャラクタとエスケープシーケンス

	名称	機能
.	ピリオド	任意の1文字
*	アスタリスク	0個以上の繰り返し
+	プラス	1個以上の繰り返し
?	疑問符	0個または1個
^	カレット	カレットの右の表記を否定、あるいは行頭
\$	ドル記号	行末
[]	ブラケット	キャラクタクラス
()	パーレン	グルーピング
	パイプ	パターンの論理和
¥	円記号	エスケープキャラクタ
¥t		タブ
¥n		改行
¥f		改ページ

エスケープシーケンス (Escape Sequence) とは、コンピュータシステムにおいて、通常の文字コードでは表せない特殊な機能を表すために記号とアルファベットを組み合わせたものであり、例えば、タブ (¥t)、改行 (¥n)、改ページ (¥f) などがある。



な機械語に変換して実行するプログラミング言語である。C, C++, FORTRAN, Javaなどはコンパイラ型言語である。コンパイラ型言語のプログラムは、まずプログラムをコンパイルすることが必要である。

上記のどの言語を用いてもテキストの処理を行うことはできるが、テキスト処理にはPerl（パールと呼ぶ）言語が向いている。

Perlはラリー・ウォール（Larry Wall）により開発され、1980年代後半に公開されたフリーのプログラミング言語である。当初は、「高価な真珠」にちなんで、真珠を意味する「Pearl」と名付けられたが、より短く3～4文字に命名するため「Perl」としたとの説がある。元々意味はないが、あとからいくつかの意味付けが考えられた。その一つの説として、Practical Extraction and Report Language（実用的なデータ取得とレポート作成言語）の頭文字を取ったというものがある。

Perlはテキストの処理、レポート作成に向けたインタプリタ型言語である。近年では、Web関係のCGIの開発にも多く使われている。Perlは、開発・公開されて20年が過ぎているので、Perlに関する書物やインターネット上の読み物も比較的が多い。Perlの詳細を説明するのは、本稿の狙いではない。しかし、テキストの処理にはPerl言語を知っておくと便利であり、またテキスト処理ツールには、しばしばPerl言語が関係しているので、その用法に関して簡潔に説明しておく。

Windows版のPerlとしては、ActiveSTATE社が開発・公開しているフリーのActivePerlがある。最新バージョンは5.10であり、日本語を含む2バイトの言語の処理が可能になっている。次のサイトからダウンロードすることができる。

<http://www.activestate.com/Products/activeperl/>

ダウンロードサイトには、MSI, AS packa, Symbolsの3種類が置かれている。インストールのしやすさなどから、初心者にはMSIをすすめる。

インストールは、ダウンロードしたファイルActivePerl-5.\*\*\*.msiをマウスの左ボタンでダブルクリックし、インストールダイアログボックスの質問に応じてボタンをクリックする単純作業である。

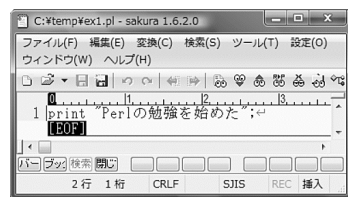
Perlプログラムは、基本的にはテキストエディタを用いてプログラムを書き、そのプログラムを拡張子plで保存し、コマンドプロンプトでプログラムを実行する。

コマンドプロンプトとは、コマンド（Command）で、コンピュータを操作するプロンプト（Prompt）である。コンピュータの「スタート」⇒「プログラム」⇒「アクセサリ」⇒「コマンドプロンプト」をクリックすると、コマンドプロンプトが開く。

簡単なPerlのプログラムとその実行例を次に示す。まず、テキストエディタで図6のように次の1行のプログラムを書く。

```
print "Perlの勉強を始めた";
```

図6 Perlプログラムの例



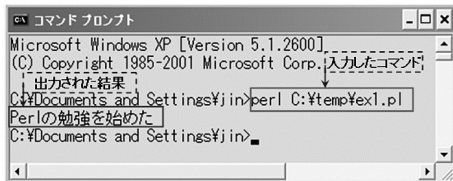
これに、ex1.plというファイル名を付け、C:\tempに保存したとする。このプログラムは「Perlの勉強を始めた」を画面に出力するプログラムである。

Perlプログラムの実行は、コマンドプロンプト上で、図7のように

```
perl C:%temp%ex1.pl
```

を入力し、[Enter]キーを押すと、プログラムが実行し、結果がコマンドプロンプトに返される。プログラムにミスがあった場合は、エラーメッセージを返す。

図7 Perlプログラムの実行の例1



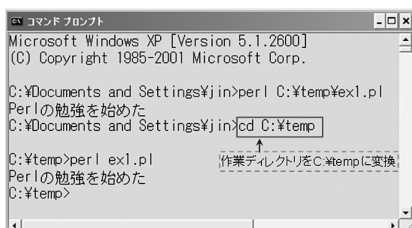
コマンドラインの中のperlは、Perl言語を実行するための宣言であり、C:%tempはプログラムを保存したドライブとフォルダ（ディレクトリとも呼ぶ）の名前である。

この1行のコマンドラインは、Cドライブの中のフォルダtempの中に保存しているPerlプログラムex1.plを実行することを意味する。上記のC:%tempのような、ファイルが置かれているドライブ、フォルダを示す文字列をパス（Path）と呼ぶ。パスは、ドライブ、フォルダ、ファイルの間を記号%で区切る。

コマンドプロンプトがフォルダtempにアクセスしている場合は、パスを指定する必要がない。次に、コマンドプロンプトの作業フォルダ（ディレクトリ）を入れ替えて、プログラムを実行する操作画面を図8に示す。

Perl言語では、正規表現が使用可能である。

図8 Perlプログラムの実行の例2



ただし、正規表現の表記法は、用いるテキストエディタと言語によって多少異なるものもある。

第2節で用いたテキストismno\_koka.txtを用いて、《》に囲まれているルビと《》を同時に削除する簡単なプログラムを次に示す。

```
while(<>){  
s/《[^]》+》//g;  
print "$_";  
}
```

このプログラムをex2.plとして保存し、コマンドプロンプト上で、コマンドライン

```
perl ex2.pl ismno_koka.txt >kekka.txt
```

を実行すると、テキストismno\_koka.txtの中の《》に囲まれているルビおよび括弧《》が同時に削除され、ファイルkekka.txtに保存される。ただし、上記のコマンドラインの表記は、処理するプログラムex2.plと処理対象のテキストismno\_koka.txtが同じフォルダの中に保存されており、コマンドプロンプトがそのフォルダにアクセスしている場合に限り有効である。そうではない場合は、プログラムと処理対象が置かれているパスを指定しなければならない。

上記のプログラムex2.plの中の

```
while(<>){処理内容}
```

は、コマンドラインで指定したテキストが空でなければ処理を行う。ただし、テキストは行単位で\$\_に格納される。

プログラムの中の

```
s/《[^]》+》//g;
```

は、ルビおよび《》を削除するために置換を行うコマンドであり、Perl言語では、次の構文となっている。

```
s/置換前/置換後/g;
```

ルビを削除する表記には、置換後の文字列が指定されていないため、文の中の正規表現「`[^ ]+`」を見つけると、すべて削除することになる。プログラムの中の

```
print "$_";
```

は、処理した文\$\_をプリントする。出力ファイルを指定しないと画面上に返すが、出力ファイルを指定すると画面には返さず、ファイルに書き込む。

#### 4. 大量のテキストのバッチ処理

前節のように、コマンドプロンプト上でコマンドラインを実行すると、1回のコマンドライン操作で1つのファイルしか処理できないので効率が悪い。何十、何百の大量のテキストファイルを処理するときには、バッチファイルを用いると便利である。

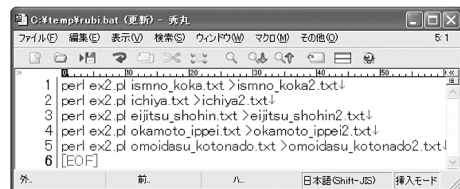
バッチファイルとは、基本的にコマンドプロンプト上の1命令を1行に、上から順に1行ずつ複数の実行命令を書き、拡張子batで保存したファイルである。バッチファイルを実行すると、第1行から命令を実行し、続いて第2行、第3行…のように、終わりの行まで命令を実行する。コマンドプロンプトで命令を実行する際には、常に監視しながら、1つの命令が終わったら、次の命令を与えなければならないが、バッチファイルを用いると特に監視する必要がない。バッチファイルの作成は非常に簡単である。例を示すために、青空文庫から表2のテキストをダウンロード・解凍し、Cドライブの中のフォルダtempに保存したとする。

この5つのファイルについて、プログラムex2.plでルビを削除するコマンドラインを図9のように書き、プログラムとファイルが置かれているフォルダに、拡張子batで保存する。

表2 作品とファイルの名前

作品の名前	ファイルの名前
イズムの功過	ismno_koka.txt
一夜	ichiya.txt
永日小品	eijitsu_shohin.txt
岡本一平著並画『探訪画趣』序	okamoto_ippei.txt
思い出す事など	omoidasu_kotonado.txt

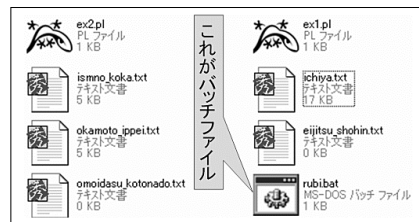
図9 バッチファイル作成画面



保存したフォルダを開くと、図10のようにバッチファイルが作成されていることが確認できる。

バッチファイルをマウスの左ボタンでダブルクリックすると、バッチファイルが実行される。上記のバッチファイルの場合は、結果が同じフォルダに作成される。

図10 バッチファイルのアイコン



正規表現、コマンドプロンプト（あるいはMS-DOS）、Perlに関する本は多く出版されている。初心者向けのテキスト処理に関する総合的な参考文献としては、中尾他（2002）、赤瀬川・中尾（2004）がある。

\*参考文献

- [1] 中尾 浩・宮川進悟・赤瀬川史朗(2002)：コーパス言語学の技法<1> テキスト処理入門：夏目書房。
- [2] 赤瀬川史朗・中尾 浩(2004)：コーパス言語学の技法<2> 言語データの収集とコーパスの構築(単行本)：夏目書房。