第 13 回：スクリプト言語の利用 (Perl)

1. エディタによるテキスト加工からより複雑なテキスト処理へ

エディタの検索・置換機能を使ったテキスト加工作業には，限界がある：

- 手間がかかる (何回も検索・置換を繰り返す必要がある)
- 正規表現の処理に時間がかかる (ファイルの改行記号(¥n)を追加･削除するといった単純な作業でも，夏目漱石『こころ』の全文テキスト kokoro.txt のような大きなファイルを扱う場合，EmEditor の正規表現では非常に遅い。←第 6 回授業参照)
- 複雑な加工が不得意 (たとえば，単語の出現回数を数える，というような単純な作業でも，EmEditor 単体では難しい)

より柔軟で複雑なテキスト処理をおこなうには, どうしてもテキスト処理用のスクリプト言語による自動処理をおこなう必要がある。スクリプト言語は sed (せど), awk (おーく), perl (ぱーる) が有名で，いずれも正規表現をサポートする。

2. スクリプト言語 Perl を使ったテキスト処理の例

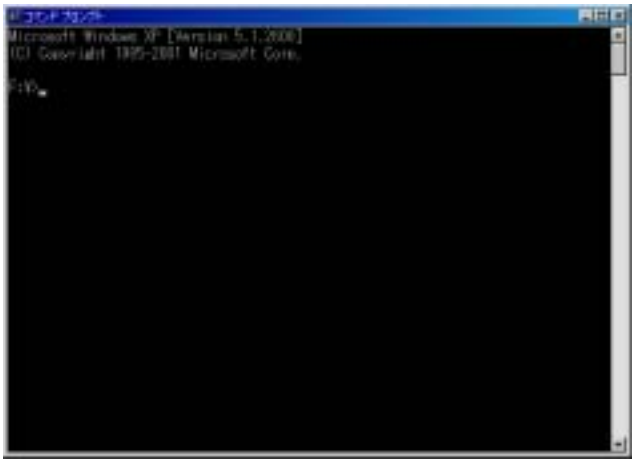
- Perl (ぱーる, *Practical Extract and Report Language*) は正規表現を使ってテキストを加工できる強力なスクリプト言語で, テキスト処理でもっともよく用いられる (もちろん，テキスト処理だけでなく，統計処理やネットワークシステムなどにも広く応用が可能)。
- 大学の PC にはバージョン 5.8.4 が入っており, Unicode (UTF-8) をはじめさまざまなエンコードのテキストを扱うことができる。

Windows XP には,「コマンドプロンプト」と呼ばれるコマンド入力用画面が標準で使えるようになっている。[スタート]→[プログラム]→[アクセサリ]→[コマンドプロンプト]で開く。以下のようにして，スクリプトを実行する準備をしよう。

C:¥>**F:** *[Enter]*
F:¥>**cd kenkyuu2005** *[Enter]*
F:¥kenkyuu2005>**dir** *[Enter]*
F:¥kenkyuu2005>**cd No13** *[Enter]*
F:¥kenkyuu2005¥No13>**dir** *[Enter]*

コマンドの例：

dir: 現在のフォルダ内のファイルの一覧を表示
cd: フォルダを移動



(コマンドプロンプトのワーキングディレクトリを file_server の Home にある kenkyuu2005 フォルダの No13 フォルダに変更し，ファイル一覧および現在のディレクトリ名を表示する。)

- perl のバージョン表示：

> **perl –v** *[Enter]* **または perl -V** *[Enter]*

スクリプトによるテキスト処理には，グラフィカルなインターフェース (GUI, graphical user interface) ではなく，以下で紹介する Windows の「コマンドプロンプト」のような，文字テキストで処理を指定する旧来のインターフェース

(CUI, character-based user interface) を使う。なお，以下で実習する形態素解析システム ChaSen [1] のように，Unix など Windows 以外の OS での利用を想定したテキスト処理ツールは，基本的に CUI での利用を前提としている (ChaSen の Windows バージョンには，コマンドプロンプトを利用するもののほかに GUI をもつものもある)。

2.1. テキスト処理の実例：形態素解析システム chasen の利用

- 『こころ』のテキスト NoRuby.txt を日本語形態素解析ツール ChaSen で形態素ごとに分割し，結果を NoRuby_cha.txt というファイル名で保存しよう。
- [スタート] → [プログラム] → [アクセサリ] → [コマンドプロンプト] を開き，以下のコマンド (網掛け部分) を**すべて半角で**入力していく。
- NoRuby_cha.txt を EmEditor で開いて出力結果を確認しよう。

```
>F: [Enter]
>cd  kenkyuu2005 [Enter]
>dir [Enter]
>cd  No13 [Enter]
>dir [Enter]
>chasen -F "%m¥t%P-¥n" NoRuby.txt > NoRuby_cha.txt [Enter][2]
>dir [Enter]…
>exit [Enter]
```

2.2. Perl によるテキスト処理の実例 (1)

- 2.1. で作成した,『こころ』のテキストを ChaSen にかけたファイル NoRuby_cha.txt を加工する **seikei.pl**，単語の頻度情報を調べる **count.pl**，および単語の出現場所 (行数) を調べるスクリプト **reference.pl** で，Perl の威力を調べてみよう。

```
>F: [Enter]
>cd  kenkyuu2005 [Enter]
>dir [Enter]
>cd  No13 [Enter]
>dir [Enter]
>perl seikei.pl < NoRuby_cha.txt > keitai.txt [Enter]
>perl count.pl < keitai.txt > keitai_count.txt [Enter]
>perl reference.pl < keitai.txt > keitai_refs.txt [Enter]
>dir [Enter]
>exit [Enter]
```

MS-DOS コマンドの解説：

<: 入力ファイル指定
>: 出力ファイル指定

---

[1] ChaSen (茶筌) は奈良先端科学技術大学松本研究室が開発しているフリーの日本語形態素解析システム。URL: http://chasen.aist-nara.ac.jp/hiki/ChaSen/

[2] オリジナル出力には読み (カタカナ) や見出し語，活用形の詳細などが含まれるが，NoRuby_cha.txt では，出現形と品詞情報のみをタブで区切って出力している。このようなカスタマイズした出力結果は， -F オプションの後に，具体的な出力フォーマットを指定する (詳細は ChaSen のマニュアルを参照するとよい)。

担当：千葉庄寿 (CHIBA Shoju) e-mail: schiba@reitaku-u.ac.jp

2.3. Perl **によるテキスト処理の実例** (2)

- 同様の作業をフィンランドの叙事詩『カレヴァラ』のテキスト kalevala.txt でもおこなってみよう。同ファイルは ISO-8859-1 でエンコードされている。エンコードとファイルの構造を kalevala.txt に合わせた **countKalevala.pl** と **referenceKalevala.pl** を使い，単語の頻度情報と出現場所 (もとのファイルと行数) を調べよう。

```
>perl countKalevala.pl < kalevala.txt > kelevala_count.txt [Enter]
>perl referenceKalevala.pl < kalevala.txt > kalevala_refs.txt [Enter]
```

3. Perl (**「ぱーる」** = Practical Extraction and Report Language)

- 「プログラミング言語」としての Perl
  - データの処理方法をコードに書いて即座に実行できる「スクリプト言語」の性格をもつ (< sed, awk)。C, C++, Java, Visual Basic など，他の殆どの言語は一旦プログラムをコンパイルしないと使えない。
  - プログラム言語として設計されているので，簡単なテキスト処理ツールだけでなく，本格的なネットワークアプリケーション等も (例えばメーリングリストや Web ブラウザもどきまで！) 作成可能。また，WWW の普及によって，CGI (common gate interface) 作成用の言語としても人気がある。
  - 「やり方は 1 つ以上ある」"There's more than one way to do it!" (Larry Wall)
  - 拡張性が非常に高い。既存の拡張ツールを組み合わせたり，書いたコードを簡単に再利用することができる。
  - Windows はもちろん，Macintosh, Unix など，多くの OS 上で使える。
  - GUI 作成環境が多少弱い。Perl/Tk 等の拡張ツールキットが使えるが，殆どの作業はコマンドラインからおこなうのが普通。従って，Windows の「コマンドプロンプト」の利用に慣れることが必要。
- 「フリーウエア」としての Perl
  - 大学 PC には，ActiveState が Windows 用に作成した Perl (ActivePerl, フリーウエア) がインストールされている。入手先： `http://www.activestate.com/`
- 「テキスト処理ツール」としての Perl
  - 強力な正規表現が利用でき，テキスト処理に強い。
    - ✧ パターンマッチング (検索)
    - ✧ データ変換 (置換)
  - Unicode の多言語テキストを利用可能 (Perl バージョン 5.8 以降で本格対応)。

- **【注意】**Perl のバージョンについて
  - Perl は 5.6 に入り，部分的に Unicode をサポートした。しかし，多言語テキストの処理機能は完全ではなく，Perl 5.6 で日本語テキストを扱うには Jcode.pm と呼ばれる拡張ツール (モジュール) が必要だった。しかし，Perl 5.8 で Perl は完全に Unicode に対応し，UTF-8 をはじめ日本語を含め全 114 種類ものエンコードを標準で扱えるようになった。
  - しかし，Unicode サポートの向上にともない，Perl が 5.6 か 5.8 かによってスクリプトの書き方が変わってしまう，という面倒な事態が生じている。他人の作成した Perl スクリプトを使う場合には, そのスクリプトが大学 PC の Perl 5.8 に対応しているかを確認し，必要なら修正する必要がある (スクリプトの変更は多少面倒である)。
  - Perl 5.6 以前のバージョンには，日本語テキストの処理専用の Perl (*JPerl*) が存在した (鈴木紀夫さん作, http://homepage2.nifty.com/kipp/perl/jperl/)。JPerl は Perl 5.6 および 5.8 (大学 PC 上) と日本語処理の方法が全く異なるので，スクリプトによっては一方でしか動作しないので注意。

4. Perl の基本

- perl スクリプトの文法チェック： > **perl –cw スクリプト名**
- perl スクリプトの起動：基本は **perl (オプション) スクリプト名**
  > perl (オプション) スクリプト名 < 入力ファイル名 > 出力ファイル名
  → 結果をファイルに出力
  > perl (オプション) スクリプト名 < 入力ファイル名 | more
  → 結果をコマンドプロンプトに出力 (スペースでページ送り，q で終了)
- スクリプトはコマンドごとにセミコロン ; で区切る
- 変数の表記：スカラーは$ (例 $scalar), 配列は@ (例 @array), 連想配列は% (例 %hash)
- 変数への値の代入は等号 = を使う：

```
$scalar = "日本語\n";
@array = ("日本語", "英語", "韓国語");
%hash = ("英語" => "English", "韓国語" => "Korean");
%hash = ("英語", "English", "韓国語", "Korean");
```

  連想配列の書き方は 2 種類あり，どちらも全く同じ意味。
- コメントは # (半角) の後に書く
- マッチは //, 置換は s///, 文字単位の置き換えは tr///

```
$scalar = s/japan/Japan/;
$scalar = tr/a-z/A-Z/;
```

- ループ (while, for, foreach)：「1 行ずつ入力し，入力のあるうちはずっと内容を繰り返せ」

```
while (<>) {
        # 繰り返し内容をここに
}
```

- 条件分岐：「もしAなら「あ」に，Bなら「い」に，他は「う」に行け」

```
if (条件A) {
        # あ
} elsif (条件B) {
        # い
} else {
        # う
}
```

- 多言語対応：以下は日本語 Shift JIS 用のスクリプトの記述例。

```
use encoding "shiftjis";
binmode STDERR, ":encoding(shiftjis) ";
```

  スクリプト自体をふくめ，全て Unicode (UTF-8)にする場合には **use utf8;** とする。

5. Perl の実例 (以下のスクリプトは Shift JIS に対応)

以下のスクリプトは No13 に入っています。使ってみてください。Shift JIS でエンコードされたテキストに対応しています。

5.1. 数える

- 行番号を付加： > **perl addlineno.pl < 入力テキスト > 出力テキスト**
- 文字数を数える (スペースで区切られているもの)：
  > **perl charcount.pl < 入力テキスト > 出力テキスト**
- 単語数を数える (スペースで区切られているもの)：
  > **perl wordcount.pl < 入力テキスト > 出力テキスト**

担当：千葉庄寿 (CHIBA Shoju) e-mail: schiba@reitaku-u.ac.jp

5.2. マッチ

- 簡単な grep 検索 (行番号つき，正規表現利用可)：
  > perl simplegrep.pl “検索文字列” < 入力テキスト > 出力テキスト

5.3. 置換

- 正規表現を使った grep 検索の結果マッチした箇所の前後にタブを挿入：
  > perl grep-n-tab.pl “検索文字列” < 入力テキスト > 出力テキスト
- 正規表現を使った grep 検索の結果マッチした行を kwic 的に整形：
  > perl grep-n-kwic.pl “検索文字列” < 入力テキスト > 出力テキスト

5.4. 並べ替え

- 正規表現を使った grep 検索の結果マッチした箇所の前後にタブを挿入し，マッチした箇所の後でソート：
  > perl grep-n-sort1.pl “検索文字列” < 入力テキスト > 出力テキスト
- 正規表現を使った grep 検索の結果マッチした箇所の前後にタブを挿入し，マッチした箇所の前でソート：
  > perl grep-n-sort2.pl “検索文字列” < 入力テキスト > 出力テキスト

6. 参考文献

6.1. Perl について

- 中島靖 (1998) 『Perl 使いへの旅立ち：日本語 TEXT 加工入門ハンドブック改訂新版』情報管理. (Windows で Perl を利用するための基礎作りから丁寧に解説。現在絶版。)
- 武藤健志／トップスタジオ (2004) 『独習 Perl』第 2 版. 翔泳社.
- 平田豊 (2004) 『Perl トレーニングブック』ソーテック社.
- 目黒編集室 (2004) 『これだけで身につく Perl 入門 例題 80』 日経 BP 社.
  ※ Perl の入門書はほかにもたくさんある。
- Wall, Larry ほか (2002) 『プログラミング Perl』第 3 版 (全 2 巻.) オライリー・ジャパン. (Perl 開発者たちによる Perl の原典。「ラクダ本」と呼ばれ，本格的に勉強する人は必須。)
- Hammond, Michael (2003) *Programming for Linguists: Perl for Language Researchers*. Oxford: Blackwell.
- 佐野洋 (2003) 『Windows PC による日本語研究法』共立出版. (Perl を使った日本語研究用の GUI ツールを紹介。古いバージョンの Perl でしか動作しないのが残念。)
- 中尾浩，赤瀬川史朗，宮川進悟 (2002) 『コーパス言語学の技法 I: テキスト処理入門』
  同 (2004) 『コーパス言語学の技法 II 言語データの収集とコーパスの構築』夏目書房. (ヨーロッパ言語についての解説が中心だが，Perl による処理を含む Windows 上でのテキスト処理に関する情報が I ではコンパクトに，II ではより実践的に解説されている。)

6.2. Perl 以外のスクリプト言語について

- 美吉明浩 (1998) 『Grep, Sed, Awk Manual & Reference』秀和システム.
- Dougherty, Dale ほか (1997) 『sed & awk プログラミング』改訂版. オライリー・ジャパン.
- Robbins , Arnold (2000) 『sed & awk デスクトップリファレンス』オライリー・ジャパン.
- Aho, Alfred V (2001) 『プログラミング言語 AWK』シイエム・シイ.
- 上田博人 (1998) 『パソコンによる外国語研究 (II) 文字データの処理』くろしお出版.
- Barnbrook, Geoff (1996) *Language and Computers: A Practical Introduction to the Computer Analysis of Language*. Edinburgh: Edinburgh University Press.

6.3. 「コマンドプロンプト」について

- 米田聡 (2002) 『Windows ユーザのための DOS/コマンドプロンプト入門』ソフトバンク.
  ※ Windows のコマンドプロンプトについての参考書はほかにも数多くある。