統計的テキスト解析（4）
～ 統計モデルと集計ツール ～

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：mjin@mail.doshisha.ac.jp

1. 統計モデル

テキストについて、何らかの処理や統計分析を行うときには、まず単位を決めることが必要である。テキストについて機械的にスペルチェックを行う際には、文字と単語を単位とし、文体分析や意味論的に統計分析を行うときは、単語、文節、文などを単位とする。文字、音素、単語、品詞、文節、文などを単位とした場合、単位ごとに記号s_iで表すと、テキストは記号列$s_1 s_2 \cdots s_{i-1} s_i s_{i+1} \cdots s_n$と見なすことができる。

記号列を統計分析する基本は、それぞれの記号がテキストの中に現れる度数（頻度）である。さらに拡張した統計モデルは、２つの記号、３つの記号、…、n個の記号が隣接して出現する共起度数である。１つの記号、隣接する２つの記号、３つの記号、…、n個の記号の度数を統計分析する方法をn-gram（エヌグラム）モデルと呼ぶ。

n-gramのnは、統計分析を行うために切り取った記号列の長さであり、nが１のときunigram（ユニグラム）、nが２のときbigram（バイグラム）、nが３のときtrigram（トライグラム）、nが４のときにはfour-gram（フォーグラム）のように呼ぶ。

大量のテキストから得られたn-gramの出現度数に関する統計データは、そのテキストの解析および自然言語処理に広く用いられている。例えば、英文ではqの後ろにはほとんどの場合、uが続くことから、qの後に続く文字が識別できない場合は、uと断定しても間違う確率は非常に小さい。このような情報は、qのbigramの統計データから得られる。

n-gramの例として、例文「きしゃのきしゃがきしゃできしゃする。」のunigram、bigram、trigramの集計結果を表１に示す。

表１の中の「相対度数」は、その項目の度数を合計の値で割ったものである。相対度数に100を乗じた百分率を用いてもよい。

表１で分かるように、文字を単位としたn-gramの場合は、集計された結果には言語学的に意味を持たないものもある。

表1　例文のn-gram（n＝1, 2, 3）

(a) unigram

1文字	度数	相対度数
き	4	0.222
し	4	0.222
ゃ	4	0.222
。	1	0.056
が	1	0.056
す	1	0.056
で	1	0.056
の	1	0.056
る	1	0.056
合計	18	1

(b) bigram

2文字	度数	相対度数
きし	4	0.235
しゃ	4	0.235
がき	1	0.059
する	1	0.059
でき	1	0.059
のき	1	0.059
ゃが	1	0.059
ゃす	1	0.059
ゃで	1	0.059
ゃの	1	0.059
る。	1	0.059
合計	17	1

(c) trigram

3文字	度数	相対度数
きしゃ	4	0.250
がきし	1	0.062
しゃが	1	0.062
しゃす	1	0.062
しゃで	1	0.062
しゃの	1	0.062
する。	1	0.062
できし	1	0.062
のきし	1	0.062
ゃがき	1	0.062
ゃする	1	0.062
ゃでき	1	0.062
ゃのき	1	0.062
合計	16	1

n-gramは文字だけではなく、単語、品詞、文節などを単位としてもよい。また、分析対象となる項目以外を無視して、隣接している項目のn-gramを用いることも可能である。例えば、テキストの中の助詞について統計分析を行うときには、テキストの中に現れている助詞以外を取り除き、助詞のみのn-gramの統計データを用いると、助詞の組み合わせの特徴を分析することが可能である。

n-gramモデルは欠点はあるものの、データの集計に便利であることから、テキスト処理を含む自然言語処理に広く用いられている。

2. データ集計のツール

テキストから表1のような文字単位のn-gramを抽出するプログラムが、インターネット上にいくつか公開されている。しかし、複数のテキストを同時に、ファイル単位にデータの集計処理を行う際には、使い勝手がよくない。そこで、本稿では、筆者の個人用として作成した簡易ソフト（MLTP：MultiLingual Text Processor）について簡潔に紹介する。

MLTPは、複数のテキストを同時に処理することを前提としており、日本語、中国語、韓国語、英語を扱うことができる。MLTPは、Java言語で書かれ、Windowsで検証が行われている。Java言語がインストールされているマシンであれば、ダウンロードして直接使用できる。

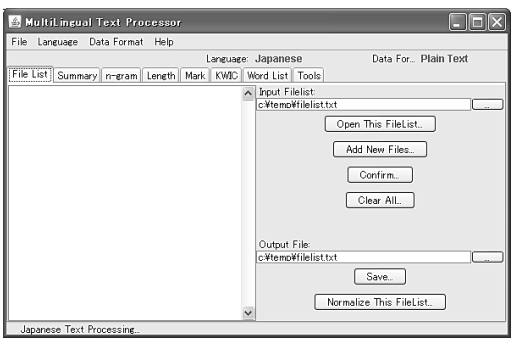
(1) ダウンロードと起動・終了

MLTPは、現時点では、次の筆者のサイトからダウンロードできる。

http://mjin.doshisha.ac.jp/MLTP/index.html

ダウンロードしたファイルmltp.zipには、コンパイル済みのプログラムmltp.jarがある。mltp.jarを左ダブルクリックすると、図1のような画面が開く。あるいはmltp.jarを選択し、マウスの右ボタンを押し、「プログラムから開く」⇒「Java(TM)Platform SE binary」をクリックしてもよい。

図1　MLTPの起動画面



また、コマンドプロンプトから起動することも可能である。テキストの量が多いときには、コマンドプロンプト上で使用するメモリを指定する必要がある。mltp.jarが置かれているフォルダにアクセスし、次のようにコマンドを実行する。

```
>java -Xmx500m -jar mltp.jar
```

コマンドの中の「−Xmx500m」は、使用するメモリを500MBに指定するオプションである。ファイルの量が多くない場合は「−Xmx500m」を省略してもよい。デフォルトは256MBになっている。

MLTPの終了は、メニューの「File」⇒「Exit」をクリックする方法と画面の右上の×印（閉じるボタン）をクリックする方法がある。

⑵ ファイルの読み込み

MLTPでは、一般のテキストファイル（Plain Text）と品詞などのタグが付いているファイル（Tagged Text）について、データの集計を行う。デフォルトはPlain Textになっている。ファイルの形式はtxt形式を前提としている。

ファイルの操作を行うためには、まず操作画面の「File List」タブをクリックする。次に、ボタン[Add New Files...]をクリックし、ファイルが置かれているドライブとフォルダを指定し、解析するファイルを選択する（図２）。選択が終わったら、[開く]ボタンを押すと、選択されたファイルが読み込まれる（図３）。このように、他のフォルダのファイルを読み込み、追加することが可能である。

ファイルの読み込み作業が終わったら、操作画面の[Confirm...]ボタンを押し、確認作業を行うことが必要である。

図２　ファイルを読み込む画面

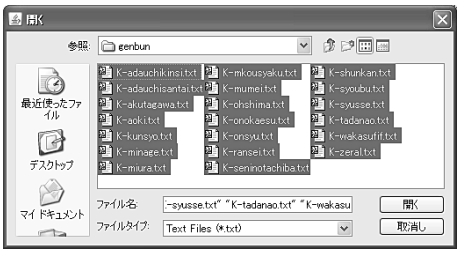
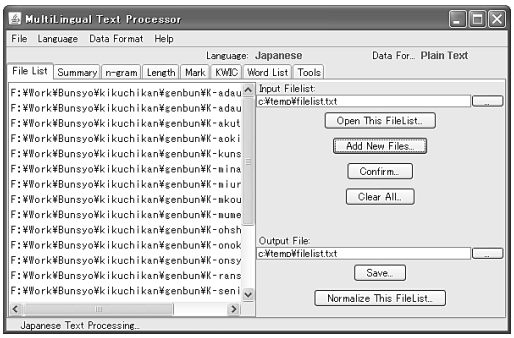


図３　ファイルが読み込まれた画面



⑶ 各タブの機能

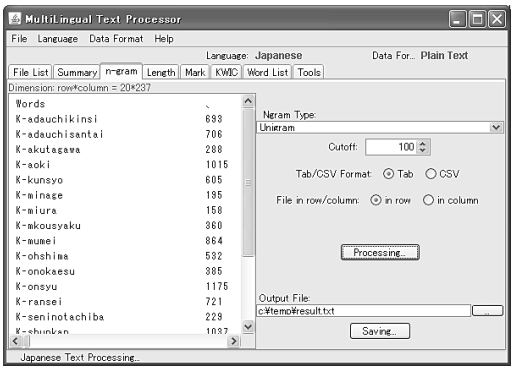
①「Summary」タブ

「Summary」タブは、読み込んだファイルごとのサイズ、漢字の数、平仮名の数、片仮名の数、文の数などのデータを集計する。集計結果は、タブ区切りとカンマ区切りの形式から選択できる。結果の保存は、「Output File」の窓でフォルダを指定し、さらにファイル名を付け、[Save...]ボタンを押す。

②「n-gram」タブ

「n-gram」タブをクリックすると、図４に示す操作画面が開く。n-gramのタイプおよび出力データの形式の指定は、操作画面の右上から順番に行う。まず、n-gramのタイプを「Ngram Type」の窓で指定する。UnigramからSix-gramまで集計可能である。次に、「Cutoff」の窓で値を指定する。この値は、出現度数が

図４　n-gramタブと集計の画面



非常に少ない項目をまとめるための閾値である。例えば、Cutoff値を100にすると、すべてのテキストにおける総度数が100未満である項目は、「OTHER」という項目にまとまる。データ表のサイズのコントロールに有効である。

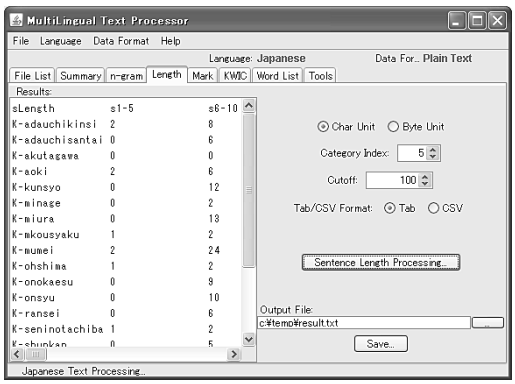
出力データの形式は、タブ区切りとカンマ区切りが選択できる。出力データの行をテキストにするには「in row」、列をテキストにするには「in column」を指定する。指定を終えたら、[Processing ...]ボタンを押すと、集計が始まる。集計にかかる時間は、用いたテキストの量と集計するn-gramのタイプに依存する。図５のメッセージ画面が現れると、集計が成功している。[了解]ボタンを押して「Processing Success」画面を閉じ、結果を保存する作業に進む。結果の解析は、データ解析の専用ソフトを用いることを前提としている。

図５　処理結果メッセージ

③「Length」タブ

「Length」タブでは、単語の長さ、文の長さを集計する。品詞タグが付いていないテキストの場合は、文の長さを集計することができる。文の長さを計るときに、何文字を１つの項目にするかは自由に設定できる。図６に、５文字を１つの単位とした文の長さの分布を求める画面コピーを示す。

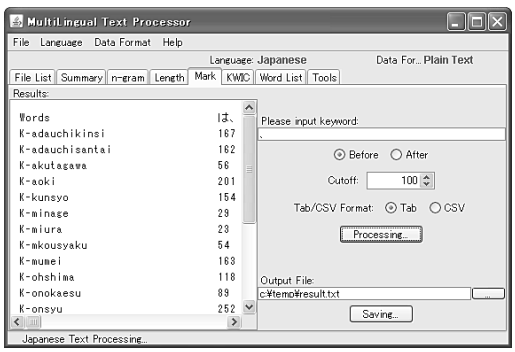
図６　Lengthタブで文の長さの分布を求める画面



④「Mark」タブ

「Mark」タブでは、個別の文字やマークを指定し、その前の文字、あるいはその後の文字に限定したbigramデータを集計する。図７に、読点「、」がどの文字の後に打たれているかに関して集計したMarkタブの画面コピーを示す。

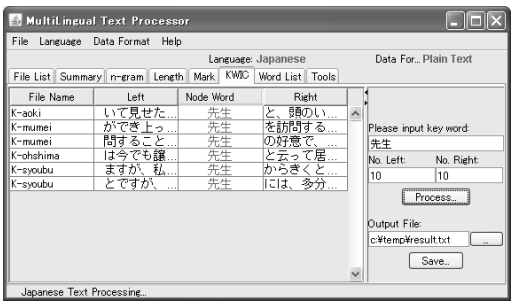
図７　Markタブの画面



⑤「KWIC」タブ

「KWIC」(クウィック、Keyword in Context)タブは、指定した文字列を検索し、その前後の一定の長さの文字列を切り取り、出力する。テキストサンプルを読み込み、「先生」をキーワードとし、その前後10文字（全角）を切り取った画面コピーを図８に示す。検索結果の表頭の"Left"あるいは"Right"を左クリックすると、三角マークが現れる。三角マークをクリックすると、検索結果を昇順・降順にソートする。結果は保存して、Excelなどに読み込んで用いることが可能である。

図８　KWICタブの画面



⑥「Word List」タブ

「Word List」タブは、単語リストを作成し、各テキストにおける、その単語の度数を集計する。単語リストでは、論理演算が用いられる。この機能は、見かけ上、異なる単語を１つのグループにしたいときに有効である。

⑦「Tools」タブ

「Tools」タブには、すべてのテキストについて、文字列を置換する機能、JUMANと茶筌の形態素解析結果をMLTPで用いる形式に変換する機能、文をランダムサンプリングする機能などを備えている。処理した結果は、指定したフォルダの中に、自動的に元のファイル名で保存される。

⑷　タグ付きデータの集計

MLTPでは、次に示すタグ付きtxt形式ファイルを前提としている。

だから＜接続詞＞、＜読点＞高松藩＜人名＞は＜副助詞＞、＜読点＞徳川宗家＜人名＞に＜格助詞＞とって＜動詞＞は＜副助詞＞御三家＜普通名詞＞に＜格助詞＞次ぐ＜動詞＞親しい＜形容詞＞間柄＜普通名詞＞である＜判定詞＞。＜句点＞

単語、文節の後に付けるタグは、全角山括弧＜＞で囲む。MLTPの「Tools」タブには、JUMAN、茶筌の形態素解析結果を上記の形式に変換する機能を備えている。ただし、JUMANは図９、茶筌は図10に示すようなフォーマットのtxt形式のファイルで出力されている必要がある。

図９　MLTPで用いるJUMANのフォーマット

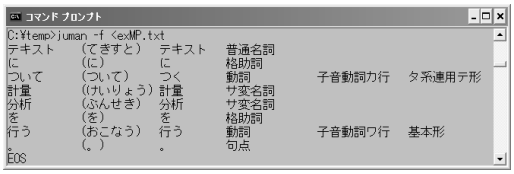
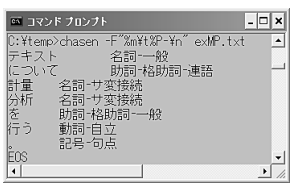


図10　MLTPで用いる茶筌のフォーマット



JUMAN、茶筌の出力結果をMLTPで用いる形式に変換する手順を次に示す。

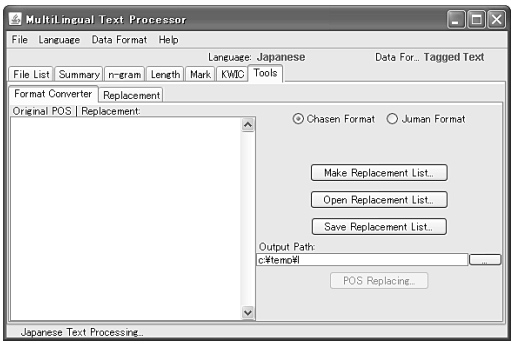
✧ MLTPのメニュー「Data Format」から「Targged Text」を指定する。

✧「File List」タブからJUMAN、茶筌の形態素解析結果のファイルを読み込む。

✧「Tools」タブをクリックし、アクティブにす

る（図11）。「Tools」タブには、２つのサブタブ「Format Converter」と「Replacement」がある。

図11　タグ付きテキストのToolsタブの画面



✧サブタブ「Format Converter」をアクティブにし、形態素解析ソフトの種類（Chasen Format, Juman Format）を指定し、[Make Replacement List...]ボタンを押すと、図12(a)の品詞選択ダイアログボックスが返される。[Select All...]ボタンを押すと、図12(b)のように、すべての品詞タグの前にチェックが付けられる。確認ボタン[Confirm...]を押すと、図13のような結果が返される。

✧形態素解析の品詞情報は、図13のように、ウィンドウの左側に、詳細の品詞情報と略した品詞情報を縦棒「｜」で区切って返す。MLTPに用いる品詞の表記は、この窓上で、「｜」の右の文字列を自由に修正・入力することによって変更できる。例えば、品詞「助詞ー副助詞／並立助詞／終助詞」を「副助詞」にしたいときは、次のように記述する。

助詞ー副助詞／並立助詞／終助詞｜副助詞

✧品詞タグの表記形式を修正した結果は、後に用いるため、[Save Replacement List...]ボタンを用いて保存することが可能である。また、[POS Replacing...]ボタンを押すと、

図12　品詞選択のダイアログボックス

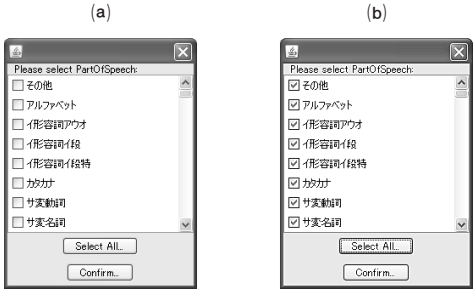
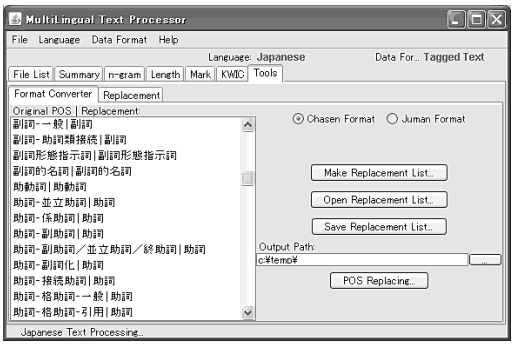


図13　品詞のタグ形式の変換画面



MLTPで使用可能なタグ付きtxtファイルがc:¥tempの中に保存される。ファイル名は、読み込んだファイルの名前と同じになる。ファイルの保存フォルダは、「Output Path」の窓で指定する。

このように処理したファイルを「File List」タブで読み込むと、「n-gram」タブで、品詞のn-gram、単語のn-gram、品詞を指定した単語のn-gramのデータを集計することが可能である。「Length」タブでは、単語の長さを集計することができる。

テキストから統計データを集計するツールとして、徳島大学総合科学部の石田基広氏は、R言語上で日本語テキストを統計的に分析するパッケージRMeCabの作成に着手し、精力的にバージョンアップを行っている。

http://cms.ias.tokushima-u.ac.jp/index.php?RMeCab