Chakoshi : A Japanese text search and collocation extraction application

日本語用例・コロケーション情報抽出システム『茶漉』

深田　淳
(パデュー大学)

キーワード
コーパス，コロケーション，用例検索，kwic

要　旨

本稿は，日本語コーパスからコロケーション情報を抽出することを主眼に設計・開発された，ウェブ上で簡便に利用できるアプリケーション『茶漉』を紹介する。ウェブ上のアプリケーションであるので，ブラウザでアクセスするだけで利用でき，ソフトウェアのインストール，サーバの設定のような導入手続きは一切必要としない。コロケーション情報を調べる以外にkwic形式による用例検索も可能である。当システムの公開が，コーパスを用いた日本語研究の発展の一助となることを願う。

1. はじめに

コーパスには様々な情報が眠っている。その中には手作業でなければ抽出できないような情報もあるだろうし，機械的処理によって取り出せる情報もあるだろう。コロケーション情報は後者に該当する。今回開発し，ここに紹介する日本語用例・コロケーション情報抽出システム『茶漉』は，ウェブ上で簡便かつ高速にコーパスからコロケーション情報を抽出することを可能にするものである[1]。コロケーションは，母語話者の言語使用を記述する上で重要な情報であるが，大規模なコーパスを対象に生起頻度などを手作業で調べることはほぼ不可能であるので，このようなツールを開発する意義は大きいと考える。コロケーションの重要性については，コーパスに基づいた辞書（Sinclair ed. 2001，Wehmeier ed. 2000など），文法書（Biber et al. 1999）などが英語の世界において重要視されていることからも窺い知ることができるし，日本語の世界でもコロケーションの概念を用いた研究が出始め，新しい知見をもたらしている（例えば，大曾他2004)。当システムの公開が，コロケーション情報を用いた研究の発展の一助となることを願うものである。

2. コロケーション

コロケーションとは，語と語の間の結びつきのことである。例えば，「ひんしゅくを買う」における「ひんしゅく」と「買う」が非常に強く結びついていることは，直感的にもわかる。実際，国語辞典の「ひんしゅく」の項には，「ひんしゅくを買う」という表現が通常記載されている。

日本語（または国語）を学習するという観点から見ると，「ひんしゅくを買う」などのイディ

オム的なものは，辞典にも載っているし，学習者にとってあまり問題にならない。ところが，辞典に一般に載っていないようなコロケーションが数多くあるのが問題になる。例えばコーヒーやお茶は（豆や葉を栽培する意味ではなく，飲み物を作るという意味において）通常「いれる」ものであり，「作る」ものではない。ところがこのコロケーションを知らない日本語学習者は，「お茶を作りました」というような文を作ってしまう。このような，何を言いたいのかはわかるが母語話者ならこうは言わない，という種類の誤用は，会話指導，作文指導などをしていると頻繁に見つかる。このようなコロケーションの習得は，母語話者らしい自然な言葉遣いにつながるものであるので，重要である（コロケーション一般に関しては，秋元 1994，Cowie 1998を参照）。

コロケーションに関連する統計指標としてはtスコア，MIスコアが代表的である（小池編 2003 pp.637-638を参照）。例えば「ひんしゅく」という語のコロケーションを調べるとする。この語の前後3語ずつ，合計6語を対象にするとしよう（この対象範囲はスパンと呼ばれる。詳細は後述する）。このスパン内に出現するすべての語と「ひんしゅく」を総当たり的に検討していくことになる。まず，tスコアは，スパン内に現れた語Wのコーパス内頻度とスパン内総語数から，Wがスパン内にどれだけ現れるはずか（期待頻度）を求め，実際のスパン内頻度との差をとるというものである。計算式は以下の通り。

$$t\text{スコア} = \frac{\text{スパン内頻度} - \text{期待頻度}}{\sqrt{\text{スパン内頻度}}}$$

ただし，期待頻度＝（コーパス内頻度÷コーパス総語数）×スパン内総語数

一方，MIスコアは，相互情報量とも呼ばれ，二つの語が共起する確率とそれぞれが個別に生起する確率との比を取って計算される。

$$\text{MIスコア} = \log_2 \frac{\text{語AとBの共起頻度} \times \text{コーパス総語数}}{\text{語Aのコーパス内頻度} \times \text{語Bのコーパス内頻度}}$$

「ひんしゅく」をこれらのスコアを使って調べると，「買う」が高いコロケーションを持つ語として現れる。

tスコアとMIスコアは，前者がコロケーションの有無の確からしさを表す統計量であるのに対して，後者はコロケーションの強弱を表すという点で異なる。またMIスコアは，語の頻度が比較的低い場合に，信頼できる指標を提供しないという欠点がある。従って，tスコアとMIスコアは語の頻度も参照しながら評価する必要がある。

3. 日本語用例・コロケーション情報抽出システム『茶漉』

本稿で紹介する『茶漉』は，コーパスから用例やコロケーション情報を簡便かつ高速に抽出する，ウェブ上で提供されるアプリケーションである。パデュー大学先端技術言語学習研究所のホームページから『茶漉』のリンクを辿っていくと，図1のようなページが現れる。ページ全体は

縦長なので，以下では四つのセクションに分けて順に紹介していく。

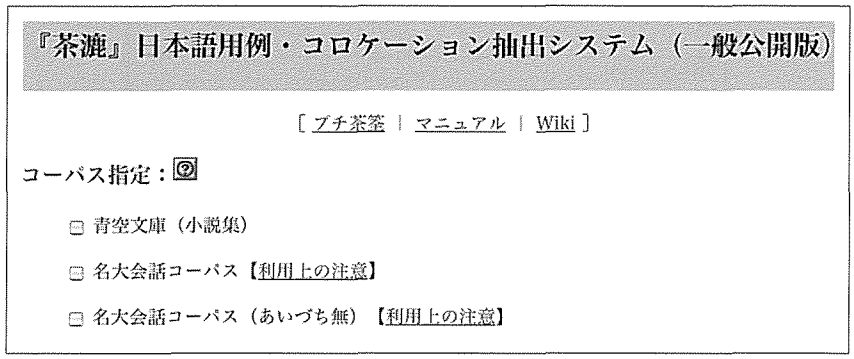

図1　『茶漉』その1

まず，「プチ茶筌」というリンクをクリックすると，次のような画面が別ウインドウ内に現れる。

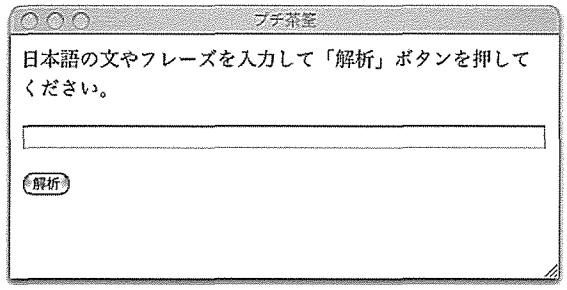

図2　プチ茶筌

これは，形態素解析システム『茶筌』[2]をCGI化したものである。ここに日本語の文や句，節を入力すると，『茶筌』がそれを形態素解析し，結果が図3のように表示される。

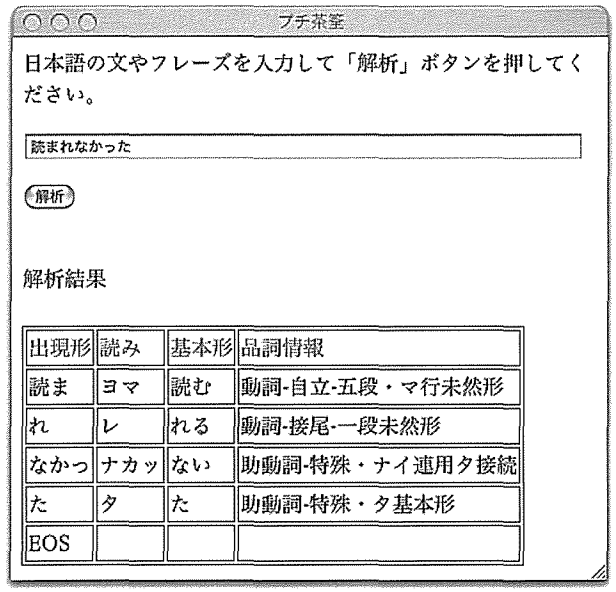

図3　プチ茶筌解析結果

以下で改めて説明するが，『茶漉』一般公開版ホームページにて現在提供されているコーパスは，形態素解析処理に『茶筌』を利用した[3]。従って，検索条件を設定する際に，『茶筌』が，ある日本語の文，節，句をどのように分析するかを知る必要が出てくる場合がある。「プチ茶筌」はその時のために用意されているのである。『茶筌』はIPADICという辞書を実装しているが，それは言語学や日本語教育における語の認定方法とかなり異なる場合があるため，利用する場合はその点に注意が必要である。形態素解析の結果が分析者の直感にそぐわない場合があるかもしれないが，それに合わせて条件を指定すれば，検索は成功するはずなので，検索上の便宜的なものと考えていただきたい。ただし『茶漉』自体は『茶筌』に依存していないので，言語学的に厳密な形態素解析をするシステムが登場すれば，データの書式を合わせるだけでそれを利用することができるようになる。

図1の「マニュアル」リンクは，クリックすると，オンラインマニュアルが呼び出せるようになっており，別ウインドウにシステムの概要，使い方，検索例が表示される。

「コーパス指定：」という文字の後にある[?]は，ヘルプボタンであり，これは『茶漉』の各セクションに付けられている。これをクリックすると，そのセクションに関連する操作方法の説明が別ウインドウに表示される。

「コーパス指定」のセクションにおいては，一つまたは複数のコーパスを検索対象に選択する。公開版の『茶漉』では，青空文庫[4]，名大会話コーパス[5]を無償で提供している。『茶漉』の仕様としては，この二つのコーパスに限定されておらず，いくらでも新しいコーパスを追加して検索対象にすることが可能である。その場合は，茶漉前処理プログラム群で新しいコーパスに処理を

施することになる。前処理に関しては以下でさらに詳しく述べる。

図4に示す「検索パターン設定」のセクションでは，まず，スパンの設定をする。

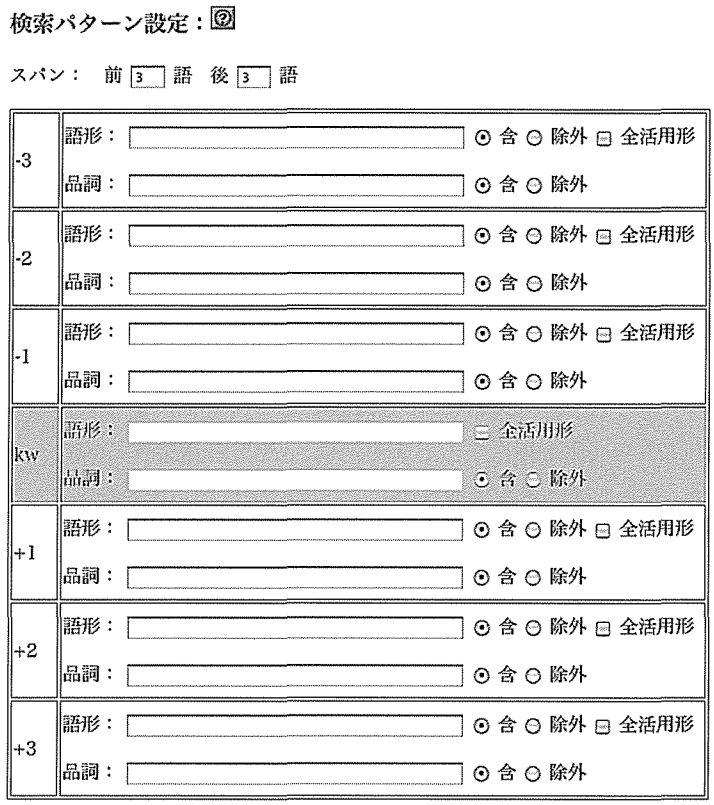

図4 『茶漉』その2

スパンとは，検索キーワードの前後の語数で規定され，コロケーションを持つ語を探す範囲となる。スパンは小さければ小さいほど処理効率がよくなるので，不必要に長いスパンを設定しないことが肝要である。スパン指定は，コロケーションの計算のみに関係するので，用例検索出力（kwic-keyword in context）には影響しない。

次に検索条件を指定するセクションに移る。まず中央部分にあるkwは，キーワードの意味であり，ここに検索したい語を入力する。検索範囲を絞るために品詞も同時に指定することができる。例えば，「で」と単に入力するとコピュラのダの連用形にも合致するし，助詞のデにも合致してしまう。語形＝「で」，品詞＝「助詞」と指定すれば助詞のデだけを拾い出すことができる。品詞には，活用形の違いなど，『茶筌』が出力する結果を細かく指定することができる。語形，品詞の両方で，複数の形式（例えば同じ語の異なる表記など）を「｜」記号で結んで一括検索することも可能である。例えば，語形に「桜｜さくら｜サクラ」と指定すると，語形が「桜」「さ

くら」「サクラ」のいずれかに一致する例が検索される。

「全活用形」というチェックボタンは，指定した語が活用語だった場合に効力を発揮する。例えば，語形＝「見る」と指定して「全活用形」をチェックした場合は，当該スロットに「見る，見，見れ，見ろ...」などの活用形が来てもよいという条件を指定したことになる。

キーワードについては，品詞のみ「含」「除外」のオプションを指定できる。品詞＝「助詞」，「含」と指定すると，品詞情報に助詞を含むものと合致し，品詞＝「助詞」，「除外」と指定すると，助詞以外の品詞のものと合致する，という条件になる。

－1～－3は，キーワードに前接する三語を指し，＋1～＋3は，キーワードに後続する三語を指す。とりあえずこれだけ指定できれば，実用に堪え得るだろうと判断した。キーワード以外のこれらのスロットでは，語形指定の際にアスタリスク文字を使って，前方一致と後方一致の指定ができる。「＊方」なら「方」で終わる語，「前＊」なら「前」で始まる語と合致する。なお，「｜」「＊」以外の正規表現のサポートはない。

検索条件指定の一例として，「こぼす」の目的語としてコロケーションの強い語を調べてみよう。まずコーパスに青空文庫を指定する。スパンは，「を」に前接するスロットのみを含めばいいので，前＝2語，後＝0語と指定する。そして，キーワードと－1のスロットを図5のように指定する。

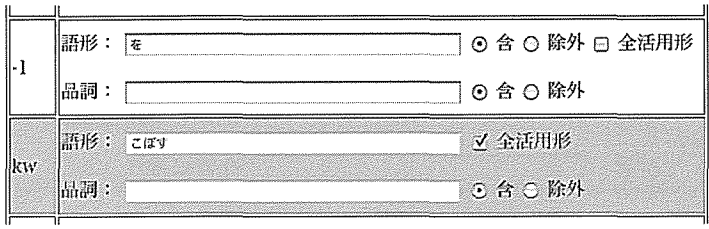

図5 「こぼす」の目的語を検索

これでコロケーション情報抽出を実行すると，図6のような結果が得られる。

コーパス総語数= 8370720
スパン語数=183
kwコーパス頻度=116

形態素	tスコア	MIスコア	コーパス頻度	スパン頻度	期待頻度
嬉し涙	2.236	14.65	14	5	0.0003
愚痴	3.605	12.85	127	13	0.0028
涙	5.286	10.70	1217	28	0.0266

図6 「こぼす」のコロケーション出力

tスコアの最大値は「涙」の5.286であるが，tスコアは，大きければ大きいほど，その語と「こぼす」の共起が単なる偶然の所産である確率が低いことを示す。MIスコアでは，「嬉し涙」＞「愚痴」＞「涙」の順になったが，これはコロケーションの強い順に対応している。

最後の二つのセクションは，次の図が示すようになっている。

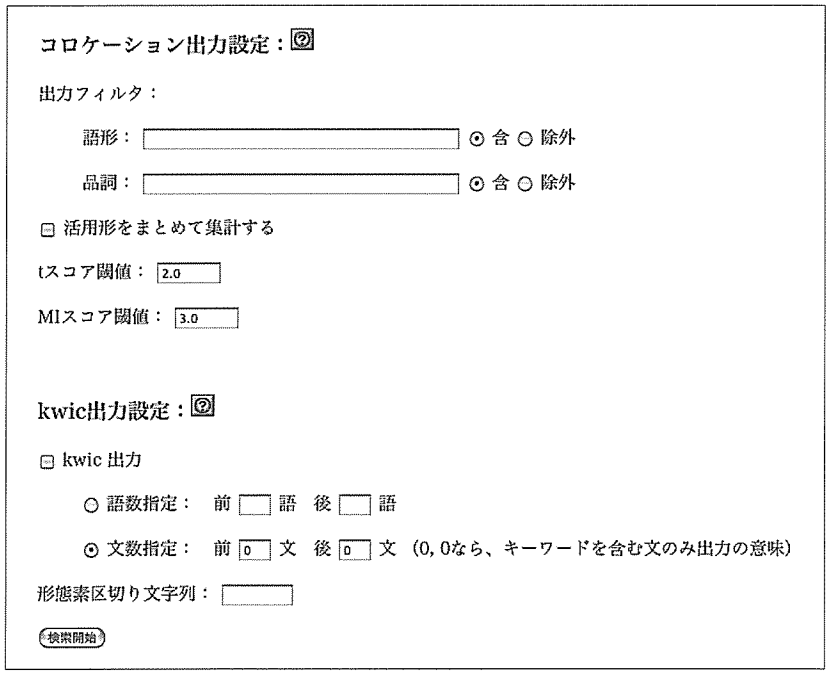

図7 『茶漉』その3

「コロケーション出力設定」のセクションでは，まず出力フィルタが指定可能である。これは，コロケーション強度の計算に含める語を制限するものである。例えば，上記の「こぼす」の例で言うと，助詞ヲは検索条件にも含まれているため，コロケーションが強く出ることは必定である。従って，語形＝「を」「除外」，と指定しておけば，コロケーションの計算結果にヲは現れなくなる。フィルタは品詞指定も可能である。

「活用形をまとめて集計する」というオプションを選択すると，「こぼさ」「こぼし」「こぼせ」などの同一語の活用形を同一語と見なして集計する。個々の活用形ではコロケーションが弱くても，それを一語にまとめて扱えば強くなるケースも出てくる。

tスコアおよびMIスコアの閾値とは，ここで指定された値を越えないとコロケーションが確かにあり，十分に強いとは言えないことを表す値のことである。デフォルト値はt=2.0，MI=3.0で，これはコーパス言語学でよく使われる目安である。この値を高くすると，コロケーションがあることの確からしさの低い語，コロケーション強度が低い語は表示されなくなる。

コロケーション情報以外に，kwic形式の出現文脈を見たい場合には，「kwic出力設定」セク

ションの「kwic 出力」ボタンをチェックしておく。出力形式としては，キーワードの前後の語数指定とキーワードを含む文の前後の文数指定のどちらかを選択できる。また，「形態素区切り文字列」で形態素の境界を示す文字列を指定できる。デフォルトは半角スペースである。以下に文全体の出力例を挙げる。

［1］ 涙 を こぼし て ママ さん ママ さん と 云っ て 笑う の です 。

図 8　kwic の出力例

［1］は，第一番目に指定したコーパスからの用例であることを示している。キーワードは，ここでは下線で示してあるが，画面上では赤色で表示される。

当システムは，世に数多く存在する全文検索システムやGREPサーチツールとは次の二点において異なる。第一は，コロケーション情報抽出を主眼においている点。第二は，形態素解析処理が施されたコーパスを用いるので，形態素タグを検索式に含めることができる点。形態素タグを用いると，例えば「「行く」の全ての活用形」「接続助詞のガ」のような検索が簡単にできる。機能的に最も近いツールは『茶器』[6]であるが，『茶漉』はブラウザで利用できるCGIである点，コーパスは限られるが一般公開サイトでインストールなどの手間なしにすぐに利用できる点，tスコアが計算できる点，コロケーション計算用に最適化されている点などにおいて異なる。特に言語研究を専門とし計算機技術に疎い研究者には，一般公開サイトのような提供の仕方が最も有用だろうと思われる。

4. 茶漉データ生成ツール群

『茶漉』は，用例検索やコロケーション計算を高速で処理する。処理速度は，検索条件の複雑さ，指定コーパスの数や規模，サーバの性能，ネットワークの速度や混み具合い，など様々な要因に左右されるが，概ねリアルタイムで対話的な作業が可能になっている。このような高速処理を可能にするためには，コーパスの前処理が非常に重要である。つまり，事前にできる処理はなるべく全部完了して，その結果をファイルに書き出し，それを参照するようにするのである。例えば，形態素解析は『茶筌』などを利用して事前に行なう。また，各単語のコーパス内頻度は変化するものではないので，予め計算しておく。このような一連の前処理は，C言語とPerl言語で記述された茶漉データ生成ツール群およびUNIXのテキスト処理ツールを組み合わせて行なう。前処理の概略は以下の通りである。

- 『茶筌』などによる形態素解析
- 形態素頻度リストの作成
- 形態素タイプインデックスファイルの作成（指定された形態素を高速に検索するため）
- 形態素トークンインデックスの作成（各トークンにアクセスするため）
- 全ての活用語のタイプおよびトークンのリスト作成
- 活用語のタイプインデックスを作成（高速検索を可能にするため）

これらのデータファイルは，『茶漉』の処理効率がなるべく高くなることを念頭に構造設計されたバイナリーファイルであり，バックエンドにデータベースなどを用いていない。また特殊な文字列処理ライブラリなども用いていない。現在使用されている文字コードはEUCである。

5. 応用例

『茶漉』は，日本語学，日本語教育の研究に広く応用が可能だと思われる。以下では応用例を二つ紹介する。

まずは，用例検索機能を使って，ある形態素や構文の使われている実例を収集するという使い方がある。ここでは「追う」という動詞を例にとってみよう。牧野他(1999)には，「追う」の動作主は有生名詞と記載されている。直感的には正しいと思われるが，実際にはどうか調べてみる。毎日新聞の一年分[7]を対象に，kw語形＝「追う」（全活用形指定）で検索を行なった。出力は，前5語，後2語に絞った。条件に合致する受動文434例のうち，上記辞典の記述の反例と思われるものが，418例（全体の96%）も見られた。

[1] 年末 ギリギリ まで **仕事** に 追わ れる 人
[1] 部下 が 毎日 **残業** に 追わ れ て
[1] て から は **仕事** に 追わ れ ，
[1] 企業 や **相続 税** に 追わ れる 人
[1] 学校 と の **連絡** に 追わ れ て
[1] 米 軍 は **対応** に 追わ れ て
[1] で の **情報 収集** に 追わ れ た
[1] 忙しく 日々 の **営み** に 追わ れ て
[1] は「 サラ金 **返済** に 追わ れ た

この結果から，少なくとも受動文においては「仕事」や「対応」といったものが「追う」の動作主として機能しており，従来の辞典記述に補足すべき点のあることが分かる。

次に「沸かす」と「沸かせる」の使い分けについて調べてみたい。「沸く」の他動詞である「沸かす」と使役形の「沸かせる」は競合していると考えられるが，何らかの使い分けがなされているのか，なされているとすればそれはどんなものか，を調べてみたい。十分な用例数を確保するために毎日新聞の六年分を検索対象として，スパン前3後0，－1語形＝「を」，kw語形＝「沸かす｜わかす」（全活用形）という条件と，スパン前3後0，－1語形＝「を」，kw語形＝「沸か｜わか」，＋1語形＝「せる」（全活用形）という条件で二回の検索を行なった。その結果，「沸かす」の目的語でコロケーション出力に現れた語は，MIスコア順で「お湯」「湯」「茶」「ふろ」「会場」「水」の6語であった。一方，「沸かせる」の方は，「客席」「場内」「観客」「観衆」「土俵」「スタンド」「会場」「ファン」「甲子園」「席」「大会」「日本」の12語であった。この結果から，「沸かす」が「水などの液体に熱を加えて熱くする」という本来の意味を，「沸かせる」が「熱狂，興奮させる」という比喩的な意味を受け持つという明らかな傾向が見られた（唯一の例外は「会場を沸かす」であるが，「沸かす」が274例あるうち，「会場」と共起しているも

のは6例のみと少数であった)。

6. むすびにかえて

以上で紹介したように,『茶漉』は高速で対話的な処理が可能であり,特殊なコンピュータ知識・技能を必要とせず,簡便で手軽に利用できるように設計された用例・コロケーション情報抽出システムである[8]。日本語関係者に一般公開サイトを広く利用していただき,ご批判,ご意見を賜りたいと考える。

このようなサービスを提供する際に問題になるのが,一般公開できるコーパスの数である。現在のところ,「青空文庫」と名大会話コーパスのみが一般公開可能である。これだけではどんなに優秀なツールを作っても,その有用性は限られてしまう[9]。そこで一般公開が可能なコーパスがあれば,提供していただくよう広く呼びかけたいと思う。

注

1 本システムは無料で一般公開する。アドレスはパデュー大学先端技術言語学習研究所(英名:Center for Technology-Enhanced Language Learning)のサーバで,http://tell.fll.purdue.edu/である。

2 奈良先端科学技術大学院大学松本研究室開発。http://chasen.naist.jp/hiki/ChaSen/

3 形態素解析システムによって形態素の分節方法が変わってくるため,今回『茶筌』で解析を行なって公開してあるコーパスから得られるコロケーションの値は,あくまでも『茶筌』とIPADICを使った場合の一例と考える必要がある。さらに口語的表現,方言などが含まれる部分には解析ミスがかなり出る可能性があるが,このサイトで提供しているコーパスでは,いまのところそのようなミスを手作業で修正していないし,解析精度の概算もしていない。これは将来的課題である。

4 インターネットの小説サイト「青空文庫」(http://www.aozora.gr.jp/)から許諾を受けて現代語の小説を抜粋したコーパス。総語数は8,370,720語。

5 名大会話コーパスは,平成13年度～平成15年度に行なわれた科学研究費補助金による研究「日本語学習辞書編纂に向けた電子化コーパス利用によるコロケーション研究」(研究代表者・大曾美恵子)の一環として構築された雑談会話コーパスである。総語数は1,217,768語。最終的にはこの倍ぐらいにする予定。

6 奈良先端科学技術大学院大学松本研究室開発。http://chasen.naist.jp/hiki/ChaKi/

7 1991年に発行された毎日新聞。総語数は27,653,257語。

8 将来的課題としては,コーパスの拡充の他には,多言語対応を射程に入れたEUCからユニコードへの文字コードの移行,コロケーション出力のソート機能,正規表現のフルサポート,現有コーパスの形態素解析の精度の調査,などを考えている。

9 公開不可能なコーパスを自分の仕事場で『茶漉』で検索できるようにしたいという研究機関には,『茶漉』自体をバイナリー形式で提供する準備もある。その場合は,UNIXサーバが扱えることが前提となるので注意されたい。『茶漉』はC言語の標準的な関数のみで記述されているので,WindowsやMac OS Xサーバに移植するのは容易だろうと思われるが,まだ移植作業は行っていない。

参考文献

秋元実治(1994)『コロケーションとイディオム：その形成と発達』英潮社

大曾美恵子他(2004)『日本語学習辞書編纂に向けた電子化コーパス利用によるコロケーション研究』平成13年度～15年度科学研究費補助金基盤研究(B)(2)(研究課題番号 13480069) 報告論文集

小池生夫(編)(2003)『応用言語学事典』研究社

牧野成一・中田清一・大曾美恵子(1999)『日常日本語バイリンガル辞典』講談社インターナショナル

Biber, D., E. Finegan, S. Johansson, & G. Leech(1999) *Longman Grammar of Spoken and Written English.* Harlow, Essex: Longman.

Cowie, A. P.(ed.) (1998) *Phraseology: Theory, Analysis, and Applications.* Oxford University Press.

Sinclair, J.(ed.) (2001) *Collins Cobuild English Dictionary for Advanced Learners.* Third Edition. London: HarperCollins.

Wehmeier, S.(ed.) (2000) *Oxford Advanced Learner's Dictionary of Current English.* Sixth Edition. Oxford: Oxford University Press.

(投稿受理日：2006年11月14日)

(最終原稿受理日：2007年5月15日)

深田　淳（ふかだ　あつし）

パデュー大学先端技術言語学習研究所

640 Oval Drive, West Lafayette, IN 47907-2039, USA

afukada@purdue.edu

Chakoshi: A Japanese text search and collocation extraction application

FUKADA Atsushi
Purdue University

Keywords
corpus, collocation, text search, KWIC

Abstract

This article introduces Chakoshi, a user-friendly web application designed and developed to extract collocations from Japanese text corpora. Because it is a web application, users need only to access the application web site, and there is no need to install special software or configure a server computer. In addition to extracting collocations, it is also capable of generating KWIC output in response to search queries. It is hoped that this publicly available system will help promote corpus-based Japanese linguistic research.