自 然 言 語 処 理 152－17
（２００２．１１．１３）

# web検索に基づく多言語動的KWIC

田中久美子† 山本真人† 中川裕志‡
† 東京大学 大学院 情報学環 ‡ 東京大学 情報基盤センター
E-mail: kumiko@ipl.t.u-tokyo.ac.jp, {masato-y,nakagawa}@dl.itc.u-tokyo.ac.jp

web 上の検索を用いて多言語の語彙用例を調べるツールを開発したので報告する。このツールは、用例のためのデータを検索エンジンから動的に得るもので、コーパスや辞書をツール内に一切持っていない。さらに、言語に非依存の解析ルーチンだけを利用しており、言語依存性がないことに大きな特徴がある。このため、多言語の生きた用例を調べることができるという利点がある。本稿ではシステムの構成を論じた上で、有用性に関する評価結果を述べる。

# Multi-lingual Dynamic KWIC based on Internet Search

**Kumiko TANAKA-Ishii† Masato Yamamoto† Hiroshi Nakagawa‡**
†Graduate School ‡Information Center of the University of Tokyo
The University of Tokyo

We present our web based dynamic KWIC system based on the internet search. When the user enters a string of words that he wants to find the usage for, the system sends the query to the search engines to obtain the corpus about the string. The corpus is then statistically analyzed and the results are displayed. As the system does not use language dependent analysis nor initial data, queries can be made in any language, even those without well established analysis methods. Also, as the corpus is dynamically obtained, the usages given to the user are always up to date.

## 1 はじめに

インターネットの普及により、国際語としての英語へのニーズが高まると同時に、英語以外の言語に接する機会も増えている。このように外国語が身近な存在となった現在、生きた言語の用例を調べる必要性は断然高まっている。

言語の用例を調べるには、古くから辞書が用いられてきた。辞書には精選された項目が記載されており、普遍的な用例を調べる用途には有用である。しかし、一方で今日的な用例が見つからなかったり、また、自分の望む具体例が載っていないことが多く、外国語の運用上は自分の語用が正しいのかどうか、不安が残ることも多い。

80 年代後半に、全文検索のための技術が提案されると [3]、応用として大きなコーパスをKWICとして用いることが一般的となった。日本語でも最新のソフトウエアの一つとして内山ら [6] が数GBのコーパスを瞬時に検索するツールを公開しており、言語の用例を調査するのに大変に役に立つ。しかし、KWICシステムが個別のコーパスの種類に依存することは宿命であり、必ずしも現代的な用例が得られない場合が多い。

以上の問題点を解決すべく、web 上の文書をコーパスとして用い、動的な KWIC ツールを作成することは、自然な発想であり、過去にも類似の提案例がすでにある [7][2]。しかし、これらは英語に対するきわめて限定されたもので、調べたい語の前後数単語を集計して表示するだけのものである。多言語への適用、用例の調査

の方法とその限界、あるいはその精度は未だ明らかにはなっておらず、研究の余地がある。

そこで我々はどの言語でも用法を調べられるkiwiを開発し、その評価を行ったので、本稿でこれを報告する。本システムはユーザが調べたい語を正規表現で入力すると、その語に関するページを検索エンジンに問い合わせる。結果として得られたページを統計処理し、用例を提示する。用例を得る母体となるデータを常に動的に得るので、本システムを用いると最新の生きた用例を得ることができる。また、kiwiでは動的に得たデータの解析手法として、言語に非依存のものを用いているため、多言語の用例を調べることができ、形態素解析などの解析ツールが整備されていない言語であっても用例を調べることができる。kiwiはこのようにシステム内には言語依存の情報を一切持たない点に大きな特徴がある。以下ではkiwiの概要を述べた後、動的な解析の手法について述べる。最後に日本語、仏語、独語、英語についてkiwiを適用してみた評価結果を示して、本システムの有効性を論じる。

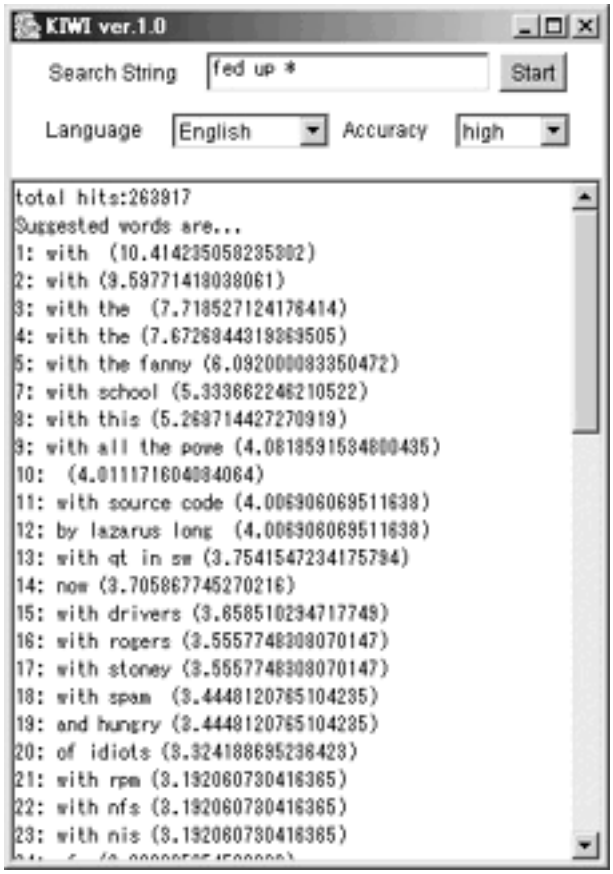


図 1: kiwiシステムの使用例

## 2 kiwiシステムの概要

### 2.1 使用例

kiwiはJava言語で書かれたシステムであり、ネットワーク上の検索エンジンを利用することが前提となっている。このため、高速に通信が可能なネットワークにつながった状態で起動されるソフトウエアである。図1にkiwiのGUIを示す。上面の横長の入力部分に2単語 (fed up) が入力され、その後に続く文字列を＊により調べている。ここで、Accuracyとは、検索エンジンから採取するコーパスの量を示している。採取するコーパスの量が多ければ候補の精度は向上するが、システムの応答速度は遅くなる。このシステムでは精度と応答速度はトレードオフの関係になっている。図では言語は英語である。また、図ではAccuracyはhighとなっており、用例に関する最初の300例を用いるように設定されている。

ユーザが右上のstartボタンを押すと、システムは特定の検索エンジンにユーザの入力を問い合わせる。図の場合には、Altavistaに問い合わせている。検索結果は集計され、下方の大きな枠内にfed upの直後に現れる文字列の候補が示されている。単語は、§3で論じる統計量で整列されており、単語の直後の括弧内に示されているのがその統計量である。同じ枠内の最上段には検索エンジンでヒットした入力の件数が表示されている。

結果には、withが第一候補に上がっており、日本人であれば高校で習う熟語が用例として与えられている。また、withの後に頻出した語も現れておりtheやsource codeといったものが示されている。

一般にfed up withの後には、動名詞が来ることがあることが辞書には記載されている。しかし、実際には、動名詞の用例は、この候補にはあがっておらず、生きた英語としては、fed up with名詞という用法が多い、ということもわかる。

このように、kiwiシステムは検索結果を用例の観点から集計しているだけの簡単なシステムであるが、外国語学習者には有用な情報が得られていることがわかる。

### 2.2 動的な単語切り出し

以上の使い方は何も英語に限ったことではないため、言語を指定することにより、英語以外の用例も調べることができる。ここで問題となるのは、言語による差異、たとえば分かち書きの有無による解析の差異や、文字集合の差異などがあるため、これをどのように処理するかという点である。

無論、kiwiの中に各言語ごとの解析手法を持たせ、ユーザが言語を切り替える際に、解析手法も切り替えて語法を調べることが解決策の一つとして考えられる。しかし、このようにすると解析手法が確立していない言語や、辞書がない言語には適用できない。そこで、いかなる言語にも対応できるシステムにするために、我々は個別の言語に依存する要素をシステムに含めない方向でkiwiを設計した。

そもそもKWICには、検索文字列の前後を一定長切り出し、動的に集計してユーザに提示するものが多い。この傾向は日本語を始めとする分かち書きしない言語では特に顕著であり、言語に依存しない手法となっている。そこでkiwiでもこの方法を取り入れて検索ページを解析するものとした。とはいえ、一定長切り出すだけとすると、全体の語用の傾向は人間の判断に任されることになってしまう。そこで、文字列の重複を調べることにより、単語相応部分を動的に抽出して、これを集計して提示するものとした。図1に示したのは、動的な単語切り出しの結果である。

文字集合についても、本システムがJava言語で書かれていることもあり、Unicodeで文字列を扱って汎用性を高めている。このように、特定の言語に依存しないシステムとして設計している。現在はAltavistaを主検索エンジンとしており、AltaVistaでサポートする25の言語は本システムで用例を調べることができる。

### 2.3 正規表現による質問入力

図1の例では、質問入力として2単語を与えたがkiwiにはより柔軟な入力として正規表現に近いものが与えられる。これにより、直後の単語のみならず直前の単語や、2単語間に来る候補を検索する事も可能である。また、“ly” で終わる文字列や、一単語離れた “s” で始まる文字列を探すなど柔軟な検索を行う事が出来る。

正規表現による入力を用いると、具体的な語用を元として用例を調べることが出来るにとどまる。例えば、英語においてある文字列の後に来る、特定の品詞の単語を調べるといったことはできない。これは言語に汎用にすることと引き換えにシステムに加わる制限である。しかし、たとえば “ly” で終わる、など、文字列に品詞が現れるような場合には、用例を調べることはできるし、また、特定の前置詞に関する用例をkiwiに前置詞を含めて入力を行って、(たとえば、fed up ofなどと入力して) 調べることができる。

以上から、kiwiシステムの本質が、候補の動的な切り出しと、それらの整列に集約されることがわかる。この点をどのように行っているのかを次節で論じる。

## 3 用例の処理

候補の切り出しは、頻出するn-gramの抽出と問題は類似している。しかし、本稿での問題は、

- 候補を切り出す検索結果は数千単語程度の小さなコーパスである。
- 動的に候補を得るため、高速な処理が必要である。
- 切り出し後に整列するため、切り出しと整列を統一的に扱いたい。

という3点の特徴がある。このような特徴を考慮して候補の文字列の生きた言語表現としての良さを評価する方法を考えなければならない。

直感的には、ある文字列が候補かどうかは、

- 適当な長さである (極端に短くも、極端に長くもない)
- 頻出する

- 後続する文字の種類が多い

という性質を満たす。例えば al という文字列の頻度が高くでも、大多数の場合に all の一部として出現するなら、むしろ all に大きな重みが与え、all を重要な文字列として切り出したい。

この考え方は Ananiadou らにより、コーパスに現れる多数の単語列から複合語を抽出するための C-value という評価関数 [1] においてすでに提案されている。C-value は本来、入れ子になった連語 (collocation) を認識し抽出するために考案された単語列の評価関数である。その特徴は、ある単語列を単に頻度の高さだけで評価するのではなく、安定して使われる単語列のうちできるだけ長いものを高く評価する点にある。これは上の語の切り出しの特徴とよく類似しているため、C-value のアイディアを元にして候補評価関数を定義することにした。ただし、Annaniadou らは語を単位としていたのに対し、我々は文字を単位としているので、その点について変更し、以下のように SC(String C-Value) を定義する。

$X$ を文字列として、$|X_i|$ を長さ $i$ の文字列とする。頻度を $N_i$, $X_i$ に続く文字の種類数を $C_i$ としてつぎの SC 値により $X_i$ を評価する。

$$SC(X_i) = log(i+1) \times log(N_i) \times (1 - \frac{1}{C_i}) \quad (1)$$

上の SC 式の 3 つの項には、候補かどうかの直感的性質にそのまま対応する。第一項が長さ、第二項が頻度、第三項が続く文字種に関するものである。文字列 $X_i$ の SC 値が文字列 $X_{i-1}$ の SC 値より高いという事は $X_i$ は $X_{i-1}$ よりも続く文字の種類数が多く、且つ頻度はそれほど減少していないことを意味する。

この SC 値を用いて候補文字列を得る。まず、入力した質問に後続する文字列を検索する場合は質問入力の直後から一文字ずつ文字列を増やしながら SC 値を計算していく。そして、以下の式を満たすときに $X_i$ を候補文字列とする。

$$SC(X_i) > SC(X_{i-1}) \quad (2)$$

質問の前方にくる文字列を検索する場合は質問入力の直前から前方に向かって SC を計算していき、同様にして候補を得る。

中間文字列を検索する場合は質問として A * B を与える。(ただし、A と B は文字列) この時 A の直後から B まで一文字ずつ文字列を増やしながら SC 値を計算していき、(2) によって候補を得る。

本方法は、局所的な SC 値だけで候補にするかどうかが決まる点に一つの特徴がある。そこで、候補に該当する可能性のある文字列を Trie として表現しておくことにより、高速に候補切り出すを行うことができる。以上から、処理の流れは以下のようなものとなる。

1. ユーザが入力した正規表現に関する検索結果を得る。
2. 検索結果のうち、正規表現に該当する部分を Trie で表現する。
3. Trie を全探索し、上の条件を満たす候補を切り出す。
4. 切り出した候補は SC 値により整列する。

これをユーザが用例を調べるたびに動的に行う。

# 4 評価

## 4.1 定型用例の検索

まず、kiwi を用いてどの程度定型的な用法が調べられるのかを調べる。表 1 に、英語、仏語、日本語での決まった用例の検索結果を示す。まず、語学学習者に利用される頻出熟語集等の中から、ランダムに 100 例挙げる。これらは、熟語は正解が二つ以上あるもの、take it easy と take it away などは除いて長さが 3 単語以上から構成されるものをランダムに選ぶ。その上で、各例を 3 分割し、そのいずれか一部を取り除いて検索した時に候補の中に取り除いた語が現われるかどうかを調べる。熟語を 3 つの部分に分割するので、取り除く部分によって前 (熟語の先頭部分を取り除いて検索)、中 (中間の一部)、後 (末尾部分) として表には記載した。

尚、熟語集は英語は TOEFL の熟語集 [10]、仏語は仏検の熟語集 [8]、日本語はことわざ辞典 [9] を用いた。

各言語につき、

- 出現率:100 例中、上位 10 位以内の正解数

表 1: 熟語の用例検索正解率

| | 出現率 出現率 | 第一 候補率 | 候補の 平均順位 |
|---|---|---|---|
| 英 前 | 100 | 81 | 1.288136 |
| 英 中 | 60 | 54 | 1.0769231 |
| 英 後 | 98 | 83 | 1.4576271 |
| 仏 前 | 80 | 53 | 1.978723 |
| 仏 中 | 64 | 60 | 1.310345 |
| 仏 後 | 95 | 76 | 1.407407 |
| 日 前 | 91 | 83 | 1.153846 |
| 日 中 | 96 | 93 | 1.0625 |
| 日 後 | 97 | 88 | 1.71134 |

- 第一候補率:100 例中、正解が第一候補として提示された数
- 平均順位:10 位以内に現れた正解が平均何番目に現れたか

について調べた。正解かどうかは、文字列が候補の一部にあれば、正解と判断した。

表によれば、前後の用例であれば、高い正解率が得られている。平均順位からも、候補として挙がる場合ではほぼ確実に第 1 位に候補が現れている事が分かる。第一候補として正解があがらない場合は、いずれも除いた単語が内容語で、機能語のみから内容語の用例を調べると言った場合であった。例えば、fed up with の第一単語を除いた場合には sign up with, keep up with などが現れ、特に間違っているとは言いがたいものが多かった。

中間の候補検索では英語、仏語共に好結果を得られていない。これは AltaVista の仕様が原因となっている。AltaVista の検索にはフレーズ検索と AND 検索の 2 種類がある。フレーズ検索では入力した文字列そのものが現れるページのみを検索するので、絞り込まれた検索結果が得られる。したがって、フレーズ検索を用いている後方、前方検索では良い結果が出ている。しかし、AltaVista のフレーズ検索では同時に複数のフレーズを検索する事が出来ないため中間の候補を検索する際には用いる事が出来ない。そこで本システムにおいては中間候補の検索に AND 検索を用いているのだが、AND 検索では質問入力における単語の順序は考慮されない。そのため、質問入力の語順通りでないページが検索結果として多数出てきてしまい、絞り込まれない。結果として中間候補の検索では精度が落ちている。この問題は分かち書きのある言語に特有の問題である。しかし、質問入力の単語が内容語の場合は良い結果が得られる。例えば、pay * card とすると by credit や with credit が候補として挙がる。なお、分かち書きをしない日本語の場合にはこのような影響を受けないため、中間候補検索でも前方、後方と同程度の良い結果が得られている。また、将来的に検索エンジンが絞込み検索に対応すればこれらの問題にも対処する事が可能となる。

次に 2 単語に関する調査として、仏語と独語における名詞の性を調べるテスト各 20 単語ずつ行った。すなわち、性を調べたい名詞の前方検索を行い、性を現す冠詞が第何番目に現れるかを調査した。表 2 に結果を示す。

表 2: 名詞の性の検索正解率 (仏、独)

| | 出現率 | 候補の平均順位 |
|---|---|---|
| 仏語 | 100% | 2.285714 |
| 独語 | 95% | 3.368421 |

表からは、高精度で目的の冠詞が得られていることがわかる。特に独語のように格変化によって冠詞が変化するような複雑な場合でも名詞の性を特定するが出来る。このように、本システムは簡易辞書として十分に用いることができる。

TOEFL や仏検は高度な語学能力を有する学習者が受ける試験であるが、その際に学習される定型熟語がこのように高い正解率で調べられるのは、本システムの有用性を示しているといえるであろう。

### 4.2 生きた用例

kiwi の一つの特徴は、生きた用例を調べられる点にある。本節では、辞書には載っていない現在よく使われている用例が kiwi により得られることを示す。表 3 に検索結果を適宜記載する。いづれも、上位 1 位 2 位が既存の辞書 [4] に載っていないが、周知の用例であるものを選

表 3: 生きた用例

| 検索正規表現 | 第一候補 | 第二候補 |
|---|---|---|
| * boys | backstreet | the |
| Star * | Wars | online |
| Osama * | Bin Laden | Bin |
| * Bach | Johann Sebastian | Richard |
| taxable * | income | fixed income |
| * McCartney | Paul | Linda |
| Zinedine * | Zidane | Saualem |
| tour * | de France | du monde |
| Peugeot * | motocycles | sport |
| * Elysees | Champs | Hotel Champerret |
| * 首相 | 小泉 | マハティール |
| 鈴木 * 逮捕 | 宗男議員秘書ら 7 人を | 宗男 |
| 読売 * | ジャイアンツ | 新聞 |
| ハリーポッター * | と賢者の石 | 100 の質問 |
| 東京 * | 都 | 大学 |
| * 純一郎 | 小泉 | 伊谷 |
| せっかく * | だから | お返事頂いたのですが |
| オマエ * | モナー | モナ |
| free * | guestbook | software |

んで記載した。

表からは辞書や KWIC では得ることのできない生きた用例が確認できる。たとえば、11 番目の例では、現在の首相を探したり、映画の題を調べたりすることも可能であることがわかる。語の用例は、時代と共に移り変わっていくものであるが、kiwi を用いると、これらをも捉えられる点に大きな特徴があるといえる。

この表からは、例えば “free *” や “オマエ *” の例からインターネットらしい偏りが伺える。従来型の KWIC の一つの難点としては読み込んだコーパスに用例が制限されることが挙げられたが、本稿の kiwi にも同様の宿命があることは当然であり、インターネットの文脈に制限がかかった用例の調査に限定されることは否めない。

## 5 関連研究

検索を工夫して用いることによる有用なシステムは多数提案されている。対訳語を検索を用いて直接得る研究や [5] に始まり、より一般的には Question-Answering システムについては多くの論文がすでに示されている。

これらの多数の提案はいわば、我々の提案の一歩先の研究である。我々は、より基礎的な観点で、web のデータから用例を抽出することを試み、web 文書に内在する言語知識の質を捉えようとした。むろん同種のアイデアは webcorp[7] によりすでに英語についてはサービス化されているし、Brill ら [2] も同じ主張を行っている。しかし、これらの研究はいずれも英語に対するものに留まり、英語は分かち書きする言語であるため、実現は日本語と比較すると易しい。

そこで、本研究では多言語化を目指し、英語以外の言語についても、インターネット上の文書の質を確かめようとした。むろん日本語特有のシステムとしての研究の道もあったが、それでは常に言語に依存した解析手法や辞書が必要となるものとなってしまう。現在では、世界の距離が多くの意味で縮まり、さまざまな言語が身近な存在となりつつあるので、どの言語でも

使える用例検索システムを目指し、文字ベースのシステムを開発した。

尚、検索エンジンで語の用例を調べることは、筆者らは語学学習時によく利用してきており、それは他の語学学習者も同様であろう。しかし、結果はあくまで検索結果であるため、結果をざっと見ることにより頭の中で集計を行ってきていた。これを解決するシステムを語学学習者としては作ってみたかった、というのが研究の端的な動機となっている。本用例検索は、語学学習者への応用のみならず、かな漢字変換や、自動翻訳といった、自然言語で多くの用例を必要とするシステムへの応用が考えられる。

本研究は未だ成文を抽出するといったところまではいっていないが、その第一歩の研究であると位置づけることもできる。

## 6 結論

本稿では、検索エンジンを利用した語の用例検索システム kiwi について報告した。kiwi はユーザが語を入力すると、その語を検索エンジンに問い合わせ、検索結果を集計することにより、用例をユーザに提示する。

kiwi の特徴は集計に必要な言語の解析処理を言語に非依存のものにしている点にある。これは、言語のデータベースを特定する従来の辞書や KWIC とは大きく異なる点である。この特徴から、システムは多言語に応用することができ、言語の解析技術や辞書が十分に整備されていない言語であっても、web 上にデータさえあれば、用例を調べることができる。また、データはすべて動的に採取するため、生きた用例を調査することができるという別の特徴もある。

実際にシステムの構成を論じた後、評価を行った。それによると、定型熟語などは 88.5%の割合で上位 10 位以内に用例が挙がった。また、生きた用例も得ることができ、現代に特徴的な用法を見ることができた。

今後は、有効な用例の絞込みに焦点を当てると共に、システムの公開を目指したい。また、自動翻訳といった複合システムの用例の動的な収集に応用を考えていきたい。

## 参考文献


[1] S. et al. Ananiadou. C-value. *hahaha*, 1996.

[2] E. Brill and J. et al. Lin. Data-intensive question answering. *Proceedings of TREC*, 2001.

[3] U. Manber and G. Myers. Suffix arrays: a new method for on-line string searches. *SIAM Journal of Computing*, 1993.

[4] many people. *Longman Dictionary of Contemporary English*. Peason Education, 2002.

[5] M. Nagata, Saito T., and Suzuki K. Using the web as a bilingual dictionary. *ACL workshop DDMT*, 2001.

[6] M. Utiyama and H. Isahara. Tools for exploring natural language. *NLPRS*, 2001.

[7] Webcorp. Webcorp home page, 1999.

[8] 久松健一. **フランス語重要表現・熟語集**. 駿河台出版社, 2001.

[9] 三省堂編修所. **三省堂実用ことわざの辞典**. 株式会社三省堂, 2002.

[10] 神部孝. *TOEFL* **英熟語** *850*. 株式会社旺文社, 2001.