高性能計算環境を用いたWebからの大規模格フレーム構築

河原 大輔　黒橋 禎夫
東京大学 大学院情報理工学系研究科
〒 113-8656 東京都文京区本郷 7-3-1
{kawahara,kuro}@kc.t.u-tokyo.ac.jp

あらまし

本稿では、高性能計算環境を利用して、Web から大規模テキストコーパスを抽出し、格フレームを構築する方法について述べる。格フレームは人間のもっている常識的な知識のうちもっとも基本的なものであり、これを自動構築するには大規模かつ偏りのないテキストが必要となる。そこで、Web から日本語文を抽出することによって大規模コーパスを作成し、それを用いて格フレームを構築するということを行う。約 4 億 Web ページから約 5 億文からなるテキストコーパスを作成し、さらにこのコーパスから約 9 万用言からなる格フレームを構築した。これらのプロセスは、巨大なデータを扱うため 1 つの計算機で行えば数年を要し現実的ではないことから、約 350CPU からなる高性能計算環境を利用することによって実現した。

キーワード　格フレーム, Web, 高性能計算環境

Case Frame Compilation from the Web using High-Performance Computing

Daisuke Kawahara　Sadao Kurohashi
Graduate School of Information Science and Technology, University of Tokyo
Hongo Bunkyo-ku, Tokyo 113-8656, JAPAN
{kawahara,kuro}@kc.t.u-tokyo.ac.jp

Abstract

This paper describes a method of constructing a wide-coverage case frames from the Web. To obtain such knowledge, an enormous amount of balanced corpus is required. We consider the Web as a balanced corpus, and first build a huge text corpus from the Web. We then construct case frames from the corpus. It is infeasible to do these processes by one CPU, and so we employ a high-performance computing environment. The acquired corpus and case frames are extremely larger than previously built corpora and case frames. The resultant case frames contain most examples of usual use, and are ready to be applied to lots of NLP applications.

Key Words　case frame, Web, High-Performance Computing

1 はじめに

計算機で文章を理解するためには、少なくとも、文章においてどの単語とどの単語がどのような関係をもっているかを明らかにする必要がある。このような単語間の関連性を解析するためには、人間がもっている常識のような幅広い知識が必要となる。

我々は、そのような知識のうちもっとも基本的な「格フレーム」をコーパスから自動構築してきた [3,8]。格フレームとは、用言とそれに関係する名詞を集めたものであり、例えば「積む」という用言の格フレームのひとつとして次のようなものが考えられる。

{ 従業員, 運転手, ···} が { 車, トラック,···} に { 荷物, 物資 } を 積む

このような格フレームは、構文・格・省略解析のような文章中の要素間の関連性解析から検索、要約、翻訳のような言語処理アプリケーションまで広く応用できると考えられる。例えば、次のような構文的曖昧性の解決に格フレームが利用できる。

- 望遠鏡で泳ぐ女の子を見た
- クロールで泳ぐ女の子を見た

これらの例において「望遠鏡で」「クロールで」はそれぞれ、「泳ぐ」「見た」の係り先の曖昧性がある。「クロールで 泳ぐ」「望遠鏡で 見る」という知識が格フレームにあれば、それぞれの構造が正しく解釈できる。しかし、新聞コーパスから構築した従来の格フレームには、これらの格フレームは含まれていない。新聞には社会・政治・経済ドメインへの偏りがあるため、このような日常的な知識は新聞にほとんど書かれていないため集まらないと考えられる。

このような問題を解決するには、より偏りなく網羅的な格フレームを獲得する必要がある。このためには、大規模な balanced corpus を用いるのが望ましい。balanced corpus としては、国立国語研究所が作成を試みているものなどがあるが [11]、それらはそもそも大規模なものを指向したものではないために格フレーム構築には適さない[1]。一方で、インターネットの爆発的な普及に伴い、様々なテキストが Web 上に存在するようになった。そこで、Web 上のテキストを一種の balanced corpus とみなし、そこから格フレームを構築するということを行う。

[1]英語では、Brown Corpus [1] や British National Corpus [6] など有名な balanced corpus があるが、大きくても 100M words であり、それほど大規模なものではない。

2 Web コーパスの作成

まず、Web から日本語文を収集しコーパスを作成する。これを Web コーパスと呼ぶ。本節では Web コーパスの作成方法について説明する。

2.1 Web ページの収集

Web ページの収集には、東京大学田浦研究室で開発されたクローラを用いた [7]。このクローラは、日本語ページの収集を指向したもので、本研究の目的に適している。収集の結果、約 4 億 Web ページが得られた。

2.2 Web ページから日本語文の抽出

各 Web ページを以下のように処理し、日本語文を抽出する。

1. エンコーディング情報を用いた日本語ページ候補の抽出
   (a) ページに charset 情報[2]が明示されており、それが日本語を表すために使われるエンコーディング (euc-jp, x-euc-jp, iso-2022-jp, shift_jis, windows-932, x-sjis, shift-jp, utf-8) であれば、そのページを選択する。utf-8 以外は日本語固有のエンコーディングであるが、utf-8 は Unicode のエンコーディング形式であり他言語も含まれる。utf-8 のページはここで選択しておき、2 の処理で日本語のページのみを選択する。
   (b) charset 情報が明示されていなければ、perl の Encode::guess_encoding() 関数[3]を用いてエンコーディングを推定する。この関数は、各エンコーディングの特徴的なバイト列を手がかりにしてエンコーディングを判定している。エンコーディングが euc-jp, shiftjis, 7bit-jis, utf8 のいずれかに判定されれば、そのページを選択する。
2. 言語情報を用いた日本語ページ判定

   1 の抽出において、明示されているエンコーディングが誤っている場合や utf-8 の場合、明

[2]HTML の meta タグ中に記述されている charset 属性

[3]perl 5.8 以上に同梱されている Encode モジュールに含まれている。

示されていないときの perl による自動判定が誤っている場合に、日本語ではないページが抽出される可能性がある。そこで、日本語の助詞の含有率を用いて日本語のページかどうかをチェックする。以下に示す助詞の文字を 0.5%以上含むページのみ用いる。

が, を, に, は, の, で

この結果、日本語と判定された約 1 億ページを得た。

3. ページからの文抽出

   ページの HTML をパースし、HTML タグと句点を利用して文のリストを得る。HTML タグとしては、例えば br, p などの改行、段落を表すタグを文区切りと認識する。pre タグは、その中のテキストをフォーマットしないことを示すものなので、その中にある改行は文区切りと認識する。残りの部分は句点で分割する。

4. 日本語文の抽出

   日本語ページと判定されていても、文ごとに見ると英語の場合もあるので、日本語文のみを抽出する必要がある。ひらがな、カタカナ、漢字のいずれかが 60%以上含まれる文のみを抽出する。

   得られた文集合には、ミラーサイトなどから同じ文が抽出されている場合もあるので、重複している文を除く。

上記の処理の結果、約 5 億日本語文からなるコーパスを作成した。その一部を表 1 に示す。

得られた Web コーパスの質を調べるために、得られた文が日本語の文であるかどうかを調査した。Web コーパスからランダムに 1,000 文を抽出し、日本語文であるかどうかを人手でチェックしたところ、995 文が日本語文であった。これより、良質な日本語コーパスが構築できたといえる。

さらに、得られた Web コーパスの特性を次のようにして調査した。まず、コーパスに含まれる異なり単語数 (未知語を除く) が、抽出した日本語の文数の増加によってどのように変化するかを調べた。これを Web コーパスと新聞コーパスについて行った結果を図 1 に示す。同じコーパスサイズにおいても、Web コーパスの方が異なり単語数が多いことから、新聞コーパスより幅広い表現を含むことがわかる。図 1 には、頻度 5 以上の異なり単語数も示している。1000 万文程度では、Web コーパスでは低頻度語が多いために新聞とほとんど差はないが、コーパスサイズが大きくなるにつれて、新聞よりも大幅に増えていることがわかる。

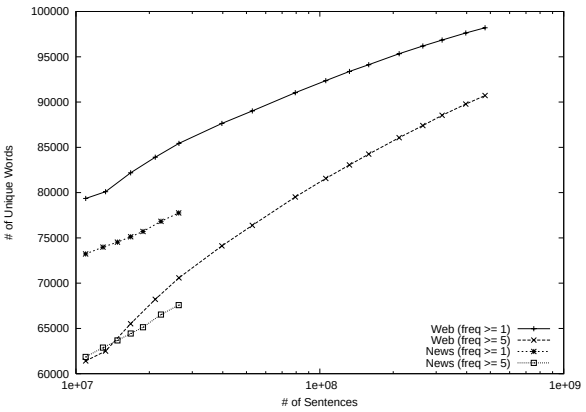

図 1: コーパスサイズと異なり単語数の関係

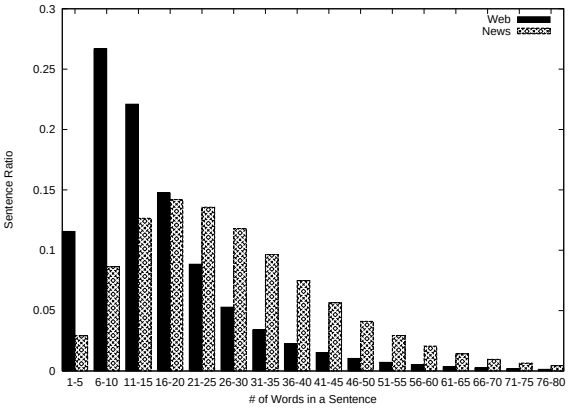

図 2: 文の長さの分布

また、文の長さを文に含まれる単語の数とし、文の長さの分布を調べた。その結果を図 2 に示す。これより、Web コーパスは、新聞コーパスと比べて短い文が多いことわかる。

3 格フレーム構築

格フレームは [8] の手法を用いて自動構築を行う。本節では、格フレーム構築手法の概要を述べる。

人間のもつ常識的知識の重要な部分である格フレームは、様々な言語現象をカバーすることが望ましい。そのような格フレームを構築するために、大

表 1: 作成した Web コーパスの一部

しょうがないので駅のレストランで食事をしようとした所、１日数本しかない山田線の存在に思い当たる。
もれなくプレゼント！
でも僕はＴシャツの上に長袖のシャツ。
今回は某アイドルの高橋一也も参加したので客が若い。
団体Aが「まちづくり」をテーマにインターネット上で公開講座を開催しようとしている。
ｈｔａｃｃｅｓｓを置いたとたんそのディレクトリ以下で.
昨年の没後４００年祭を機に復元した井戸を紹介する木下さん
恋は、真剣勝負。
ほめ言葉が多くって嬉しいですね。
開校式並びに入学式を挙行、初代校長佐治勝弥、職員１０名沖館小学校校舎一部を併用す。
いまだに言うでしょう。
「買いパラ」を見たと伝えれば、お買い上げ合計金額より５%引きいたします。
政治も危機的状況ですし、物資も不足しています。
そういう長期的な存在理由とか、長期的なビジョンとか、何故ここが国のお金で、我々の税金でやらなければいけないのか、その辺を評価する上で何かお考えになられていますでしょうか。
河北郡津幡町南中条・バリアフリー対応の学校案内や生徒会活動の紹介など
思いやりのある優しい子に育ってネ。
工学的諸問題に対処する際に必要な，線形代数・解析・確率・統計などの数学に関する知識を理解できること.
昔は，秋田の海で，猫もまたいで通る位，沢山とれた。
毎月の費用もわずかです！
「いつもながら気合いが入ってますけど、最近もレースは結構出てるんですか？」
合成化学者の間だけで埋没している物質群の掘り起こしと、物性科学の最新の成果による新たな分子設計指針の提案は緊急課題である。
腐女子の行く道、萌える道：ニュースを取り上げてみる非常に気になるコメントです。
読み終わったあと、とにかくこわいという感情と、こんな理不尽が許されているショックと、その中でも生きている人たちのすごさが強烈に残った。
工事の実施個所を知りたい方はこちらへ
動物より遅い人間チームが４１もあったなんて、情けないぞ。
と言うことは、ゼロ金利解除時期についての思惑が大きく振れることはできれば避けなければならない、ということにもなると思います。
ダライ・ラマが語る―母なる地球の子どもたちへ

規模コーパスから漸進的に確からしい情報を抽出する。

まず最初に、大規模コーパスを構文解析し、その解析結果から第 1 段階の格フレームを構築する (図 3)。格フレームを構築する際の最大の問題は、用言の用法の曖昧性である。つまり、同じ表記の用言でも複数の意味、用法をもち、とりうる格や用例が異なる。例えば、以下の 2 つの例は、用言は「積む」で同じであるが用法が異なっている。

(1) a. トラックに荷物を積む
b. 経験を積む

用法が異なる格フレームを別々につくるために、我々は、格フレーム収集の単位を用言とその直前の格要素の組とした。「積む」の例では、「荷物を積む」「経験を積む」を単位として格フレームを収集する。さらに、「荷物を積む」「物資を 積む」などかなり類似している格フレームをマージするためにクラスタリングを行う。

上記の第 1 段階の構築手法では構文解析を用いているために、基本的に格助詞の付属している格要素を収集している。このため、得られる格フレームは、二重主語構文、外の関係、格変化のような複雑な言語現象には対処できないという問題がある。この問題に対処するために、上記で得られた格フレームを用いて再度テキストを解析し、新たな情報を格フレームに与える。新たに得られる情報は、1 回目の格フレーム構築では扱うことができなかった係助詞句 (「〜は」や「〜も」) や被連体修飾詞に関する関係である。

(2) この車はエンジンが良い

例えば、上例において、構文解析の段階では「車は」は解釈できなかったが、格解析では「{ エンジン } が よい」という格フレームを用いることによって、格フレームにガ格以外の格がないことから「車は」は 2 つ目のガ格であり、「{ エンジン } が よい」は二重主語構文をとることがわかる。

(3) その問題は彼が図書館で調べている

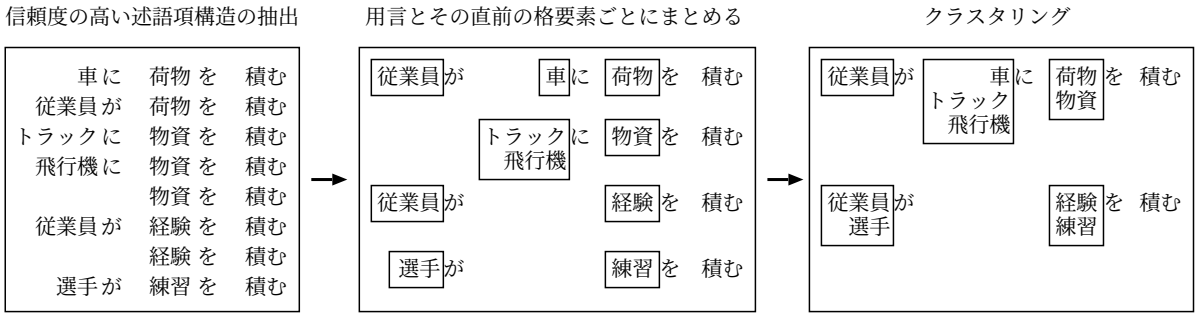

図 3: 格フレーム構築の概要

この例文の「問題は」は、すでに得られている格フレーム「{ 問題, 課題 } を { 図書館 } で 調べる」のヲ格の用例群に合致するため、格解析ではヲ格と解析されるだけで、新しい情報は得られない。同様に、被連体修飾詞は構文解析では扱われないが、格解析では、格フレームのガ格、ヲ格などの用例と類似しているかどうか調べることによって解釈される。例えば、「業務を営む免許」の「免許」は、格フレーム「{ 銀行, 会社 } が { 業務, ビジネス } を 営む」のどの格の用例とも類似せず、外の関係と呼ばれる関係をもっていると判定され、この情報が格フレームに加えられる。

4 高性能計算機グリッドを用いた格フレーム構築

格フレームを構築のために、まず、Web コーパスを形態素・構文解析する。形態素・構文解析には JUMAN, KNP を用いる。JUMAN, KNP による形態素・構文解析の解析速度は約 20 文/s であり、約 5 億文の解析には約 10 カ月かかることになる。これは非現実的であるので、高性能計算機グリッドを用いて解析を行った。この処理は、Web コーパスを約 1 万個に分割し、グリッド環境用シェル GXP[4] [2] を用いて約 350CPU の計算機グリッドにこれらの解析ジョブを投入することによって行った。その結果、約 1 日間で解析を終えることができた。

次に、Web コーパスの解析結果から格フレームを構築した。解析結果を用言ごとに分割することによって約 9 万個のデータを得、同グリッドを用いてそれぞれから格フレームの構築を行った。この処理には約 7 日間かかった。

[4] http://www.logos.ic.i.u-tokyo.ac.jp/phoenix/

表 2: 構築した格フレームの統計情報

		Web	新聞
用言数		89243	18246
(内訳)	動詞	40860	12641
	形容詞	4121	991
	名詞+判定詞	44262	4614
用言あたりの平均格フレーム数		34.3	17.5
格フレームあたりの格の平均数		3.2	2.4
格あたりの平均用例数		72.9	29.8
格あたりの平均異なり用例数		26.9	4.2

表 4: 格フレームの有無

	一致	類似	なし
Web 格フレーム	98	26	7
新聞格フレーム	54	57	20

5 Web 格フレーム構築結果

構築した格フレームの統計情報を表 2 に示す。比較のために、新聞記事 26 年分 (2600 万文) から構築した格フレームの統計情報も示す。また、構築した格フレームの例を表 3 に挙げる。新聞格フレームに対応するものがあれば併記してある。

得られた格フレームのカバレージを調べるために、普通の文に対して格フレームが収集されているかどうかをチェックした。解析対象文として、小学生用の国語辞典である例解小学国語辞典 [9] の定義文からランダムに 100 文を抽出したものを用いた。そのうち直前格要素をもつ用言は 131 個あり、これらに対応する格フレームが構築できているかどうかを調べた。その結果を表 4 に示す。表に示すよう

表 3: 構築した格フレームの例

泳ぐ { イルカ, 生, 魚, ..} が { 海, 水中, 海中, ..} を { クロール, 平泳ぎ, バタフライ, ..} で

寝そべる { 人, 男 } が { ビーチ, 砂浜, 浜辺, ..} に

磨く { 私, 男性, 人, ..} が { ブラシ, 所, トイレ, ..} で { 歯, 奥歯, 前歯 } を
⇔ 新聞: { 人, イヌ, 園児, ..} が { 歯 } を 磨く

煎る { 母 } が { 豆, 大豆, ごま, ..} を { 火, 強火, 中火, ..} で
⇔ 新聞: { 豆 } を 煎る

録画する { 旦那, 妹, 知人 } が { 番組, 放送, 特番, ..} を { ビデオ, ディスク, テープ, ..} に { 標準, ビデオ, モード } で
⇔ 新聞: { 番組, 放送 } を 録画する

プレイする { 人, 自分, 私, ..} が { 人, モード, キャラクター, ..} で { ゲーム, 版, シナリオ, ..} を

インストールする { あなた, 学生, 者, ..} が { ソフト, ドライバ, ソフトウェア } を { コンピュータ, パソコン, マシン } に { 手順, 接続, 方法, ..} で { CD, ROM, コレクション, ..} から

フィルタリングする { 上, ML, 元 } の { メール, データ, ファイル, ..} を

逼迫する { 国, 市, 都, ..} の { 財政, 予算, 会計 } が

に、そのうち 98 個は対応する格フレームがあり、26 個は非常に似ている (シソーラスで同ノード) 格フレームが存在していた。これを新聞格フレームでも調査したところ、54 個は対応する格フレームがあり、57 個は非常に似ている格フレームが存在していた。これらより Web 格フレームはカバレージが高いことがわかる。

6 関連研究

Web からテキストコーパスを抽出している研究として、関口らによる研究がある [10]。関口らは、HTML タグや文字種の情報を用いて良質の日本語文を抽出し、Web から 220MB のテキストコーパスを構築している。構築したコーパスを用いて格フレームを構築するなどの評価実験を行い、新聞コーパスよりも大きなコーパスを構築できたと主張している。このコーパスの規模は、新聞記事 10 年分程度の大きさであると推測でき、本研究で構築した Web コーパスと比べると約 1/50 であり、はるかに小さい。

一方、サーチエンジンを用いて単語や句の頻度を調べ、それを様々な NLP のタスクに利用する研究が近年活発に行われている。例えば、Lapata らはサーチエンジン AltaVista を用いて単語 n-gram の頻度を取得し、訳語選択、スペル訂正および複合名詞解析など様々な NLP タスクに用いている [4,5]。訳語選択などいくつかのタスクについては、既存のテキストコーパスを用いた手法より精度がよいが、その他のタスクについてはあまり精度はよくないことを述べている。その原因としては、n-gram の頻度情報のみしか用いることができず、品詞や構文情報などの言語情報が利用できないことを挙げている。これに対して、我々の作成した Web コーパスは、様々な言語解析技術を適用することができ、様々な NLP タスクにおける利用が期待される。

7 おわりに

本稿では、高性能計算環境を利用して、Web から大規模テキストコーパスを抽出し、格フレームを構築する方法について述べた。コーパスとしては、良質の約 5 億日本語文からなるものを構築するこ

とができた。そのコーパスから格フレームを自動構築し、新聞コーパスから構築したものよりもカバレージが高く、常識的な知識を多く含む格フレームとなった。構築した大規模格フレームは、構文・格・省略解析のような文章中の要素間の関連性解析から検索、要約、翻訳のような言語処理アプリケーションまで広く利用することができる。

謝辞

Web データ、計算機グリッドの使用許可を下さった東京大学大学院情報理工学系研究科の田浦健次朗助教授に感謝いたします。

参考文献

[1] Winthrop Nelson Francis and Henry Kucera. *Frequency Analysis of English Usage: Lexicon and Grammar*. Houghton Mifflin, 1982.

[2] Kenji Kaneda, Kenjiro Taura, and Akinori Yonezawa. Virtual private grid: A command shell for utilizing hundreds of machines efficiently. In *2nd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid 2002)*, 2002.

[3] Daisuke Kawahara and Sadao Kurohashi. Fertilization of case frame dictionary for robust Japanese case analysis. In *Proceedings of the 19th International Conference on Computational Linguistics*, pp. 425–431, 2002.

[4] Frank Keller and Mirella Lapata. Using the web to obtain frequencies for unseen bigrams. *Computational Linguistics*, Vol. 29, No. 3, pp. 459–484, 2003.

[5] Mirella Lapata and Frank Keller. The web as a baseline: Evaluating the performance of unsupervised web-based models for a range of NLP tasks. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 121–128, 2004.

[6] Geoffrey Leech. 100 million words of English: the British National Corpus. *Language Research*, Vol. 28, No. 1, pp. 1–13, 1992.

[7] Toshiyuki Takahashi, Hong Soonsang, Kenjiro Taura, and Akinori Yonezawa. World wide web crawler. In *Poster Proceedings of the 11th International World Wide Web Conference*, 2002.

[8] 河原大輔, 黒橋禎夫. 格フレーム辞書の漸次的自動構築. 自然言語処理, Vol. 12, No. 2, pp. 109–132, 2005.

[9] 田近洵一（編）. 例解小学国語辞典. 三省堂, 1997.

[10] 関口洋一, 山本和英. Web コーパスの提案. 情報処理学会 自然言語処理研究会 2003-NL-157, pp. 123–130, 2003.

[11] 柏野和佳子, 丸山岳彦, 稲益佐知子, 茂木俊伸. 語の出現分布からみた月刊雑誌と新聞コーパスの特性調査-用例収集資料としての多様性の検討-. 言語処理学会 第 11 回年次大会発表論文集, pp. 380–383, 2005.