

## $n$ グラム統計によるコーパスからの未知語抽出

森 信介 長尾 眞

京都大学工学研究科 電子通信工学専攻

〒606-01 京都市左京区吉田本町

{mori,nagao}@kuee.kyoto-u.ac.jp

あらまし

自然言語処理において、辞書は単語の文法的機能や意味の情報源として必要不可欠であり、辞書に登録されていない単語を減少させるため、辞書の語彙を増強する努力がなされている。新語や専門用語は絶えず増え続けているため、辞書作成の作業は多大な労力を要するのみならず、各解析段階での未知語との遭遇は避けられず、大きな問題の一つとなっている。この問題を解決するため、本論文では、 $n$  グラム統計を用いて、コーパスからの単語の抽出とその単語が属する品詞の推定を同時に行なう方法を提案する。この方法は、同一品詞に属する単語の前後に位置する文字列の分布は類似するという仮定に基づく。実験の結果、本手法が未知語の品詞推定や辞書構築に有効であることが確認された。

キーワード 未知語 品詞 辞書 コーパス  $n$  グラム統計

## Unknown Word Extraction from Corpora Using $n$ -gram Statistics

Shinsuke Mori Makoto Nagao

Department of Electrical Engineering, Kyoto University

Yoshida-honmachi, Sakyo, Kyoto, 606-01 Japan

{mori,nagao}@kuee.kyoto-u.ac.jp

Abstract

Dictionaries are indispensable for NLP as a source of information of grammatical functions or meanings of words. Much endeavor is being made to reinforce their vocabulary. Given continuous increase of new words or technical terms, building a dictionary takes vast effort and unknown words are inevitable at any step of analysis and this causes a grand problem. To solve this problem, we propose a method to extract words from a corpus and estimate part-of-speeches(POSs) which they belong to simultaneously using  $n$ -gram statistics, based on the supposition that distributions of strings preceding or following words belonging to the same POS are similar. Experiments have shown that this method is effective to infer the POS of unknown words and build a dictionary.

key words Unknown Word, Part-of-speech, Dictionary, Corpus,  $n$ -gram statistics

## 1 はじめに

自然言語処理に必須のデータのの一つに辞書が挙げられる。このため、今日まで辞書の構築に多くの努力がなされてきた。しかし、自然言語処理が多方面で応用される現在、各分野特有の専門用語や、絶えまなく増加する新語を人手で登録していくことには限界がある。このため、各解析段階で未知語との遭遇は避けられず、解析誤りの一因となっている。形態素解析では、未知語の多くは名詞であるから、文字種などの情報をもとにして形態素に分割し、名詞として解析しておくことで問題なく解析できることが多い。しかし、複数の文字種で構成される未知語や、名詞以外の品詞に属する未知語も存在するため、より高い精度の解析を目指す上で障害となっている。以上の議論を踏まえると、未知語をコーパス等から自動的に取り出す必要があると結論できる。

長尾と森<sup>(1)</sup>は、頻度と前後の文字列のばらつきに基づいて語句を抽出する研究を行なった。他に、定型表現を取り出す研究などがなされている<sup>(2) (3) (4) (5) (6)</sup>。これらの研究は、取り出した文字列に対して、再現率や正解率を計算するなどの、定量的評価を行なっていない点で問題である。また、取り出した文字列を辞書に登録し、実際の解析に用いるためには、まず品詞を推定する必要がある。しかし、大規模なコーパスからの単語の獲得とその品詞推定を行なった研究は、我々が知る限りにおいて全くない。

このような観点から、本論文では、コーパスからの単語の抽出とその品詞の推定を同時に行なう方法を提案し、この方法が有効であることを実験的に示す。以下の節では、本手法の基礎となる統計的仮定に対する理論的考察、単語の抽出と品詞推定の具体的方法、実験の結果とその評価について順に述べ、最後に結論を述べる。

## 2 仮定とその理論的考察

本節では、まずコーパス中に出現する文字列の「環境」を定義し、その意味を説明する。次に、本論文で提案する単語抽出の手法の基礎となる仮定について理論的考察を行なう。

### 2.1 文字列のコーパス中の環境

本論文では、ある文字列のコーパス中の環境を、その文字列の前後の文字列のコーパス中での条件付確率の確率分布と定義する。ここで、前後の文字列は、ある特定の長さの文字列、あるいは特定の個数の形態素列がなす文字列であるとす。先行する文字列の確率分布を左確率分布と呼び、後続する先行する文字列の確率分布を右確率分布と

頻度	確率	文字	文字	頻度	確率	
13	6.8%	、	楽し	い	16	8.3%
6	3.1%	。		か	2	1.0%
2	1.0%	う		く	3	1.6%
13	6.8%	が		げ	4	2.1%
10	5.2%	て		さ	8	4.2%
8	4.2%	で		そ	10	5.2%
4	2.1%	と		て	1	0.5%
1	0.5%	ど		ま	7	3.6%
2	1.0%	な		み	43	22.4%
14	7.3%	に		む	38	19.8%
19	9.9%	の		め	16	8.3%
4	2.1%	は		も	4	2.1%
7	3.6%	も		ん	40	20.8%
2	1.0%	ら				
2	1.0%	り				
1	0.5%	ろ				
82	42.7%	を				
1	0.5%	折				
1	0.5%	変				
		交				

(頻度:192)

図 1: 文字列「楽し」の環境

呼ぶ。例えば、文字列「楽し」の EDR コーパスにおける環境は図 1 のようになる。この図の左側は、文字列「楽し」に先行する 1 文字の頻度と条件付確率 (左確率分布) であり、右側は後続する 1 文字の頻度と条件付確率 (右確率分布) である。一般に離散確率分布はベクトルと見なすことができるので、左確率分布ベクトルと右確率分布ベクトルを順に並べた結果もベクトルとなる。これが環境であり、以下では  $D$  で表わす。

以上に述べた環境は、各品詞が出現することができるパターン集合とみなすことができ、それぞれの品詞の文法的な特徴を表わしていると考えられる。

### 2.2 文字列の環境に対する仮定

一般に、ある文字列  $\alpha$  がある品詞  $pos$  に属する単語であるとき、この文字列のコーパス中の環境  $D(\alpha)$  は、その品詞の環境  $D(pos)$  に類似すると考えられる。文字列が複数の品詞に属することもあるので、より正確には、文字列がある品詞として現れる確率とそれぞれの品詞の環境の積の総和が、文字列の環境に類似すると考えられる。よって、文字列  $\alpha$  がある品詞  $pos_k$  に属する確率を  $p(pos_k|\alpha)$  とし、品詞  $pos_k$  の環境を  $D(pos_k)$  とすると以下の式が成り立つと考えられる<sup>1</sup>。

$$D(\alpha) \approx \sum_k p(pos_k|\alpha) D(pos_k) \quad (1)$$

この式で、総和は考慮の対象となる全ての品詞に対して計算される。例えば、文字列「楽し」がコーパス中で形容詞

<sup>1</sup> 活用語は終止形ではなく語幹を形態素とする。

と動詞としてそれぞれ確率  $p(\text{形容詞}|\text{楽し})$ ,  $p(\text{動詞}|\text{楽し})$  で用いられているとすると、文字列「楽し」の環境は以下の式を満たすと考えられる。

$D(\text{楽し})$

$\approx p(\text{形容詞}|\text{楽し})D(\text{形容詞}) + p(\text{動詞}|\text{楽し})D(\text{動詞})$   
 コーパスに対する統計結果に式 (1) を実際に適用しても、一般に独立変数  $p(\text{pos}_k|\alpha)$  の数は、確率分布ベクトル  $D$  の次元、すなわち、等式の数より少ないため、連立方程式と見なして解いても解が得られるとは限らない。また、環境はコーパスに対する統計により得られるものであり、理想的な値を得ることができないことにも注意しなければならない。よって、式 (1) の左辺と右辺がある基準で最も類似する  $p(\text{pos}_k|\alpha)$  の組を求めることが問題となる。この類似度の基準としては、ベクトル間のユークリッド距離を用いることとする。以上の議論から、この問題は、文字列が各品詞として用いられる確率の確率分布ベクトル  $p$  を決定変数とし、以下で定義される目的関数  $F$  を最小化する最適化の問題となる。

$$F(p) = |D(\alpha) - \sum_k p_k D(\text{pos}_k)|^2 \quad (2)$$

決定変数ベクトル  $p$  は品詞の数を  $n$  とし、 $p_k = p(\text{pos}_k|\alpha)$  として、以下のように定義される。

$$p = (p_1, p_2, \dots, p_n)$$

また、 $p$  の要素は確率であるから、可能領域  $V$  は以下のようになる。

$$V = \{p \mid 0 \leq p_k \leq 1, \sum_k p_k = 1\} \quad (3)$$

さらに、 $F(p)$  の最小値は、文字列  $\alpha$  の環境  $D(\alpha)$  がいくつかの品詞の環境に分解できる場合、すなわち文字列  $\alpha$  がいくつかの品詞に属する場合、比較的小さい値となり、分解できない場合は比較的大きな値となる。全ての単語はある品詞に属するので、このことを考え合わせると、 $F(p)$  の最小値は文字列が単語である度合を表わすと考えられる。

以上に述べたことをまとめると、以下の最適化問題を解くことで、文字列がそれぞれの品詞に属する確率を求めることができる。

$$\min F(p) \quad \text{subject to } p \in V \quad (4)$$

このとき、 $F_{\min}(p)$  は文字列が単語である度合を表わす。

### 3 具体的な方法

本節では、前節で述べた仮定に基づいてコーパス中の任意の文字列に対して、単語である度合と各品詞に属する確

しかし	接続詞	へ	助詞
、	記号	の	助詞
元日	名詞	姿勢	名詞
の	助詞	を	助詞
紙面	名詞	示	動詞
は	助詞	す	語尾
新し	形容詞	年賀状	名詞
い	語尾	だ	助動詞
年	名詞	。	記号

図 2: EDR コーパスの一部

率を計算する方法を具体的に示す。

#### 3.1 各品詞の環境の計算

各品詞の環境は、形態素解析済みのコーパスに対して、その品詞に属する形態素の前後の文字列に対して統計をとることで求められる。本研究では、形態素解析済みのコーパスとして、EDR コーパス<sup>(7)</sup>を用いた。図 2 は、EDR コーパスの最初の一文である。以下では、この情報からの名詞の環境の計算を例として、各品詞の環境の計算手順を具体的に説明する。

各品詞の環境は、以下の処理を順に行なうことで計算する。

1. 左右の確率分布ベクトルの全ての成分を 0 とする。
2. 注目している品詞をコーパス中で順に探し、その品詞に対応する形態素の前後の文字列に対応する確率分布ベクトルの成分をインクリメントする。
3. 左右の確率分布ベクトルを注目している品詞の頻度で割る。

確率分布ベクトルは、実際にはかなり疎であり、各成分を指すキーは自然数とするより文字列とした方が実装上都合が良いので、これをハッシュを用いて実現した<sup>2</sup>。存在しないキーの値は 0 であるとしておく。

上述の手順を、図 2 のコーパスにおける名詞の前後一文字の環境の計算を例として、具体的に示す。ここで、左右の確率分布ベクトルを  $P_l, P_r$  とする。

1.  $P_l, P_r$  の全てのキーとその値をクリアする。
2. 図 2 の中で品詞が名詞である行を上から順に探す。
  - 3 行目：先行文字 = 「、」、後続文字 = 「の」  
 $P_l(、) = P_l(、) + 1, P_r(の) = P_r(の) + 1$
  - 5 行目：先行文字 = 「の」、後続文字 = 「は」  
 $P_l(の) = P_l(の) + 1, P_r(は) = P_r(は) + 1$

<sup>2</sup> 各成分を指すキーの数は有限であるから、有限次元のベクトルのキーである、1 から始まる連続する整数と対等であり、ハッシュはベクトルと見なせる。

頻度	確率	文字	文字	頻度	確率
1	20%	、	名詞	だ	1 20%
1	20%	い	の	の	1 20%
1	20%	す	は	は	1 20%
2	40%	の	へ	へ	1 20%
(頻度:5)			を	を	1 20%

図 3: 名詞の環境

- 以下、同様の処理を最後の名詞「年賀状」まで繰り返す。
3.  $P_l$  の全てのキーの値をその合計で割る。  
 $P_r$  の全てのキーの値をその合計で割る。  
この結果得られる環境を図 3 に示す。

### 3.2 コーパス中の文字列の環境の計算

コーパス中の文字列の環境は、文字列のコーパス中での頻度、すなわち  $n$  グラム統計を用いて計算される。例えば、文字列「楽し」の頻度  $f(\text{楽し})$  が 192 であり、文字列「楽しむ」の頻度  $f(\text{楽しむ})$  が 38 であれば、文字列「楽し」の右確率分布の「む」の値は以下のように計算される。

$$P_r(\text{む}) = P(\text{楽しむ}|\text{楽し}) = \frac{f(\text{楽しむ})}{f(\text{楽し})} = \frac{38}{192}$$

このように、左右の確率分布は  $n$ -gram と  $(n+1)$ -gram とから計算できる<sup>3</sup>。この  $n$  グラム統計には、長尾と森

- (1) が提案した手法を用いた。この方法では、任意の文字列に対して、そのコーパス中での全ての出現位置を連続的に取り出すことができるので、図 4 のようなコーパスのソート結果を得ることができる。各文字列の環境は、この結果を用いて、以下の処理を順に行なうことで計算する。
1. 左右の確率分布ベクトルの全ての成分を 0 とする。
2. コーパスのソート結果の中で注目している文字列が始まる位置を見つける。
3. ソート結果を順に見て行き、注目している文字列の前後の文字列に対応する確率分布ベクトルの成分をインクリメントする。
4. 左右の確率分布ベクトルを注目している文字列の頻度で割る。

前後の文字列を取り出す際に、文字数を指定する方法の場合は指定文字数を取り出し、形態素数を指定する方法の場合には、すでに計算済みの品詞の確率分布ベクトルのキーの中で最も長く一致する文字列とする。品詞の環境の計算と同様に、実装にはハッシュを用いた。

<sup>3</sup> より一般的には、 $n$ -gram と  $(n+k)$ -gram との比較である。

げながらのやりとりが 楽し い。いったい今、世界  
画をほうふつとさせて 楽し い。これほどさまにな  
とに交えていくのは、 楽し い。経済のしくみを大  
っている方がはるかに 楽し い。建具卸売業を経て  
食事をする時はとても 楽し い。今、特に子供が1  
ルだ。遊覧船の復活は 楽し い。水の上のにぎわい  
を店に支払う。踊って 楽し い。ただでなく、脚と足  
もかわならず、食事を 楽し いものと感じていなか  
とに、決まった筆者の 楽し いコラムが登場します  
。私は戦争のために、 楽し い温かい家庭生活を味  
市民にとって水族館は 楽し い教室であり、憩いの  
の大好きなお祭りだ。 楽し い催しがあれば、数多  
酒を飲んでもネアカな 楽し い酒だし、趣味も豊か  
日本には、まだ走って 楽し い道路がない。サイク  
した。山登りを趣味に 楽し い老後を過ごしている  
。炭鉱が消えてから、 楽し い話題が少なかった。  
分の陳列の場に戻る。 楽し かった2匹は、冬の氷  
相好が崩れ走っていて 楽し かったことの1つに、  
新し、消費者が気軽に 楽し く、新型車の情報を得  
年会を、ゆったりと、 楽し くやるコツを知ってい

図 4: コーパスのソート結果

上述の手順を、図 4 のコーパスのソート結果における文字列「楽し」の前後一文字の環境の計算を例として、具体的に示す。ここで、左右の確率分布ベクトルを  $P_l, P_r$  とする。

1.  $P_l, P_r$  の全てのキーとその値をクリアする。
2. 文字列「楽し」の開始位置を見つける (図 2 の 1 行目)。
3. 文字列が「楽し」でなくなるまで左右の文字列を調べる。
  - 1 行目: 先行文字 = 「が」、後続文字 = 「い」  
 $P_l(\text{が}) = P_l(\text{が}) + 1, P_r(\text{い}) = P_r(\text{い}) + 1$
  - 2 行目: 先行文字 = 「て」、後続文字 = 「い」  
 $P_l(\text{て}) = P_l(\text{て}) + 1, P_r(\text{い}) = P_r(\text{い}) + 1$
  - 以下、同様の処理を文字列「楽し」が続く限り返す。
4.  $P_l$  の全てのキーの値をその合計で割る。  
 $P_r$  の全てのキーの値をその合計で割る。

この結果得られる環境を図 1 に示す。なお、コーパス中の全ての文字列を対象とする場合は、ソートされたコーパスを順に見ていくだけでよく、注目している文字列の開始位置を見つける必要はない。

### 3.3 最適化問題の解法

各品詞の環境と文字列の環境が与えられると、式 (2) のパラメーターが決定され、式 (4) の最適化問題を実際に解くことで、文字列が各品詞に属する確率が実際に計算される。この最適化問題の解法は本研究とは直接関係がないので、以下に簡単に述べるにとどめる。

式 (2) から、この最適化問題の目的関数は、決定変数ベ

クトル  $p$  の凸関数である。また、式 (3) で与えられる可能領域も凸であるから、局所解が大域解となる。よって、可能領域中の任意の点から出発し、領域をはみ出ないように勾配ベクトルの可能領域に対する射影に沿って点を移動する。これを勾配ベクトルの射影が 0 ベクトルになるまで繰り返す。この結果、文字列が各品詞に属する確率のベクトル  $p$  と単語とみなせる度合  $F_{min}$  が求められる。

### 3.4 既知語と未知語の区別

コーパスから抽出された単語は、辞書に登録されているか否かで未知語と既知語に分けられる。よって、辞書として何を用いるかが問題となる。本研究では、一般に利用されている日本語形態素解析システム JUMAN<sup>(8)</sup> の辞書に登録されているか否かを未知語の基準とした<sup>4</sup>。

既知語と未知語を区別するために、抽出された文字列に活用語尾などを補って JUMAN で解析し、その結果、未知語を含んでいたり、記号や数字の連続となった場合、これを未知語として抽出する。

## 4 単語抽出実験の結果とその評価

前節で述べたように、各品詞の環境を得るために EDR コーパスを用いた。この EDR コーパスを生コーパスと見なして再現率と適合率を計算した。さらに、雑誌「日経サイエンス」に対しても抽出実験を行ない、既存の辞書に載っていない未知語の数を算出した。以下では、実験の条件、実験の結果、および結果に対する評価について述べる。

### 4.1 実験の条件

本研究では、未知語の抽出が目的であるから、未知語として出現する可能性が高い以下の品詞を対象とした<sup>5</sup>。

サ変名詞・非サ変名詞・ラ行五段活用動詞・形容詞・形容動詞

また、環境の計算の対象となる前後の文字列を平仮名連続と句読点に限り、左右の確率分布ベクトルの合計が、それぞれ 1 となるように正規化した。この目的は、環境の比較に要する計算時間を減少させることである。漢字や片仮名からなる文字列は、頻度が低いので結果にはほとんど影響しないと考えられる。さらに、2 文字以上で頻度が 10 以上の括弧などの記号を含まない文字列を抽出の対象とした。1 文字からなる未知語は、皆無と言ってよいであろう

<sup>4</sup> 辞書として新明解国語辞典第二版<sup>(9)</sup>を用いている。

<sup>5</sup> 上記以外の品詞を対象にすることもできるが、統計を用いているので、解析済みコーパスにある程度以上の頻度で出現していることが要求される。

表 1: EDR コーパスにおける再現率と適合率

閾値	再現率		適合率	
	のべ	異なり	のべ	異なり
0.10	26.2%	12.6%	96.8%	86.2%
0.15	46.4%	28.3%	94.0%	80.4%
0.20	61.8%	44.0%	90.7%	74.9%
0.25	73.2%	57.1%	87.4%	69.1%
0.30	79.8%	66.7%	—	—
0.35	84.4%	73.7%	—	—

から、再現率に対する影響はほとんどないと思われる。また、統計を用いているので頻度があまり低いと結果の信頼性が低くなる。

### 4.2 実験の結果

単語抽出の段階での精度を求めるために、形態素解析済みの EDR コーパスを生コーパスと見なして文字列の抽出実験を行なった。表 1 は、この結果から算出された各閾値における再現率と適合率である。ただし、平仮名のみで構成される文字列を評価の対象としていない。この点については次節で述べる。再現率と適合率の計算は、品詞と文字列の両方を区別して行なった。「のべ」の欄は頻度を加味した場合の結果であり、「異なり」の欄は頻度を加味しない場合の結果である。再現率の計算における頻度には、EDR コーパスから計算される頻度を用い、適合率の計算における頻度には、以下の式で計算される、本手法が推定した頻度を用いた。

$$p(pos|\alpha) \cdot f(\alpha)$$

この式で、 $f(\alpha)$  は文字列  $\alpha$  のコーパス中での頻度であり、 $p(pos|\alpha)$  は  $\alpha$  が品詞  $pos$  に属する推定確率である。文字列  $\alpha$  が、品詞  $pos$  に属するか否かは人手で判断した。適合率の計算は、推定確率が 0.1 以上の品詞を対象とした。この理由は、出力が大量であるため人手によるチェックが非常に困難な作業であることである。また、表 1 において、閾値 0.30 以上の適合率が算出されていないのも同様の理由による。実際、閾値 0.25 以下で、人手によるチェックの対象となった文字列と品詞の組合せは 7,367 個であった。

次に、抽出された文字列を 3.4 節で述べた方法を用いて既知語と未知語に区別した。この結果、日経サイエンス (1990 年分: 618,347 文字) から抽出された未知語は 268 個であった。表 2 は日経サイエンスから抽出された文字

表 2: 「日経サイエンス」から抽出された文字列の例

$F_{min}$	頻度	文字列	サ名 形容 形動 名詞 ラ行
0.04	115	自然	0.00 0.00 0.69 0.31 0.00
0.05	117	弱	0.00 0.96 0.00 0.00 0.04
0.05	179	十分	0.00 0.00 1.00 0.00 0.00
0.08	103	タンパク質*	0.00 0.00 0.00 1.00 0.00
0.11	11	LAN*	0.00 0.00 0.00 1.00 0.00
0.11	63	MHC分子*	0.00 0.00 0.00 1.00 0.00
0.16	16	吐き戻し*	0.19 0.00 0.00 0.81 0.00
0.17	33	人工現実感*	0.00 0.00 0.00 1.00 0.00
0.17	58	RU486*	0.00 0.01 0.00 0.90 0.09
0.18	40	キラル*	0.00 0.02 0.98 0.00 0.00

\* は未知語であることを表わす。

列の例である。

### 4.3 結果の評価

表 1 から、閾値が低いほど適合率が高いことが分かる。この結果は、2.2 節で述べた仮定から導かれる結果と一致する。また、出現頻度が高いほど推定が正確である傾向があった。これも、統計的手法を用いていることから自然に予測される。誤りの多くは、以下の 2 種類に属する。

- 平仮名のみからなる文字列の切り出し
- 形容動詞と名詞の識別

平仮名のみからなる文字列が誤りとなるのは、多くの場合単語の切り出しが不適切である場合であった。平仮名のみからなる文字列も評価の対象にした場合、表 1 の適合率は、それぞれ約 5% 程度低くなる。平仮名からなる新語や専門用語は少ないと考えられるので、これは大きな問題ではない。形容動詞と名詞の前後に位置する文字列には、他の品詞の組合せほどの差異がないため、名詞でしかない文字列が、低い確率ではあるが形容動詞でもあると推定される場合や、この逆の場合があった。この問題に対処するためには、双方の品詞に特有な語尾との共起をチェックすることが考えられる。

抽出された単語の中には、表 2 に例示したように、複数の文字種からなる文字列も多数あった。このような文字列も抽出できる点は、本手法の長所の一つである。もう一つの長所として、多品詞語であっても品詞が推定できる点が挙げられる。例として、表 2 中の「弱」は、「弱い」と「弱る」の語幹であることが挙げられる。それぞれの品詞の推定確率は、形態素解析や仮名漢字変換などの辞書の

拡張の際に、出力に対して順序を与えるためのコストなどの情報として有効である。本手法の短所は、頻度が低い単語の精度が低いことである。これは、統計的手法を用いる場合には避けられない問題である。

本手法を応用して辞書を増強する場合は、再現率と適合率の閾値や頻度との関係に注意し、目的に応じてこれらを設定することが重要であろう。

### 5 おわりに

本論文では、自然言語の各解析段階で問題となる未知語に対処するため、コーパスからの単語の抽出とその品詞の推定を同時に行なう方法を提案した。この方法は、同一品詞に属する単語の前後に位置する文字列は類似するという仮定に基づく。実験の結果、この方法が有効であることが確かめられた。本手法は未知語の辞書構築に非常に有効である。

### 参考文献

- (1) 長尾眞, 森信介. 大規模日本語テキストの n グラム統計の作り方と語句の自動抽出. 情報処理学会研究報告, pp. 1-8, 1993.
- (2) 新納浩幸, 井佐原均. 疑似 n グラムを用いた助詞的定型表現の自動抽出. 情報処理学会論文誌, Vol. 36, No. 1, pp. 32-40, 1995.
- (3) 北研二, 小倉健太郎, 森元つよし, 矢野米雄. 仕事量基準を用いたコーパスからの定型表現の自動抽出. 情報処理学会論文誌, Vol. 34, No. 9, pp. 1937-1943, 1993.
- (4) 新納浩幸. 文字列と後続文字との接続割合の変化を利用した定型的文末表現の自動抽出. 情報処理学会研究報告, pp. 39-46, 1994.
- (5) 池原悟, 白井 諭, 河岡 司. N-gram を用いた連鎖型共起表現の自動抽出法. 言語処理学会第 1 回年次大会発表論文集, pp. 313-316, 1995.
- (6) 浦谷則好. ニュース原稿データベースからの表現パターンの抽出. 情報処理学会第 50 回全国大会, 第 3 巻, 1995.
- (7) 日本電子化辞書研究所. EDR 電子化辞書仕様説明書, 1993.
- (8) 松本裕治, 黒橋禎夫, 宇津呂武仁, 妙木裕, 長尾眞. 日本語形態素解析システム JUMAN 使用説明書 version 2.0. 京都大学工学部長尾研究室, 1994.
- (9) 金田一京助, 柴田武, 山田明雄, 山田忠雄. 新明解国語辞典. 三省堂, 第二版, 1974.