Vol. 39 No. 7 情報処理学会論文誌 July 1998

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[††]

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ず，大きな問題の1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．

Unknown Word Extraction from Corpora Using n-gram Statistics

SHINSUKE MORI[†] and MAKOTO NAGAO[††]

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.

1. はじめに

自然言語処理に必須のデータの1つに辞書があげられる．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うえで障害となっている．以上の議論をふまえると，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[1]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った．ほかに，定型表現を取り出す研究などがなされている[2)~6)]．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った研究は，我々が知る限りにおいてまったくない．

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

本章では，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う．

† 日本アイ・ビー・エム株式会社東京基礎研究所
Tokyo Research Laboratory, IBM Research
†† 京都大学
Kyoto University

頻度	確率	文字	文字	頻度	確率
13	6.8%	、	楽し	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		（頻度：192）	

図1 文字列「楽し」の環境
Fig. 1 Environment of the string "楽し."

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る．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呼ぶ．たとえば，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 $\boldsymbol{D}$ で表す．

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の集合と見なすことができ，それぞれの品詞の文法的な特徴を表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 $\boldsymbol{\alpha}$ 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 $\boldsymbol{D}(\boldsymbol{\alpha})$ は，その品詞の環境 $\boldsymbol{D}(pos)$ 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 $\boldsymbol{\alpha}$ がある品詞 pos_k に属する確率を $p(pos_k|\boldsymbol{\alpha})$ とし，品詞 pos_k の環境を $\boldsymbol{D}(pos_k)$ とすると以下の式が成り立つと考えられる☆．

$$\boldsymbol{D}(\boldsymbol{\alpha}) \approx \sum_k p(pos_k|\boldsymbol{\alpha})\boldsymbol{D}(pos_k) \qquad (1)$$

この式で，総和は考慮の対象となるすべての品詞に対して計算される．たとえば，文字列「楽し」がコーパス中で形容詞と動詞としてそれぞれ確率 p(形容詞 | 楽し), p(動詞 | 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．

$$\begin{aligned}&\boldsymbol{D}(\text{楽し})\\ &\approx p(\text{形容詞}\,|\,\text{楽し})\boldsymbol{D}(\text{形容詞})\\ &\quad + p(\text{動詞}\,|\,\text{楽し})\boldsymbol{D}(\text{動詞})\end{aligned}$$

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 $p(pos_k|\boldsymbol{\alpha})$ の数は，確率分布ベクトル $\boldsymbol{D}$ 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 $p(pos_k|\boldsymbol{\alpha})$ 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 $\boldsymbol{p}$ 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．

$$F(\boldsymbol{p}) = \left| \boldsymbol{D}(\boldsymbol{\alpha}) - \sum_k p_k \boldsymbol{D}(pos_k) \right|^2 \qquad (2)$$

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

$$\boldsymbol{p} = (p_1, p_2, \ldots, p_n)$$

また，$\boldsymbol{p}$ 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．

$$V = \left\{ \boldsymbol{p} \,|\, 0 \le p_k \le 1, \sum_k p_k = 1 \right\} \qquad (3)$$

さらに，$F(\boldsymbol{p})$ の最小値は，文字列 α の環境 $\boldsymbol{D}(\alpha)$ 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 α 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．すべ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，$F(\boldsymbol{p})$ の最小値は文字列が単語である度合を表すと考えられる．

☆ 活用語は終止形ではなく語幹を形態素とする．

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
Fig. 2 An example of EDR corpus.

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．

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

このとき，$F_{\min}(\boldsymbol{p})$ は文字列が単語である度合を表す．

3. 具体的な方法

本章では，前章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[7]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うことで計算する．

(1) 左右の確率分布ベクトルのすべての成分を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．

上述の手順を，図2のコーパスにおける名詞の前後1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 のすべての成分を0とする．
(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」

$P_l(の) = P_l(の) + 1$, $P_r(は) = P_r(は) + 1$

- 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 のすべてのキーの値をその合計で割る．
P_r のすべ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．

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

図3 名詞の環境
Fig. 3 Environment of the noun.

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．たとえば，文字列「楽し」の頻度 f(楽し) が192であり，文字列「楽しむ」の頻度 f(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．

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

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☆．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すべ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うことで計算する．

(1) 左右の確率分布ベクトルのすべての成分を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いき，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

☆ 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
とに変えていくのは、 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 いだけでなく、脚と足
もかかわらず、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
Fig. 4 Result of sorting corpus.

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．

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1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 のすべ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」が続く限り繰り返す．

(4) P_l のすべてのキーの値をその合計で割る．
P_r のすべ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，コーパス中のすべ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，詳細は付録A.1に譲る．

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 $\boldsymbol{p}$ 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，領域をはみ出さ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 $\boldsymbol{p}$ と単語と見なせる度合 $F_{\min}$ が求められる．

3.4 既知語と未知語の区別

未知語抽出の目的は，ある自然言語処理システムの辞書の増強なので，抽出された文字列をその自然言語処理システムにとって未知語であるか否かを判別することが必要である．この際に注意しなければならないのは，登録語に分割されうる文字列の扱いである．本論文では，形態素解析を想定した場合の未知語の選別について述べる．形態素解析システムとしては，JUMAN[8]を用いて，これを基準として未知語の選別を行った．形態素解析の場合，抽出された文字列が単に登録語であるばかりでなく，既知語の連接となる場合も未知語として辞書に登録する必要はない．この区別を実現するために，抽出された文字列に推定された品詞の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こととした．

4. 単語抽出実験の結果とその評価

前章で述べたように，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☆．

☆ 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

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1文字と句読点に限り，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，皆無といってよいであろうから，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（74,525文；1,746,388形態素；2,709,007文字）を生コーパスと見なして文字列の抽出実験を行った．この結果から$F_{\min}$に対する様々な閾値に対して算出された再現率と適合率☆を**表1**に掲げる．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．

$$f(\boldsymbol{\alpha}, pos) = p(pos|\boldsymbol{\alpha}) \cdot f(\boldsymbol{\alpha})$$

この式で，$f(\boldsymbol{\alpha})$は文字列$\boldsymbol{\alpha}$のコーパス中での頻度であり，$p(pos|\boldsymbol{\alpha})$は$\boldsymbol{\alpha}$が品詞$pos$に属する推定確率である．文字列$\boldsymbol{\alpha}$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．

日経サイエンス（1990年分：618,347文字）についても同様の単語抽出の実験を行った．抽出された文字列と各品詞の確率を**表2**に掲げた．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，抽出された単語のうち，268個が未知語と判定された．

抽出された単語を辞書に登録することによる形態素解析の精度の変化を調べるために，単純マルコフモデルに基づく確率的形態素解析器をEDRコーパスから推定し，これらの単語を辞書に追加した．確率的形態素解析においては，各単語がそれぞれの品詞から生成される確率値を与える必要がある．この値は非常に重要であり，場合によっては解析精度が下がることがある．これは，コスト最小法を用いた場合も同様であり，単語を辞書に追加する際のコストによって精度が増減する．我々の提案する方法では，文字列が各品詞として用いられる確率が得られるので，以下の式のように，各単語の頻度を確率的形態素解析器の推定の対象となるコーパスにおける頻度に換算した値とすることで，その単語が学習コーパスに出現した場合の確率とともに形態素解析器の辞書に追加することができる．

$$f'(\boldsymbol{\alpha}, pos) = \frac{size_e}{size_s} f(\boldsymbol{\alpha}, pos)$$

ここで$size_e$と$size_s$はそれぞれEDRコーパスと日

表1 EDRコーパスにおける再現率と適合率
Table 1 Recall and precision on EDR corpus.

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%	—	—

表2 「日経サイエンス」から抽出された文字列の例
Table 2 Examples of extracted words from "Science."

$F_{\min}$	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す．

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．

☆ 再現率と適合率の定義は以下のとおり．

$$\text{再現率} = \frac{\text{正しく抽出された単語数}}{\text{抽出された文字列と品詞の直積の数}}$$

$$\text{適合率} = \frac{\text{正しく抽出された単語数}}{\text{テストコーパスに含まれる単語数}}$$

経サイエンスの単語数である．このようにして得られた形態素解析器で，日経サイエンスの記事の1つ「ガリウムヒ素半導体の進展」(1990年4月号，pp.80–90)を解析した．その結果，正解率（正しい分割と正しい品詞が与えられた単語の割合）は98.2%であり，抽出された未知語を辞書に登録しない場合の正解率95.9%を有意に上回った．これは，本手法の有効性を示す結果の1つである．

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出しが不適切である場合であった（この場合には推定された品詞にかかわらず誤りである）．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1つである．もう1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あげられる．例として，表2中の「弱」は，「弱い」と「弱る」の語幹であることがあげられる．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 逞，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，*Proc. 1st Annual Meeting of the Association for Natural Language Processing*, 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，第50回情報処理学会全国大会論文集，Vol.3 (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) 西川ヨシカズ，三宮信夫，茨木俊秀：最適化，岩波書店 (1982).

付 録

A.1 最適化のアルゴリズム

本文で述べた制約条件付きの最小化問題は以下のように定義された．

$$\min D(\boldsymbol{p}) \quad \text{subject to} \quad \boldsymbol{p} \in V$$

ここで $\boldsymbol{p}$, $D(\boldsymbol{p})$, V はそれぞれ，決定変数，目的関数，可能領域であり，以下のように定義される．

- 決定変数

$$\boldsymbol{p} = (p_1, p_2, \ldots, p_n)$$

- 目的関数

$$D(\boldsymbol{p}) = \left| \sum_k p_k \boldsymbol{D}_k - \boldsymbol{D} \right|^2$$

- 可能領域

$$V = \left\{ \boldsymbol{p} \mid 0 \le p_k \le 1, \sum_k p_k = 1 \right\}$$
$$= \left\{ \boldsymbol{p} \mid 0 \le p_k, \sum_k p_k = 1 \right\}$$

以下では，これを解くアルゴリズムを説明する．

A.1.1 定義と式の簡略化

まず，以降の式変形で用いる記号を定義し，いくつかの式をあらかじめ簡略化しておく．

$$A = \begin{pmatrix} \boldsymbol{D}_1 \cdot \boldsymbol{D}_1 & \boldsymbol{D}_1 \cdot \boldsymbol{D}_2 & \cdots & \boldsymbol{D}_1 \cdot \boldsymbol{D}_n \\ \boldsymbol{D}_2 \cdot \boldsymbol{D}_1 & \boldsymbol{D}_2 \cdot \boldsymbol{D}_2 & \cdots & \boldsymbol{D}_2 \cdot \boldsymbol{D}_n \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{D}_n \cdot \boldsymbol{D}_1 & \boldsymbol{D}_n \cdot \boldsymbol{D}_2 & \cdots & \boldsymbol{D}_n \cdot \boldsymbol{D}_n \end{pmatrix},$$

$$\boldsymbol{b} = \begin{pmatrix} \boldsymbol{D}_1 \cdot \boldsymbol{D} \\ \boldsymbol{D}_2 \cdot \boldsymbol{D} \\ \vdots \\ \boldsymbol{D}_n \cdot \boldsymbol{D} \end{pmatrix}, \quad \boldsymbol{p} = \begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{pmatrix},$$

$$\boldsymbol{E} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}, \quad \boldsymbol{e}_1 = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

$$\boldsymbol{e}_2 = \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix}, \cdots, \boldsymbol{e}_n = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}$$

A.1.1.1 目的関数

$$D(\boldsymbol{p}) = |\sum_k p_k \boldsymbol{D}_k - \boldsymbol{D}|^2$$
$$= \sum_k p_k \boldsymbol{D}_k \cdot \sum_k p_k \boldsymbol{D}_k$$
$$-2 \sum_k p_k \boldsymbol{D}_k \cdot \boldsymbol{D} + \boldsymbol{D}^2$$
$$= {}^t\boldsymbol{p} A \boldsymbol{p} - 2\boldsymbol{p} \cdot \boldsymbol{b} + \boldsymbol{D}^2 \qquad (5)$$

ただし，行列やベクトルの左肩の t は転置を表すものとする．

A.1.1.2 目的関数の勾配ベクトル

目的関数の勾配ベクトルは，決定変数で偏微分することで得られる．

$$\frac{\partial D}{\partial p_i} = 2\boldsymbol{D}_i \cdot \left(\sum_k p_k \boldsymbol{D}_k - \boldsymbol{D} \right)$$
$$= 2 \left(\sum_k p_k \boldsymbol{D}_i \cdot \boldsymbol{D}_k - \boldsymbol{D}_i \cdot \boldsymbol{D} \right)$$

であるから

$$\nabla D = \left(\frac{\partial D}{\partial p_1}, \frac{\partial D}{\partial p_2}, \cdots, \frac{\partial D}{\partial p_n} \right) = 2(A\boldsymbol{p} - \boldsymbol{b})$$

である．

A.1.1.3 与えられた直線上で目的関数の最小値を与える点

決定変数 $\boldsymbol{p}$ がある点 $\boldsymbol{p}_i$ と傾き $\Delta\boldsymbol{p}$ がなす直線上にある場合は，媒介変数 t を用いて以下のように表せる．

$$\boldsymbol{p} = \boldsymbol{p}_i - t\Delta\boldsymbol{p}, \quad \frac{d\boldsymbol{p}}{dt} = -\Delta\boldsymbol{p} \qquad (6)$$

この直線上で目的関数 $D(\boldsymbol{p}(t))$ の最小値を与える点 $\hat{\boldsymbol{p}}$ は，式 (5)，(6) から $D(t)$ が下に凸な 2 次関数であることが分かるので，以下のように $dD/dt = 0$ を解くことで求められる．

$$\frac{dD}{dt} = \sum_k \frac{\partial D}{\partial p_k} \frac{dp_k}{dt}$$
$$= \frac{d^t\boldsymbol{p}}{dt} \nabla D = -2\Delta^t\boldsymbol{p}\{A(\boldsymbol{p}_i - t\Delta\boldsymbol{p}) - \boldsymbol{b}\}$$

よって，

$$\frac{dD}{dt} = 0 \Leftrightarrow \Delta^t\boldsymbol{p}\{A(\boldsymbol{p}_i - t\Delta\boldsymbol{p}) - \boldsymbol{b}\} = 0$$
$$\Leftrightarrow t = \frac{\Delta^t\boldsymbol{p} A(\boldsymbol{p}_i - \boldsymbol{b})}{\Delta^t\boldsymbol{p} A \Delta\boldsymbol{p}}$$

ゆえに，

$$\hat{\boldsymbol{p}} = \boldsymbol{p}_i - \frac{\Delta^t\boldsymbol{p} A(\boldsymbol{p}_i - \boldsymbol{b})}{\Delta^t\boldsymbol{p} A \Delta\boldsymbol{p}} \Delta\boldsymbol{p} \qquad (7)$$

である．

A.2 アルゴリズム

式 (5) から明らかなように，この最小化問題の目的関数が強意の凸関数であるから，以下のような処理をたかだか決定変数の数だけ繰り返すことで，大域的最小点を求めることができる（**図 5** 参照）[9]．

(1) $\boldsymbol{p}$ の初期値を可能領域の中心とする．

$$\boldsymbol{p}_i = \frac{1}{n}\boldsymbol{E}$$

(2) 制約条件が活性であるか否かを表すベクトル $\boldsymbol{a}$ を初期化する．

$$\boldsymbol{a} = (0, 0, \cdots, 0)$$

(3) 目的関数の勾配ベクトルを求め，不活性条件が

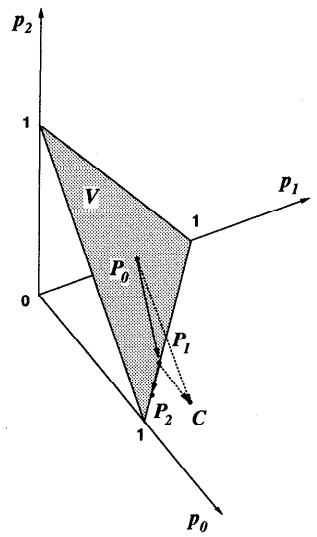


図5 最適化の概念図
Fig. 5 Concept of the optimization.

張る空間に対する射影を求める．

$$\Delta \boldsymbol{p} = \nabla D - \sum_k a_k \frac{\nabla D \cdot \boldsymbol{e}_k}{\boldsymbol{e}_k \cdot \boldsymbol{e}_k} \boldsymbol{e}_k - \frac{\nabla D \cdot \boldsymbol{E}'}{\boldsymbol{E}' \cdot \boldsymbol{E}'} \boldsymbol{E}'$$

ここで $\boldsymbol{E}'$ は，$\boldsymbol{E}$ の不活性条件が張る空間に対する射影である．

$$\boldsymbol{E}' = \boldsymbol{E} - \sum_k a_k \frac{\boldsymbol{E} \cdot \boldsymbol{e}_k}{\boldsymbol{e}_k \cdot \boldsymbol{e}_k} \boldsymbol{e}_k$$

(4) 点 $\boldsymbol{p}_i$ と傾き $\Delta \boldsymbol{p}$ がなす直線上で目的関数 D の最小値を与える点 $\hat{\boldsymbol{p}}$ を式 (7) を用いて計算する．

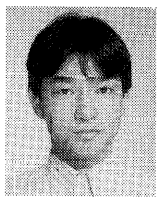
(5) $\hat{\boldsymbol{p}}$ が可能領域に含まれるか否かを判断する．可能領域に含まれるならば，これを解として終了する．含まれなければ，$\boldsymbol{p}$ が可能領域に含まれるという条件下で t の最大値を以下の式を用いて計算する．ただし，$(\boldsymbol{x})_k$ はベクトル $\boldsymbol{x}$ の第 k 成分を表す．

$$t = \max \frac{(\boldsymbol{p}_i)_k}{(\Delta \boldsymbol{p})_k}$$

また，最大値を与える k に対応する制約条件を活性とする $(a_k = 1)$．

(6) $\boldsymbol{p}_{i+1} = \boldsymbol{p}_i - t\Delta \boldsymbol{p}$ とし，処理 (3) へ戻る．

(平成 9 年 7 月 28 日受付)
(平成 10 年 4 月 3 日採録)



森 信介（正会員）

1995 年京都大学大学院工学研究科修士課程修了．1998 年同大学大学院博士後期課程修了．同年日本アイ・ビー・エム入社，東京基礎研究所副主任研究員．工学博士．計算言語学の研究に従事．言語処理学会会員．

長尾 眞（正会員）

1959 年京都大学工学部電子工学科卒業．工学博士．同大学工学部助手，助教授を経て，1973 年より同大学工学部教授．国立民族学博物館教授を兼任 (1976.2～1994.3)．同大学大型計算機センター長 (1986.4～1990.3)．日本認知科学会会長 (1989.1～1990.12)．パターン認識国際学会副会長 (1982～1984)．日本機械翻訳協会初代会長 (1991.3～1996.6)．機械翻訳国際連盟初代会長 (1991.7～1993.7)．電子情報通信学会副会長 (1993.5～1995.4)．情報処理学会副会長 (1994.5～1996.4)．京都大学附属図書館長 (1995～1996)．京都大学工学研究科長 (1995～1997)．京都大学総長 (1997～)．