疑似Nグラムを用いた助詞的定型表現の自動抽出

新納浩幸† 井佐原均††

本論文では簡易な字面処理によって，助詞に相当する定型表現（助詞的定型表現）をコーパスから自動抽出する手法について述べる．ここで抽出する表現は，例えば「に関して」や「に基づく」のように，助詞的な働きをする定型的な表現である．これらの定型表現は処理上，一単語として扱うのが妥当であり，予め収集しておく必要がある．定型表現を自動抽出する従来の手法の多くは対象言語が英語である．しかし日本語の場合，英語と異なり，単語間の共起の強さを計るには，基本的に文を単語に分割するための形態素解析が必要である．しかも形態素解析には，曖昧性，未知語などの問題がついてまわり，単語間の共起の強さを計るのは英語ほど容易ではない．完全な字面処理からのアプローチとしては，「ある文字列が1つのユニットになっていればその文字列の前後には様々な種類の文字が現れる」というアイデアをもとに，大規模コーパスから得られたNグラムによって定型表現を取り出す手法がある．本手法は基本的にこの考え方を利用する．ただし，助詞的定型表現の持ついくつかのヒューリスティックスと句読点の情報を活用し，完全なNグラムを作ることを避け，そのサブセットである疑似Nグラムと呼ぶある種の文字列の頻度情報だけを利用する．結果として，簡易な字面処理だけによって，定型表現の抽出が可能となっている．このため，本手法は，実験の拡大，再現が容易であるという利点も持つ．

Automatical Extraction of Frozen Patterns to Act as a Postpositional Particle by Pseudo N-gram

Hiroyuki Shinnou† and Hitoshi Isahara††

This paper presents a new method to extract frozen patterns to act as a postpositional particle automatically from a Japanese corpus. It should be more effective to handle a frozen pattern as one word. Therefore, it is necessary to include frozen patterns in dictionary as lexical entries. Conventional research for automatical extraction of frozen patterns set English as a target language. But, in Japanese, morphological analysis is necessary to recognize words, so there exist different difficulties from English. The proposed method is based on the N-gram method which need not morphological analysis. In addition, by using heuristics rules for frozen patterns to act as a postpositional particle, it can be avoided to build up complete N-gram table. Thus, this method is very simple. it can be easily applied a large scale experiment without expensive resources.

1. はじめに

本論文では簡易な字面処理によって，助詞に相当する定型表現（以下，助詞的定型表現と呼ぶ）をコーパスから自動的に抽出する手法について述べる．

ここで抽出しようとしている表現は例えば「に関して」や「に基づく」のように助詞的な働きをする定型的な表現である．これらの多くは一種の慣用表現であり，個々の構成語からはその意味を作り出せないために，その表現自体を一語として取り扱うのが自然である．また慣用表現とは判定しがたいが，構成語間に強い共起関係を持った定型表現であっても，処理効率の面からは一語として取り扱うことが望ましい．このような定型表現をどのように収集するかは，自然言語処理における重要な問題である．

特に本論文では助詞的定型表現に着目する．助詞的定型表現は，一般に構成語の順序が不変であり，構成語の間に別の語の挿入が起こらないことから，一語として扱う効果が高いからである．また日英翻訳を行う場合，この種の表現の多くは，英語において前置詞一語に対応しており，これらの表現を収集しておくことで，無用な変換処理を避けることができる．また近年，用例検索を用いた英作文支援システムの提案がいくつかなされているが[1),2)]，隅田のシステム[3)]のように，入力文と登録文との類似性を文の骨格に求める場合には，文中の機能語（つまり助詞相当語句）を適切に求めることが重要であり，この点からも助詞的な定

† 茨城大学工学部システム工学科
Department of Systems Engineering, Faculty of Engineering, Ibaraki University
†† 電子技術総合研究所知能情報部自然言語研究室
Natural Language Section, Machine Understanding Division, Electrotechnical Laboratory

型表現を収集し，充実させておくことが望まれる．

定型表現や慣用表現を収集するには，通常，人手によるしかなく，膨大な時間と手間が必要である[4]．また慣用表現の定義は曖昧であるため[5]，集めた表現が一貫性をもっているかどうかも疑わしい．例えば「に関して」を慣用表現と捉えることは，多くの人が納得するであろうが，「に限って」を慣用表現として扱えるかどうかは判断の分かれるところである．

これらの点から定型表現や慣用表現の自動抽出の試みがなされているが[6)~8)]，その多くは対象言語が英語である．英語の場合，単語区切りが明確であり，単語間の共起の強さを計る手段が比較的容易である．一方，日本語の場合，単語間の共起の強さを計るには，基本的に，文を単語に分割するための形態素解析が必要である．しかも形態素解析には，曖昧性，未知語などの問題がついてまわり，単語間の共起の強さを計るのは英語とは違った困難性を持っている．完全な字面処理からのアプローチとしては，長尾の研究がある[9]．これは，大規模コーパスからNグラムを作成する手法であるが，そこでは「ある文字列が1つのユニット（単語に相当する語句）になっていればその文字列の前後には様々な種類の文字が現れる」というアイデアをもとに，作成したNグラムを用いて定型表現を取り出せることが示唆されている．

本論文では上記の長尾の示したアイデアを基に，助詞的定型表現の持ついくつかの性質やいくつかのヒューリスティックスを導入することで，形態素解析を行わずに，単純な字面処理だけで助詞的定型表現をコーパスから抽出する手法を提案する．

本手法の特徴は，「助詞的定型表現は，ほとんどの場合，平仮名文字だけで構成され，漢字が含まれていてもその漢字列の長さは非常に短い」，「日本語の自立語の先頭文字は漢字で表記される」というヒューリスティックスを用いたことである．これらによってある種の文字列の出現頻度（これをここでは疑似Nグラムと呼んでいる）だけを調べることで，助詞的定型表現かどうかの判定が可能になっている．更に，抽出対象を助詞的定型表現に限定したことと，「日本語の句読点は単語の切れ目を表す」という性質を用いたことによって，疑似Nグラム中の文字列から助詞的定型表現の候補をまず最初に抽出している．そして，それら候補に対してだけ，疑似Nグラムを用いた判定法を行えばよいために効率的な手法となっている．本手法は形態素解析を必要としないために簡易であり，しかも完全なNグラムを作る手法と比較して計算量も小さく，実験の拡大，再現が容易であるという利点を持つ．また抽出する表現を助詞的定型表現に限定しているために，句読点の情報を利用したり，指示代名詞や取り立て詞といった特別の品詞に対して補正を入れるなどの文法的なヒューリスティックスも有効に用いることができる．

2. 助詞的定型表現の自動抽出

本論文の手法は，

【第0段階】 まず最初に取り出したい助詞的定型表現の最小の長さ k と，取り出したい助詞的定型表現に含まれる漢字列の長さの最大長 n を設定しておく．

【第1段階】 次にある種の文字列（k と n に依存する）だけを対象に，その文字列のコーパス中の出現頻度を調べる．その際に，その文字列の前後に句読点があるものの個数も記録しておく．ここで集めた文字列の頻度および前後の句読点情報を疑似 N_k^n グラムと呼ぶことにする．

【第2段階】 次に疑似 N_k^n グラムからいくつかのヒューリスティックスを用いて，助詞的定型表現となりそうな文字列を選択する．

【第3段階】 最後に疑似 N_k^n グラムを用いて，第2段階で選択された文字列各々が助詞的定型型表現となるか否かを決定する．

2.1 準　　備

本論文の手法を説明するための準備として，α_n 文字列という用語を定義する．

[定義] α_n 文 字 列

α_n 文字列とは平仮名と漢字だけから構成され，しかも，漢字列の長さは n 以下であるような文字列である．

上記の定義から，“きょうはあめです”，“きょうは雨です”，は α_1 文字列だが，“きょうは、あめです”（読点が含まれている），“きょうはアメです”（カタカナが含まれている），“今日は雨です”（漢字列“今日”の長さが2）は α_1 文字列ではない．

2.2 変数の設定

本手法の第0段階として，抽出したい助詞的定型表現の最小の長さ k と，抽出したい助詞的定型表現に含まれる漢字列の最大長 n を設定する．

k を小さく，n を大きく設定するほど抽出できる定型表現の種類は多くなるが，その反面計算量も大きく

なる．本論文では実験的に $k=4$, $n=1$ と設定した．

以下の記述で単に疑似Nグラムとあれば，疑似 N_4^1 グラムのことであり，α 文字列とあれば，α_1 文字列のことである．

2.3 疑似Nグラムの作成

本手法の第1段階として，コーパスからα文字列をすべて抽出する．抽出した結果を

α 文字列　a0 a1 a2 b0 b1 b2　　(1)

の集合の形でまとめる．ここで a1 は α 文字列の前の文字が読点であった回数，a2 は α 文字列の前の文字が句点であった回数，a0 は α 文字列の前の文字が読点でも句点でもなかった回数，b1 は α 文字列の後ろの文字が読点であった回数，b2 は α 文字列の後ろの文字が句点であった回数，b0 は α 文字列の後ろの文字が読点でも句点でもなかった回数を表している．

例を示す．以下の例文から長さ4以上のすべてのα文字列を抽出することで，疑似Nグラムの一部を作成してみる．注意として，通常文頭の文字の前には句点が存在している．この例の場合もそう考えることにする．結果を**図1**に示す．

【例文】「そこで先の条件に基づいて、かなりの語を集めたとしても、また新たな未知語が発見されてしまうため，辞書に関しても事情は単純ではない。」

図1はこの例文だけからα文字列を取り出したものだが，実際はコーパス中のすべての文からα文字列を取り出すことで疑似Nグラムを作成する．

そこで先	0	0	1	1	0	0
こで先の	1	0	0	1	0	0
に基づい	1	0	0	1	0	0
基づいて	1	0	0	0	1	0
かなりの	0	1	0	1	0	0
を集めた	1	0	0	1	0	0
集めたと	1	0	0	1	0	0
めたとし	1	0	0	1	0	0
たとして	1	0	0	1	0	0
.....						
そこで先の	0	0	1	1	0	0
に基づいて	1	0	0	0	1	0
を集めたと	1	0	0	1	0	0
集めたとし	1	0	0	1	0	0
めたとして	1	0	0	1	0	0
.....						
を集めたとして	1	0	0	1	0	0
集めたとしても	1	0	0	0	1	0
されてしまうた	1	0	0	1	0	0
れてしまうため	1	0	0	0	1	0
を集めたとしても	1	0	0	0	1	0
されてしまうため	1	0	0	0	1	0

図1 例文に対する疑似Nグラム
Fig. 1 Pseudo N-gram for the example sentence.

2.4 助詞的定型表現の候補の抽出

本論文の第2段階として，上記で得られた疑似Nグラムの文字列から助詞的定型表現と考えられるものを選択する．まず最初に，本論文は以下のヒューリスティックスを利用していることに注意する．

(**H.1**)「助詞的定型表現は，ほとんどの場合，平仮名文字だけで構成され，漢字が含まれていてもその漢字列の長さは非常に短い」

(H.1)はヒューリスティックスであり論理的な根拠はないが，通常思いつくような助詞的定型表現は(H.1)を満たしている．しかも漢字列の長さは1以下のものが目立つ．そこで本論文では漢字列の長さを1以下として実験してみることにする．すると(H.1)は以下のように読み換えることができる．

(**H.1′**)「助詞的定型表現は α_1 文字列である」

(H.1′)によって目的とする助詞的定型表現は，疑似Nグラム中の文字列に存在している．

次に疑似Nグラム中から助詞的定型表現と考えられるものを選択するために，以下のヒューリスティックスを利用する．

(**H.2**)「助詞的定型表現の先頭の単語は助詞である」

(**H.3**)「助詞的定型表現の前の文字に読点（、）や句点（。）は現れない」

(**H.4**)「助詞的定型表現の次の文字に句点は現れない」

これらのヒューリスティックスは文法的にも妥当であり，ヒューリスティックスというよりも助詞的定型表現の性質といってもよいものである．

(H.2)の判定は以下の文字列が先頭文字列になっていることを確認することによって行う．

「が」「の」「に」「を」「へ」「と」「から」
「より」「で」「まで」

この処理は文字列の一致だけで見ており，形態素解析は行わない．また，上記にあげた文字列は格助詞であり，本論文では抽出する定型表現を格助詞的定型表現に暗に想定している．他の助詞を含めると別の種類の助詞的定型表現も抽出できると考えられるが，ここでは助詞的定型表現の中心となる格助詞的なものだけをまず対象にする．

(H.3)は式(1)における a1 と a2 が0であることを確認する．(H.4)は式(1)における b2 が0であることを確認する．(H.3)によって自立語は省くことが

でき，(H.4)によって助動詞的な定型表現も省くことができる．

図1で示した文字列からは，

「に基づい」「を集めた」「としても」「に関して」
「ではない」「に基づいて」「を集めたと」
「に関しても」「を集めたとし」「を集めたとして」
「を集めたとしても」

の11個が候補として取り出される．一般にこの候補の中には，文字列が単語の列から成り立っていないものや，先頭の単語が格助詞ではないものも含まれてしまう．以下の処理によって，この候補の中から正しい定型表現を抽出する．

2.5 助詞的定型表現の判定

本手法の最後の処理として，上記で選択した助詞的定型表現の候補の各々が実際に定型表現になっているかどうかを疑似Nグラムを用いることで判定する．

判定は3つの条件から行う．第1の条件はその文字列の先頭文字（列）はある単語となっている．第2の条件はその文字列の末尾の文字（列）はある単語となっている．第3の条件はその文字列中の単語どうしの結合度が強い．これらの条件を満たした文字列を助詞的定型表現と判定する．

2.5.1 直前文字からの判定（判定条件1）

ここでの判定条件は，助詞的定型表現の候補の文字列の先頭文字（列）が単語であることを調べる．ある文章から切り出した文字列が意味をなすには，その文字列が単語の列になっている必要があるために，当然満たすべき性質である．

今，助詞的定型表現の候補の文字列（β）の長さがn，出現回数がmとする．疑似Nグラムから以下の形をした長さ $n+1$の文字列とその頻度・句読点情報を取り出す．

$c_1\beta\ a0_1\ a1_1\ a2_1\ b0_1\ b1_1\ b2_1$
$c_2\beta\ a0_2\ a1_2\ a2_2\ b0_2\ b1_2\ b2_2$
…
$c_k\beta\ a0_k\ a1_k\ a2_k\ b0_k\ b1_k\ b2_k$

今βの先頭の文字列が単語，つまりこの場合，格助詞である場合，その直前にはほとんどの場合，名詞が現れる．さらに，日本語の場合，多くの名詞は漢字で表記される．つまり基本的に$c\beta$（cは平仮名）の形をしたα文字列の出現回数は，βの出現回数に比べて小さいことが予想できる．本論文では基本的にこの性質を利用しており，以下の式の値がmに比べて非常に小さければβの先頭文字（列）は単語であると考える．

$$\sum_{i\in H}(a0_i+a1_i+a2_i) \qquad (2)$$

where H is a set of i such as c_i is 平仮名

ただし，平仮名で利用される名詞として，指示代名詞（「これ」「この」etc）と形式名詞（「の」「こと」etc）は多用されるために，そのための補正を入れる．

$$\sum_{i\in G}0.1\cdot(a0_i+a1_i+a2_i)$$

where G is a set of i such as c_i is「れ」,「の」or「と」

また平仮名1文字による普通名詞の利用は非常に少ないため，$c\beta$の前に句読点が現れる場合はほとんどないと考えられる．この点を考慮して式(2)を以下のように補正する．

$$\sum_{i\in H-G}(a0_i+10.0\cdot(a1_i+a2_i))$$

判定は，式(3)の値が，0.1以下を条件とした．

$$\frac{1}{m}\cdot\left\{\sum_{i\in G}0.1\cdot(a0_i+a1_i+a2_i)+\sum_{i\in H-G}(a0_i+10.0\cdot(a1_i+a2_i))\right\} \qquad (3)$$

2.5.2 直後文字からの判定（判定条件2）

ここでの判定条件は，助詞的定型表現の候補の文字列の末尾の文字(列)が単語であることを調べる．前述したように，ある文章から切り出した文字列が意味をなすには，その文字列が単語の列になっている必要があるために，この性質も当然満たすべきものである．

今，助詞的定型表現の候補の文字列（β）の長さがn，出現回数がmとする．疑似Nグラムから以下の形をした長さ $n+1$の文字列とその頻度・句読点情報を取り出す．

$\beta d_1\ a0_1\ a1_1\ a2_1\ b0_1\ b1_1\ b2_1$
$\beta d_2\ a0_2\ a1_2\ a2_2\ b0_2\ b1_2\ b2_2$
…
$\beta d_k\ a0_k\ a1_k\ a2_k\ b0_k\ b1_k\ b2_k$

今βが助詞相当語句である場合，その直後にはほとんどの場合，名詞，動詞，形容詞，形容動詞のいずれかが現れる．つまりほとんどの場合，自立語が現れる．さらに，日本語の場合，ほとんどの自立語の先頭文字は漢字で表記される．つまり基本的にβd（dは平仮名）の形をしたα文字列の出現回数は，βの出現回数に比べて小さいことが予想できる．

本論文では基本的にこの性質を利用して，以下の式の値がmに比べて非常に小さければβの末尾文字(列)は単語であると考える．

$$\sum_{i\in H}(b0_i+b1_i+b2_i) \qquad (4)$$

where H is a set of i such as d_i is 平仮名

ただし，取り立て詞（「は」「も」etc）は助詞相当語句の直後にも現れることが多い．また形式名詞「の」も助詞相当語句の直後に現れやすい．このための補正を入れる．

$$\sum_{i\in F} 0.1\cdot(b0_i+b1_i+b2_i)$$

where F is a set of i such as d_i is「は」,「も」or「の」

また，「は」「も」「の」以外の平仮名が直後に現れたβdの文字列の更に直後に句読点が現れる場合はほとんどないと考えられる．この点を考慮して式(4)を以下のように補正する．

$$\sum_{i\in H-F}(b0_i+10.0\cdot(b1_i+b2_i))$$

判定は，式(5)の値が，0.1以下を条件とした．

$$\frac{1}{m}\cdot\left\{\sum_{i\in F}0.1\cdot(b0_i+b1_i+b2_i)+\sum_{i\in H-F}(b0_i+10.0\cdot(b1_i+b2_i))\right\} \qquad (5)$$

2.5.3 単語間の結合力からの判定（判定条件3）

上記までの抽出処理，判定処理によって，抽出できた表現の中には，統語的には助詞的定型表現の形をなしているが，実際は定型表現と考えられない表現も存在する．例えば，「を招いて」という文字列は α 文字列であり，先頭文字が格助詞「を」であり，直前直後の文字は漢字表記される名詞や動詞が多用されるために，上記までの処理によって抽出されている可能性が高い．しかしこの表現は定型表現とは考えられない．

ここでは定型表現の構成単語は結合力が強いという性質を用いてこれらの表現を取り除く．例を用いて説明する．上記例の「を招いて」という表現は，例えば「太郎が花子を招いて、…」という文に出現するが，この文は「花子を太郎が招いて、…」という文にも置き換えることができる．つまり「招いて」の直前の文字は「を」である場合のほかに「が」であるものの割合も少なからずある．一方，「に対して」という表現は例えば「私は政治に対して懐疑的だ」という文に出現するが，この文は先のような入れ換えが起こらないために，「対して」の直前の文字はほとんどの場合「に」である．この性質を用いて先頭の1文字と残りの文字列との結合力をみることで，第3の判定を行う．

今，助詞的定型表現の候補の文字列（β）が以下の形をしており，長さがn，出現回数がmとする．

$$\beta=e_1\cdot e_2\cdots e_n$$

疑似Nグラムから以下の形をした長さnの文字列とその頻度・句読点情報を取り出す．

$f_1\gamma$ $a0_1$ $a1_1$ $a2_1$ $b0_1$ $b1_1$ $b2_1$

$f_2\gamma$ $a0_2$ $a1_2$ $a2_2$ $b0_2$ $b1_2$ $b2_2$

…

$f_k\gamma$ $a0_k$ $a1_k$ $a2_k$ $b0_k$ $b1_k$ $b2_k$

where $\gamma=e_2\cdot e_3\cdots e_n$

今 β が定型表現である場合，文字 e_1 と e_2 は非常に強い結合力がある．このため，f_i のほとんどは e_1 と同じ文字になると考えられる．本論文では基本的にこの性質を利用して，以下の式(6)の値が0.2以下の場合に，β の先頭文字（列）は単語であると判定する．

$$\frac{1}{m}\cdot\sum_{i\in E}(a0_i+a1_i+a2_i) \qquad (6)$$

where E is a set of i such as $f_i\neq e_1$

3. 抽出実験と評価

3.1 抽出実験

本手法の有効性を確かめるために小規模のコーパスを利用した抽出実験を行った．コーパスとしては朝日新聞の記事1か月分（テキストデータ部分約9メガバイト）を利用した．

第1段階の処理として作成された疑似Nグラムの文字列の種類は1,302,492種類であった．

次にこの疑似Nグラムから助詞的定型表現の候補を取り出した．ただしここでは出現頻度が10以下のものは対象から外すことにした．結果として3,031種類の文字列が選択された．次に2.3節で述べた判定条件を各々独立に試した．判定条件1では1,318種類，判定条件2では443種類，判定条件3では903種類の文字列が選択された．最後にこれらの条件をすべて満たした文字列をとることで，最終的に91種類の文字列を助詞的定型表現として抽出した．この抽出した表現を**図2**に示す．図2の各々の表現は○，△，×の3種類によって，区分けされているが，○は正しい抽出と評価したもの，×は誤った抽出と評価したもの，△は評価が困難なものである．更に○の中は4つに分類されている．これら評価の詳細については次章で述べる．

3.2 評　価

本実験で抽出した表現が定型表現であるかどうかを客観的に評価することは困難であるが*，抽出した表

* これは定型表現に対する客観的な定義が困難であることの裏返しである．

○	において，における，について，にとって，に関して，に関する， に基づき，に対して，に対する
	によって，に沿った，に沿って，に向けた，に向けて，に際して， にわたり，に比べて，に面した，のもとで，をめぐる，を巡って， をめどに，にわたって，を除いた，を除くと，を通じて，を踏まえた， を踏まえて，になぞらえた
	からみると，にちなんで，に照らして，をはさんで，からすれば， をはじめとする，のあり方について，が続く中で，を伴わない
	においては，においても，にかけての，についての，については， についても，によっては，に向けての，に対しては，に対しても， のなかには，にとっても，までにも，としての，の上では
△	の狙いは，の側から，とは別に，と題した，と題して，と題する， に限らず，との間で，に伴って，の間から，の招きで，になるとの， をだまし，を含めた，を使えば，を終えて，の調べでは，の話として， を見ながら，を取り巻く，を受けたが，から開かれる，に運ばれたが， をとりながら，を見ていると，を行うことを，を守るための， で開かれていた，の調べによると，を受けて行われる， での主なやりとりは，をにらんだ
×	から引き，から売り，よりやや，にとって大きな，でありながら， がんのため

図 2 抽出された定型表現
Fig. 2 Frozen patterns to be extracted.

現を以下のように分類することで疑似的に評価することにした．それぞれの括弧内の○は正しい抽出，×は誤った抽出，△は不明という判定を示している．

（1） 文法的に助詞的定型表現でないもの（×6種類）

例えば「がんのため」という表現は文法的に助詞相当の働きをしていない．また「でありながら」の構成単語の「で」は助詞の「で」でなく，助動詞「だ」の連用形「で」である．また「から引き」は，「から引き出す」などの複合語の一部が取り出されたものである．これらは文法的に助詞的定型表現でない．

（2） 辞書に一語として登録されているもの（○9種類）

例えば「において」という表現は，辞書に一語として登録されている．このような表現は定型表現と考えて不都合はない．

（3） 日英翻訳の際に前置詞に訳されるもの（○20種類）

例えば「に沿って」という表現が使われている例文を既存の辞書から探すと，「沿う」という項目に以下の例文を見つける．

・「川に沿って歩いて行った」
We walked along the river.

上記の例文の場合，「に沿って」が前置詞 along に対応している．このように英訳する際にその表現が前置詞として訳される場合は，その表現を1つの語として考えるのは自然であるため，これらの表現は定型表現と判定する．

また「に沿って」が定型表現と判定された場合，その変化形である「に沿った」も定型表現と判定する．

（4） 英訳する場合に他の表現に言い換えたり，解析・変換の際に大きな構造変換を伴うようなもの（○9種類）

例えば「をはじめとする」や「をはさんで」を使った以下の例文の翻訳を見てみる．

・校長をはじめとする5人の先生がその会に出席した．
Five teachers, including the principal, attended the meeting.

・警察隊と学生は道をはさんで睨みあった．
The police and the students glared at each other from opposite sides of the street.

それぞれ「をふくめて」や「の反対側から」などのようにいい替えを行っている．このような表現を英訳する場合には，大きな変換処理が必要になり，予め1語として捉えることは有益であると思われるため，定型表現と判定する．

（5） （2），（3），（4）などの判定方法で定型表現と判定された表現に取り立て詞が継ったもの（○15種類）

多くからの格助詞には取り立て詞が後続できる（例「から」＋「も」→「からも」，「に」＋「は」→「には」など）．このような組合せは数も多くないために，機械処理では分割して処理せずに1つの語として取り扱う場合が多い．この点から，（2），（3），（4）の判定法で助詞的定型表現と判定される表現に取り立て詞が継ったものは助詞的定型表現と判定する．

（6） 上記の項目以外（△32種類）

この分類に入るものは，例えば「と題して」のように，分解して処理しても特に問題がないものである．これらを定型表現のように1語として処理した方がよいかどうかは，単純には判定できない．

△の判定については，考察の項で述べるが，特に定型表現のように1語として取り扱うことで不都合はないと考えている．このため実験では，正しい抽出は93.4%であり，本手法の簡易性を考慮すると，有効な抽出ができていると考える．

表 1 補正の評価
Table 1 Evaluation of corrections.

	絞り込みの変化	抽出数の変化	○の変化	△の変化	×の変化
補正 1	−311	−21	−14	− 6	− 1
補正 2	+23	+ 5	+ 3	+ 1	+ 1
補正 3	−79	−23	−17	− 6	0
補正 4	+ 7	0	0	0	0

次に判定条件 1，2 で用いた補正について述べる．本論文では以下の 4 つの補正を行っている．

補正 1 通常の名詞は漢字列で表記されるが，指示代名詞と形式名詞は平仮名で表記されるため行った式(2)の補正（判定条件 1）．

補正 2 平仮名 1 文字による名詞は通常使われないために行った式(2)の補正（判定条件 1）．

補正 3 助詞の直後は通常自立語であるが，取り立て詞などは助詞の直後にも現れるために行った式(4)の補正（判定条件 2）．

補正 4 平仮名 1 文字による名詞は通常使われないために行った式(4)の補正（判定条件 2）．

各々の補正を行わなかった場合に，各々の判定条件による絞り込みの数がどのように変化するかを**表 1** に示す．また表 1 には最終的に抽出される助詞的定型表現の判定（○△×）の数がどのように変化するかも示す．

補正 1 と補正 3 は判定条件を緩める方向に働くので，その補正を外した場合には，絞り込みが厳しくなり，最終的に得られる数は少なくなる．上記実験でもそのような結果が出ているが，補正を外したことで省かれる表現のほとんどは正解（○△）なので，この補正は有効に機能している．また逆に補正 2 と補正 4 は判定条件を厳しくする方向に働くので，その補正を外した場合には，絞り込みが緩くなり，最終的に得られる数は多くなる．上記実験でもそのような結果が出ている．ただし，新たに取り出せた表現の多くも正解であった．これは補正 2 と補正 4，特に補正 2 は逆効果になっていることを示している．調査した結果，指示代名詞「その」「あの」「この」の 2 文字目の「の」が本手法では格助詞だと認識され，その 2 文字前の文字（例えば「その」なら「そ」の前の文字）が句読点である場合もあったからである．

4. 考　　察

まず実験で△と判定されたものについて述べる．通常，ある表現を 1 語として取り扱うことで不都合があるのは，その表現内に別の語が挿入されたり，構成単語の語順が変化したりする場合である．また記憶する量や検索の問題もある．しかし，ここで抽出した表現は別の語の挿入や，語順の変化がなく，しかも一般の単語よりも出現頻度が高い．これらのことから，△と判定された表現であっても定型表現として処理することで不都合はないと考える．

判定条件の補正については，補正 4 はあまり意味をなさないことが確認された．また補正 2 は逆効果になっている．補正 2 は助詞「の」の扱いを考慮するように補正を設定し直しても良いが，これらのヒューリスティックスは省いても良いと思われる．句読点情報は主に本手法の第 2 段階（疑似 N グラム中の文字列から助詞的定型表現の候補を取り出す）で利用されており，第 3 段階の判定条件にまで持ち込む必要はなかったと考える．

本手法は簡易であることが大きな特徴であるが，そのためにある程度の質の劣化は否めない．抽出できなかった定型表現のいくつかを対象に取り出せなかった原因を調べた．まず最大の原因は，その表現がコーパス中にほとんど出現しないことである．実験で用いたコーパスは，その大きさが小さいという点と，新聞記事であり文体が統一されていたためと予想している．次に判定条件 3 で棄却されているものが日だった．実験の際の判定条件 1，2，3 は数字的に見れば，判定条件 2（直後文字からの判定）が厳しい条件になっているが，実際には判定条件 3（単語間の結合力からの判定）の方で正しい定型表現が削られている場合が多い．例えば，「の近くに」という表現は定型表現と考えられるが，“近くに”という文字列の前の文字は**図 3** のようになっている．本手法のように単純に全体の割合だけで見ると，この時点で却下されてしまう．これを避けるためには，判定条件 1，2 のように補正を入

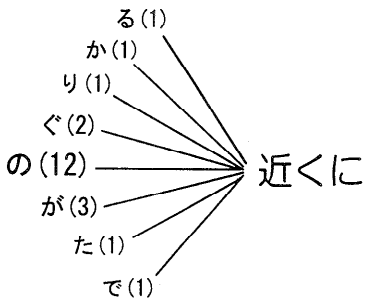


図 3 前置する文字
Fig. 3 Characters which precede a phrase.

れたり，単純な割合ではなく，例えば出現分布を考慮するような工夫を判定条件3に盛り込む必要がある．また語義の違いにより，先頭の格助詞が異なる場合があり，この点でも判定条件3で落とされてしまう．この点は字面だけの表層的な情報からだけでは対応が困難であると思われる．一方で，本手法は比較的文字列の長さが長い定型表現を取り出せるという長所がある．文字列の長さが長い定型表現は，人間が手作業で集めた場合には，見落としやすいものである．この点から本手法を手作業による収集の補助的なツールとして利用するのも良いと考える．

抽出の第0段階で行われる n や k の設定について述べる．本論文では n を1，k を4と設定した．これは当初抽出したいと考えていた表現が，すべて長さが4以上で，漢字列の長さが1以下であったからである．当然，これらの数値は可変であり，更に詳しく取り出したい場合には n を2，k を3程度で行うのが良いであろう．

本手法と長尾の研究[9]で示唆された手法とを比較する．基本的に文字列の前後文字を見て，その文字列が1ユニットになることを調べるというアイデアは同じである．ただし後者の手法では，長さ n の文字列の前後文字を調べるために，長さ $n+1$ のすべての文字列を求めている．一方，本論文の目的とする助詞的定型表現の抽出のために必要となる長さ $n+1$ の文字列は，「自立語の先頭文字は漢字で表記される」，「ほとんどの名詞は非平仮名列で表記される」というヒューリスティックスから，長さ $n+1$ の α 文字列で十分である．さらに助詞的定型表現自身が α 文字列となっている．これらのことから，本手法では完全なNグラムを作成せずに，疑似Nグラムでそれを代用できている．また抽出する表現が助詞的定型表現に限定しているために，いくつかのヒューリスティックを用いて，抽出候補の文字列を絞り込むことができる．このため判定条件を適用する文字列の数が少なく，抽出処理自体が軽い．また本手法は句読点の情報を積極的に用いている点も特徴である．先の研究では単語の区切りを判定するために文字種の数だけに注目しているが，句読点が直前に存在すれば，その文字列の先頭文字(列)は単語になっているはずなので，判定条件の部分で重みをつけた方が確からしい結果が出ると思われる．

本手法は完全に字面処理だけによって定型表現を抽出している．これは実験の拡大，再現が容易という利点がある．近年「コーパスからの知識獲得」の研究が盛んだが[10]，それらの多くは加工されたデータが必要であったり，構文解析が必要であったりする．そのため別の環境での実験の再現，あるいは大規模な実験への拡大が容易ではない．一方，本手法は基本的に字面処理であり，生のテキストを用意するだけで実験が可能である．プログラミングも容易であり，この実験での処理の多くは awk や sort といった unix 上のツールを用いて行えている．

5. おわりに

本論文では簡易な字面処理だけによって，助詞に相当する定型的な表現をコーパスから自動的に抽出する手法について述べ，その有効性を実験によって確かめた．

本手法は助詞的定型表現の持ついくつかのヒューリスティックスを活用しており，完全なNグラムを作ることを避け，そのサブセットである疑似Nグラムと呼ぶある種の文字列の頻度情報だけを利用している．結果として，簡易な字面処理だけによって，定型表現の抽出が可能となっている．このため，本手法は，実験の拡大，再現が容易であるという利点を持つ．

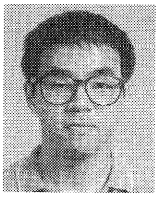
今後は本手法を応用して，助動詞に対応する定型表現や動詞句に対応する定型表現などの自動抽出を試みたい．また本実験を異なる分野のコーパスに適用し，分野別の定型表現も収集したい．

参考文献

1) Sato, S.: CTM: An Example-Based Translation Aid System, *Proc. of COLING-92*, pp. 1259-1263 (1992).
2) 武田明子，古郡廷治：例文をもとにした英文書作成支援システム，情報処理学会論文誌，Vol. 35, No. 1, pp. 53-61 (1994).
3) 隅田栄一郎，堤　豊：翻訳支援のための類似用例の実用的検索法，電子情報通信学会論文誌，Vol. J74-D-II, No. 10, pp. 1437-1447 (1991).
4) 首藤公昭，吉村賢治，武内美津乃，津田健蔵：日本語の慣用表現について，情報処理学会自然言語処理研究会，66-1, pp. 1-7 (1988).
5) 野村直之，高橋一裕：3軸モデルによる慣用表現の分類，第41回情報処理学会全国大会論文集，3-75 (1990).
6) Church, K. W. and Hanks, P.: Word Association Norms, Mutual Information, and Lexicography, *Proc. of ACL-89*, pp. 76-83 (1989).
7) 加藤真人，相沢輝昭：外電ニュースの定型文抽出とその英日機械翻訳，情報処理学会自然言語処理研究会，93-2, pp. 7-14 (1993).
8) 北　研二，小倉健太郎，森元　逞，矢野米雄：仕

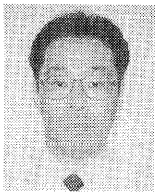
事量基準を用いたコーパスからの定型表現の自動抽出，情報処理学会論文誌，Vol. 34, No. 9, pp. 1937-1943 (1993).
9) 長尾 真，森 信介：大規模日本語テキストのnグラム統計の作り方と語句の自動抽出，情報処理学会自然言語処理研究会，96-1, pp. 1-8(1993).
10) 松本裕治：頑健な自然言語処理へのアプローチ，情報処理，Vol. 33, No. 7, pp. 757-767 (1992).

(平成6年 3月30日受付)
(平成6年10月13日採録)



新納 浩幸（正会員）

1961年生．1985年東京工業大学理学部情報科学科卒業．1987年同大学大学院理工学研究科情報科学専攻修士課程修了．同年富士ゼロックス，翌年松下電器を経て，1993年4月より茨城大学工学部システム工学科助手，現在に至る．自然言語処理の研究に従事．人工知能学会，言語処理学会，ACL 各会員．



井佐原 均（正会員）

1954年生．1978年京都大学工学部電気工学第2学科卒業．1980年同大学大学院工学研究科電気工学専攻修士課程修了．同年通商産業省電子技術総合研究所入所．現在同所知能情報部自然言語研究室主任研究官．主たる研究テーマは，自然言語処理，知識表現，機械翻訳など．日本認知科学会，人工知能学会，ACL などの会員．