自 然 言 語 処 理 104－6
（1994．11．17）

# 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

新納 浩幸

茨城大学 工学部 システム工学科

本論文では，定型的文末表現をプレーンなテキストから自動抽出する手法を提案する．本手法は完全な字面処理であるために，頑健性，実施容易性に優れている．

本手法ではある文字列が定型的かどうかを判定するために，その文字列に対する２つのグラフを利用する．１つは，その文字列を読み進める場合に，次に現れる文字がどの程度の確率で定まるかの変移を表したものである．もう１つは，先の場合の逆で，その文字列の後ろの文字列が定まると前置される文字がどの程度決定されていくかを見たグラフである．

# Automatical extraction of frozen patterns appeared in sentence end by connect probability between characters and strings

Hiroyuki Shinnou
shinnou@lily.dse.ibaraki.ac.jp

Ibaraki University.
Dept. of Systems Engineering
4-12-1, Nakanarusawa-cho, Hitachi-shi, Ibaraki-ken 316, Japan

This paper presents a new method to extract frozen patterns automatically from a Japanese corpus without morphological analysis.

In order to judge whether a target string is a frozen pattern or not, we use two graphs for the string. One is the graph, which shows stochastic of appearance a certain character (A) in back of the string ( $\alpha$ ), where $\alpha$ is a part of the target string ( $\alpha$ A*). Another is the graph, which shows stochastic of appearance a certain character (B) in front of the string ( $\beta$ ), where $\beta$ is a part of the target string (*B $\beta$ ).

## 1 はじめに

自然言語処理では，単語間の共起性が強い表現に対して，その表現を形態素に分解して処理せずに，1語として捉えた方が実用的である場合が多い．代表的な例が慣用表現である．一般に慣用表現は個々の構成語から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 [7]．テンプレートを用いた翻訳も，この考え方を発展させたものと捉えられる [2]．また外国語習得の面でも，共起性の強い表現を単語のように，1つの概念に対応する固定した文字列として捉え，それらを記憶しておくことが効果的である．その他，音声認識，ＯＣＲにも，共起性の強い表現を記憶しておくことが，そこでの曖昧性の解消に役立つことが知られている [1][3]．

このように単語間の共起性が強い固定的，定型的な表現を1つの固まった単語のように捉えることには意味があるが，その収集は困難である．なぜなら，それら表現の定義は曖昧なため，個々の表現に対して人間の判断が必要となり，その収集には膨大な時間と手間がかかるからである [8]．また人手による収集では，その網羅性，一貫性などの問題点もある．

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 [7][1][6][9]，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．このため，日本語に関しては，既存の実験を別の環境で再現することが比較的困難となっている．

本論文では，定型的文末表現を対象に，プレーンなテキストからの完全な字面処理による自動抽出の手法を提案する．定型的文末表現とは「なければならない」のような助動詞相当の慣用表現や「も少なくない」などの文末に現れる定型的な表現である．多くの応用分野で，この種の表現を1語として記憶しておくことは有効である．

本論文ではある文字列が定型的かどうかを判定するために，その文字列に対する2つのグラフを利用する．1つは，その文字列を読み進める場合に，次に現れる文字がどの程度の確率で定まるかの変移を表したものである．もう1つは，先の場合の逆で，その文字列の後ろの文字列が定まると前置される文字がどの程度決定されていくかを見たグラフである．この2つのグラフを利用することによって，任意の文字列に対して，その文字列内の単語間の共起が一定以上の強さを持っているかどうかを判定でき，一語として捉えられる定型表現を抽出できる．

ただしそれら2つのグラフを作成するには，多大な計算を必要とするために，任意の全ての文字列に対して，その文字列に対するグラフを作成することは現実的ではない．本論文では定型表現を特に定型的文末表現に限定し，予め定型的文末表現の候補となる文字列を絞り込み，それら文字列に対してだけ，上記の2つのグラフを作成する．定型的文末表現に限定することで，いくつかの言語的なヒューリスティックスも利用できるようになり，現実的に可能な処理となっている．

最後に本手法の有効性を確かめるために，小規模のコーパス (約6万文，約40Mbyte) を用いた抽出実験を行なう．この結果についても述べる．

## 2 定型表現中の接続割合の変化

文字列 $\alpha = a_1 a_2 \cdots a_n$が定型表現かどうかの判定のために，以下のような条件付き確率を考える．

$$P_1 = P(a_2 \mid a_1)$$

$$P_2 = P(a_3 \mid a_1 a_2)$$

$$P_3 = P(a_4 \mid a_1 a_2 a_3)$$

$$\cdots$$

$$P_{n-1} = P(a_n \mid a_1 a_2 \cdots a_{n-1})$$

$P_{k-1} = P(a_k \mid a_1 a_2 \cdots a_{k-1})$ は文字列 $a_1 a_2 \cdots a_{k-1}$ の直後に $a_k$ の現れる確率を表す．横軸に文字の位置 i，縦軸に確率 $P_i$ をとったグラフを，文字列$\alpha$に対する順方向確率変移グラフと呼ぶことにする．

意味のある文字列（あるいはその一部）の順方向確率変移グラフは概ね右上がりのグラフになるが，所々に下降する部分もある．下降した地点（極小値を与える地点）は，形態素の切れ目の部分である．

例えば,「話をきいたことがある」のグラフは図1のようになる．極小値になっている部分を見ると「話しを」の後,「話しをき」の後,「話しをきいたことが」の後となっており，それらは次に活用の変化や自立語が現れる部分であり，次の文字には多数の種類の文字が現れることができるために確率が低くなっている．

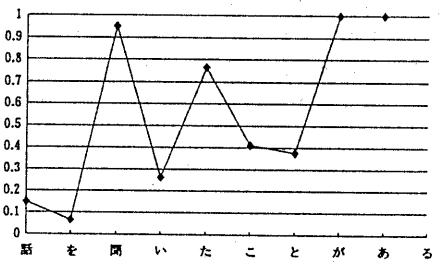


図 1:

定型表現の最も基本的な性質はその内部の単語間の結合力の強さである．今，2つの単語$\alpha$と$\beta$からなる定型表現 $\gamma = \alpha\ \beta$ に対する上記グラフを考える．通常，グラフは$\alpha$と$\beta$の切れ目で下降する．しかし，定型表現では単語間の結合力が強いために，下降した場合でもその値はある一定値以上の大きさをとる．これはある（一連の）言葉がくれば，ほとんどの場合，後ろの言葉が決まってくるような定型表現に相当している．例えば,「なければならない」の順方向確率変移グラフを図2に示す．通常，文節の後でグラフは下降するので,「なければ」の後でグラフは下降している．しかし「なければ」の後には多くの場合「ならない」が現れるために，下降してはいるが，その値は高い値になっている．

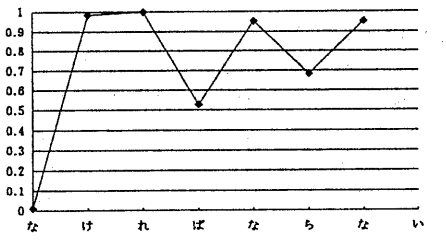


図 2:

本論文では文字列$\alpha$に対する順方向確率変移グラフの各々の地点の値が，ある一定以上の値をとれば，それは定型表現であると判定することにした．

また定型表現には後続する単語が前置する単語を規定する場合もある．このような定型表現は，先の判定法では抽出できない．例えば「に関する」という定型表現を考えてみる．助詞「に」の後ろには様々な単語が現れるために,「に関する」の順方向確率変移グラフでは,「に」の地点で非常に小さい値をとる．このため先の判定法ではこの文字列は棄却されてしまう．

そこで本論文では逆向きの確率変移グラフを作成し，後ろの文字列から前の文字の現れる確率を調べることにした．まず文字列 $\alpha = a_1 a_2 \cdots a_n$ に対して，以下のような条件付き確率を考える．

$$P'_{n-1} = P'(a_{n-1} \mid a_n)$$
$$P'_{n-2} = P'(a_{n-2} \mid a_{n-1} a_n)$$
$$\cdots$$
$$P'_2 = P'(a_2 \mid a_3 a_4 \cdots a_n)$$
$$P'_1 = P'(a_1 \mid a_2 a_3 a_4 \cdots a_n)$$

$P'_k = P'(a_k \mid a_{k+1} a_{k+2} \cdots a_n)$ は文字列 $a_{k+1} a_{k+2} \cdots a_n$ の直前に $a_k$ の現れる確率を表す．横軸に文字の位置 i，縦軸に確率 P'i をとったグラフを，文字列$\alpha$に対する逆方向確率変移グラフと呼ぶことにする．このグラフを利用して，逆向きの単語間の結合力の強さを計ることが出来る．

例として「に関する」の順方向確率変移グラフと逆方向確率変移グラフを図3に示す.「に関する」は,「に」の後ろには様々な単語が現れるために，その地点の順方向確率変移グラフの値は非常に小さい．しかし,「関する」に前置する単語はほぼ「に」になるために，逆方向確率変移グラフでは，その地点で大きな値を取っている．

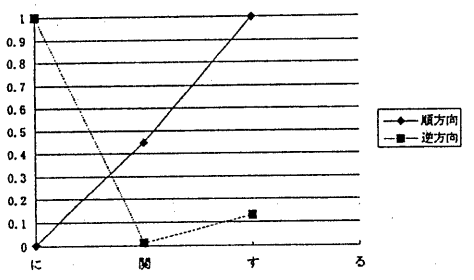


図 3:

本論文では文字列$\alpha$に対する順方向確率変移グラフのある地点の値がある一定以上の値をとらなかった場合でも，その地点の逆方向確率変移グラフの値を参照し，その値がある一定以上の値の値であれば，それは定型表現であると判定することにした．

## 3 定型的文末表現の自動抽出

### 3.1 候補の選出

基本的に前節で説明した判定法を用いて，ある文字列が定型表現になるかどうかを判定する．ここで判定を行なう文字列をコーパス中の任意の文字列にしてしまうと，計算量が膨大になる．このため，本論文の手法を用いる場合，抽出する定型表現のタイプを限定して，適当な数にまで候補の文字列を絞っておく方が現実的である．そのため，本論文では定型的文末表現を対象にする．

概ね文末表現は文末に現れるので，コーパス中の全ての文から，後ろ n 文字 (n = 3 ～ 20) の文字列を取り出し，これらを定型的文末表現の候補の文字列とする（図 4 参照）．

```
N = 20   割してから，構文解析を行わなくてならない
N = 19     してから，構文解析を行わなくてならない
N = 18       てから，構文解析を行わなくてならない
   • • • • • •
N =  5                              てならない
N =  4                               ならない
N =  3                                らない
```

図 4:

これらの候補も非常に数が多いので，妥当なヒューリスティックスを導入して候補を少なくする．ここでは以下の 2 つのことを行なう．

- 出現頻度が一定数（ここでは 5 とした）以上に達しないものは，候補から削除する．
- 文末表現には読点や括弧などの記号は入らないとして記号を含む文字列は，候補から削除する．

### 3.2 確率変移グラフの作成と定型表現の判定

前章で求めた候補に対して，それぞれ順方向及び逆方向確率変移グラフを作成する．まず順方向のグラフから作成する．

文字列$\alpha = a_1 a_2 \cdots a_n$ が先の候補となっている場合，グラフを作成するために必要となる条件付き確率は

$$P_1 = P(a_2 \mid a_1)$$

$$P_2 = P(a_3 \mid a_1 a_2)$$

$$P_3 = P(a_4 \mid a_1 a_2 a_3)$$

$$\cdots$$

$$P_{n-1} = P(a_n \mid a_1 a_2 \cdots a_{n-1})$$

の n − 1 種類である．このように予め計算しなければならない条件付き確率を候補の文字列に対して全て用意し重複したものを省く．その後にコーパスからそれぞれの条件付き確率を求める．

各々の条件付き確率は以下の式で求められる．

$$P(a_k \mid a_1 a_2 a_3 \cdots a_{k-1}) = \frac{I(a_1 a_2 a_3 \cdots a_{k-1} a_k)}{I(a_1 a_2 a_3 \cdots a_{k-1})}$$

ここで $I(a_1 a_2 a_3 \cdots a_m)$ は文字列 $a_1 a_2 \cdots a_m$ がコーパス中に現れた回数である．

次に逆方向確率変移グラフであるが，これは候補及びコーパスを逆順にすれば上記と同じプログラムを用いて算出できる．具体的に説明すると，まず前章で選出した候補を全て逆向きに並べ直す．

「なければならない」 $\Longrightarrow$ 「いならなばれけな」

これらの候補に対して前述した計算すべき条件付き確率の種類を求め，重複したものを省く．次にコーパスを 1 本の長い文字列と考え，それを反転させたコーパスを作成する．

「私は今日学校へ行きました．そして勉強をしました．..... 面白かったです.」
$\Longrightarrow$ 「. すでたっか白面..... たしましを強勉てしそ．たしまき行へ校学日今は私」

この反転させたコーパスを用いて先の条件付き確率を求めればよい．

上記の処理によって得られた各候補の文字列に対する順方向確率変移グラフと逆方向確率変移グラフを用いて，その候補が定型表現かどうかを判定する．判定は前節で説明したように，以下の条件 (a)(b) のどちらかが成り立っていれば，定型表現と判定する．

(a) 順方向確率変移グラフの各地点の値は設定値 A 以上である．

(b) 順方向確率変移グラフのある地点の値が設定値 A 以下であっても，その地点の逆方向確率変移グラフの値が設定値 B 以上である．

本論文では設定値A，設定値Bともに 0.3 としている．この値は動詞の屈折を考慮した経験的な値である．例えば「に関して」を考えると「に関」を読んだだけでは，「に関して」や「に関する」などあるため，少なくとも「し」と「す」が後続する可能性が同じくらいに高く，接続の確率は高くても 0.5 程度である．さらに「に関し」までを読んでも「に関して」の他に「に関し，」「に関した」などもあるために接続の割合は高くても 0.4 程度である．このように共起性が非常に強い定型表現であっても文字レベルで見ると，屈折などが起きそれほど大きな値にはならない場合がある．本論文では次に現れる文字は 3 種類程度は許すという考え方から設定値A，設定値Bを 0.3 とした．

### 3.3 包含関係からの選択

上記のまでの処理では，包含関係にあるような文字列も取り出してしまう．例えば以下のような文字列を同時に取り出してしまう．

「なければならない」
「ければならない」
「ればならない」
「ばならない」

このように連続した包含関係になっている文字列からは，その中から1つだけを選択することにする．ここでは文字列の長さが最も長いものか2番目に長いものを選択することにする．

文字列の長さが最も長いものと2番目に長いものに注目すると，包含関係になる原因は大きく2つの場合がある．1つは先頭が助詞でありその助詞を含めるものと含めないものがでる場合である．

例）　○　「に違いない」
　　×　「違いない」

もう1つは動詞の活用語尾を含んだものと含んでいないものが出てしまう場合である．

例）　×　「ることもある」
　　○　「こともある」

一般に助詞がついている場合には助詞がついている方を選びたい．つまり長い方を選びたい．一方，動詞の活用語尾がついているものはついてない方を選びたい．つまり短い方を選びたい．

この選択には逆方向確率変移グラフを利用する．先頭の文字が活用形の語尾の場合は，いくつかの文字種が現れることができる．例えば連体形であれば「(す) る」や「(し) た」など，そのために先頭位置の逆方向確率変移グラフの値は小さい．逆に助詞の場合には，定型表現の一部であるために一定以上の値を持っている．このことから包含関係にあるような文字列からは，最も長い文字列の先頭位置の逆方向確率変移グラフの値を見て，それが一定値（ここでは 0.5 とした）以上であれば長い方を選択し，そうでなければ，1つ短い方の文字列を選択することにする．

ここで設定した値 0.5 も経験的なものである．ここで計算する条件付き確率は条件がきつくなるほど，1 に近付くので，先の設定値A，Bよりも若干高めに設定した．

## 4 実験とその評価

### 4.1 実験

本手法の有効性を確認するために，小規模のコーパス（新聞の社説などテキスト約6万文 40Mbyte）を利用して，定型的文末表現の抽出実験を行なった．

まず定型的文末表現の候補として，コーパスの各文の文末から 3 ～ 20 文字分の文字列を取り出す．単純には1文に対して 18 種類の文字列が抽出できるので，その総数は高々108 万種類（文の総数 × 18）である．重複した文字列を持つこともあるために，実際は，835,542 種類の文字列が取り出せた（選別0）．次にこれらの中から出現頻度が5 以下のものは省いた（選別1）．この結果，5,983 種類の文字列が残された．更にこれらの中から，読点や括弧などの記号を含むものを除いた（選別2）．この結果，5,586 種類の文字列が残された．次にこの 5,586 種類の文字列の文字列に対して，順方向確率変移グラフと逆方向確率変移グラフを作成し，これらグラフを用いて定型表現でないものを省いた（選別3）．その結果，922 種類の文字列が残された．最後に包含関係にある文字列から妥当なものを選択し，その他を省いた（選別4）．以上によって 484 種類の文字列を，定型的文末表現として抽出した．選別1から選別4における文字列の数の変化の様子を表1に示す．

また順方向確率変移グラフと逆方向確率変移グラフとの面積を (文字列の長さ − 1) で割った値は，概略，接続関係の強さを示しているので，その値の大きいものから，最終的に抽出できた表現の上位 30 個を表2に示す．

| 文字列長 | 選別 0 | 選別 1 | 選別 2 | 選別 3 | 選別 4 |
|---|---|---|---|---|---|
| 3 | 11,393 | 1,205 | 1,128 | 87 | 13 |
| 4 | 22,734 | 1,603 | 1,509 | 204 | 127 |
| 5 | 35,455 | 1,368 | 1,286 | 183 | 89 |
| 6 | 44,835 | 924 | 881 | 195 | 114 |
| 7 | 51,664 | 487 | 454 | 122 | 59 |
| 8 | 55,088 | 260 | 224 | 74 | 45 |
| 9 | 56,486 | 100 | 77 | 39 | 24 |
| 10 | 56,645 | 30 | 24 | 15 | 11 |
| 11 | 56,065 | 6 | 3 | 3 | 2 |
| 12 以上 | 445,177 | 0 | 0 | 0 | 0 |
| 合計 | 835,542 | 5,983 | 5,586 | 922 | 484 |

表 1:

| 順位 | 表現 | 確率の平均 |
|---|---|---|
| 1 | をご存知だろうか | 0.685590 |
| 2 | と専門家はいう | 0.649520 |
| 3 | 花を咲かせている | 0.642423 |
| 4 | 便りをいただいた | 0.642278 |
| 5 | なければならない | 0.640850 |
| 6 | にはどうしたらいいのか | 0.640688 |
| 7 | なければならぬ | 0.633443 |
| 8 | ではないでしょうか | 0.628690 |
| 9 | るわけにはいかない | 0.624290 |
| 10 | がきこえてくるようだ | 0.621174 |
| 11 | ようなところがある | 0.614348 |
| 12 | ることに変わりはない | 0.614253 |
| 13 | とものの本にある | 0.608378 |
| 14 | が必要なのではないか | 0.603975 |
| 15 | してもいいのではないか | 0.600966 |
| 16 | に取り組んでいる | 0.599599 |
| 17 | のではあるまいか | 0.597057 |
| 18 | 尋常ではない | 0.592254 |
| 19 | 逮捕された | 0.591822 |
| 20 | が紹介されている | 0.590314 |
| 21 | という表現があった | 0.588514 |
| 22 | がおもしろかった | 0.588045 |
| 23 | 息をひきとった | 0.587512 |
| 24 | れるゆえんだ | 0.587298 |
| 25 | いるのではあるまいか | 0.586241 |
| 26 | ているような気がする | 0.584164 |
| 27 | かもしれない | 0.583982 |
| 28 | すぎるのではないか | 0.583963 |
| 29 | ているようにみえる | 0.583899 |
| 30 | が寄せられている | 0.583586 |

表 2:

### 4.2 評価

ある文字列が定型表現になっているかどうかを客観的に判定することは非常に困難である．ここでは主観的な判断によって抽出結果についての評価を行なうことにした．

まず抽出したものが形態的に文末表現にならない場合を調べた．以下のような場合がある．

1. 活用語尾がついたままである．(33 種類)

   例えば，「るかも知れない」や「るわけにはいかない」などが取り出されているが，これらは本来は「かも知れない」や「わけにはいかない」という形で取り出したかったので，これらは不正解とした．

2. その文字列自身が 1 語である．(24 種類)

   例えば，「聞こえる」や「興味深い」などは，その文字列自身が 1 語であり定型表現ではない．本手法は固定的な文字列を取り出そうとするために，この種の文字列が抽出されることは避けられらい．

3. 形態素の途中で切られている．(2 種類)

   これらは「０００万円」「たしてそうだろうか」の 2 種である．後者が「はたしてそうだろうか」として抽出できなかったのは，コーパス中に「はたしてそうだろうか」と「果たしてそうだろうか」の 2 種類が存在し，各々の出現頻度は 5 に満たないが，両者の合計が 5 を越えたためである．

抽出した 484 種類中，上記の 59 種類が形態的に文末表現になっていない．この点では 87.8 % の正解率と考えられる．

次に上記の形態的に正しい文末表現のうち，その表現を構成している形態素の数を調べた．形態素の数が多いほど1語として扱う効果が高いと思われる．平均して3.41個の形態素から構成されていた．

最後に形態的に正しい文末表現を，主に英訳する場合を想定して，以下のように分類した．

1. 助動詞あるいは補助動詞的なもの．(38種類)

   例えば，「なければならない」「かもしれない」「べきだ」「ている」「てほしい」「て下さい」「た方がいい」「傾向がある」「にすぎない」などがこのタイプである．

2. 定型的表現と考えられるもの．(150種類)

   例えば，「と望みたい」「ではあるまい」「がものをいう」「というほかない」「というわけだ」「といわれている」「のはなぜだろう」「に思えてならない」などがこのタイプである．第1のタイプとは区別が微妙なものもあるが，英訳する場合を想定して，別表現に置き換えたり，比較的決まりきった言い方がある場合をここの分類とした．

3. 上記以外．(237種類)

   例えば，「がめだつ」「が起きた」「が始まった」「が贈られた」などである．このタイプは，ほとんどが「助詞＋動詞句」の形であり，英訳する場合に，特に一語として扱わなくともよいと思われるものである．ただし第2のタイプとの区別が微妙なものもある．

1.や2.のタイプの表現を定型表現と捉えても良いと考えると，抽出した総数484種類のうち，正解は188種類で，正解率は38.8％である．形態的に不備なものは，辞書を用いれば機械的に排除できるので，それらを除いて考えれば，44.2％の正解率である．本手法が完全な字面処理と，完全な自動抽出という点を考慮すれば，この程度でも良好な結果だと判断している．また分類上の3.のタイプは，1つの単語として扱わなくても処理はできるが，効率上は1つの単語として扱った方が良い場合も多いと思われ，この部分の判定によっては正解率が向上する．

## 5 考察

本手法は基本的にコーパス中のNグラム ($N \leq 20$) をとることができれば，順方向確率変移グラフや逆方向確率変移グラフはともにそのNグラムから計算できるために，Nグラムを使った手法とも考えられる．Nグラムを効率的に作成する手法も考案されているので[4]，そのアプローチからも本手法は実現できる．ただし，Nグラムを作成した後でどのように定型表現を抽出するかは未解決の問題であり，本手法はそのための1つのアプローチを示したものとも考えられる．

また，本手法は抽出する表現を定型的文末表現に限定しているために，完全にNグラムを作る必要がない分，効率的である．更に抽出する表現を限定することで，いくつかの言語的なヒューリスティックスも利用でき，最初の候補の選出の段階で相当小さな数にまで候補を減らすことができる．例えば「付属語は平仮名から構成され，漢字が含まれていてもその長さは1である」[5]や，「ある表現の直前に漢字や読点が現れれば，それは文末表現ではない」などのヒューリスティックスも有効に利用できると予想している．

定型表現を抽出する手法としては，仕事量基準という観点からの研究もある[7]．これはその表現を1まとまりと予め考えておくことが，その表現の各構成語からその表現全体の構造をを作り出す場合と比べて，どの程度処理の負担（仕事量）が軽減されるかに注目して，定型表現かどうかを判定する．この手法は言語的に妥当な定型表現を抽出することが目的ではないので，言語的なヒューリスティックスを用いない点や定型表現の判定に共起性と同程度に出現頻度を重視する点が，本手法やNグラムを使った手法とは異なっている．また形態素解析も必要とされ，実施の容易性が本手法やNグラムの手法と比べるとやや低い．しかし，定型表現の客観的な評価基準を与える試みであり，仕事量基準という観点は，本手法で抽出した表現を評価する際の1つの尺度になると予想している．

特に，1語として扱う場合の処理の軽減という考え方をとれば，文字列の長さの長い方を，より優先的に取り出す手法が望ましい．仕事量基準でもその定義式から明らかなように，文字列（この場合は単語数）の長い方が基準値が高くなる設定になっている．この点は，本手法でも実現できている．文字列に対する順方向確率変移グラフや逆方向確率変移グラフの面積を文字列の長さで割った値は，ほぼ文字列の長さが長いものの方が大きな値をとる傾向があるからである．これは文字列の長さが長い方がより後ろの部分の文字が現れる確率が1に近付いてい

くからである.

また本手法の計算のほとんどは，順方向確率変移グラフと逆方向確率変移グラフを作成するための多数の条件付き確率を求める処理に費やされる．本論文では，条件付き確率を求める場合，複数の文字列を検索式としてテキストの1度の走査で文字列一致を判定するアルゴリズム (AC 法)[10] を改良して用いている．これによってコーパスを1度だけ（逆方向もあるため厳密には2度）走査するだけで，目的の条件付き確率が一度に求められている．

最後に本手法の実施容易性について述べる．近年，コーパスからの知識獲得の研究が盛んだが [11]，そこにはタグ付けされたコーパス，精巧なパーザ，充実した辞書などが実際には必要である場合が多く，その実験を別の環境で実施したり，実験を拡大したりすることが困難な場合が多い．コーパスからの知識獲得では大規模なコーパスが必要であるため，できるだけ生のままのテキストを対象にした方がよい．また形態素解析などは，処理自体にまだ不備な面があることや，それを解析するための辞書によっても微妙に結果が異なることから，汎用的な手法とはいい難い．本手法のように字面処理だけによるものは，生のコーパスさえ用意すれば，どこでも同じように実験の再現が容易に出来るという点で優れていると考える．字面処理だけによるために精度的には問題があるが，これも．1次的な候補の抽出程度に捉えたり，人間の判断が介入することを妥協したりすることで，ツールとしても有効に利用出来ると考える．

## 6 おわりに

本論文では文字列を順方向，あるいは逆方向に読み進める際に，次に現れる文字がどの程度予想できるかを示した確率を利用することで，定型表現を抽出する手法を提案した．特に定型的文末表現に限定することで，予め候補を絞り，確率を計算する処理を軽減した．最後に実験によって本手法の有効性を確認した．

本手法は完全な字面処理のため，容易に実現可能であるし，実験の再現や拡大も容易であるという特徴を持つ．

今後は本手法を使って，別種類の定型表現の自動抽出も試みたい．また構成単語が離れた位置に現れるギャップを持つような定型表現についても考察したい．

## 参考文献


[1] Church,K.W. and Hanks,P. : "Word Association Norms, Mutual Information, and Lexicography", Proc. of ACL-89,pp.76-83 (1989).

[2] Furuse,O. and Iida,H. : "Cooperation between Transfer and Analysis in Example-based Framework", Proc. of COLING-92, pp.645–651(1992).

[3] Kita,K., Omoto,T., Yano,Y., and Kato,Y. : "Application Corpora in Second Language Learning – The Problem of Collocational Knowledge Acquistion", Proc. of Second Annual Workshop on Very Large Corpora, pp.43–56(1994).

[4] Nagao,M. and Mori,S. : "A New Method of N-gram Statistics for Large Number of n and Automatic Extraction of words and Phrase from Large Text Data of Japanese", Proc. of COLING-94, pp.611–615(1994).

[5] Shinnou,H. and Ishahara,H. : " Automatically Extracting Simple Auxiliary Phrase from a Corpus", Proc. of Natural Language Processing Pacific Rim Symposium, pp.136–143(1993).

[6] 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).

[7] 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).

[8] 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).

[9] 新納浩幸, 井佐原均 : " コーパスからの関係表現の自動抽出", 情報処理学会論文誌,Vol.35,No.11,to appear (1994).

[10] 高橋恒介 : "『テキスト検索プロセッサ』– 第2章 文字列検索のソフトウエアアルゴリズム", pp.21–58, 電子情報通信学会, 東京 (1991).

[11] 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).