自然言語処理 96－1
（1993．7．9）

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

長尾 眞　森 信介
京都大学工学部 電気工学第二教室

要旨

シャノンによる情報理論の確立により，自然言語をマルコフ過程としてとらえ，言語のもつ性質を明らかにしようという立場が提案された．この立場は，ある n 文字の組合せがどのような頻度で生じるか(nグラム)を調べることにその中心があったが，計算機の性能やテキストデータの不足などにより、大規模なテキストに対して、あるいは大きな n に対して言語統計を取ることが行われなかった．我々は，今日の計算機を用いるとこれが実現できると考え，大規模なテキストの任意の n についての n グラムを簡単にとる方法を考案し，200万文字から3000万文字の中規模の日本語テキストデータに対し，nグラム統計をワークステーションを使って比較的短時間でとることに成功した．その結果，種々の n に対する n グラム統計を比較して調べることによって言葉として有意義なものが取り出せるということが明らかになった．同時にさらに大きいテキストを用いることの必要性と、可能性が明らかになった．

A method of n-gram statistics for large text data of Japanese, and the automatic extraction of words and phrases.

Makoto Nagao　Shinsuke Mori
Department of Electrical Engineering, Kyoto University

Abstract

In the process of establishing the information theory, C.E.Shannon proposed the Markov process as a good model to characterize a natural language. The core of this idea is to examine the frequency of a string composed of n characters (n-grams), but this statistical analysis of large text data and for a large n has never been carried out because of the low capability of computer and the shortage of text data. Taking advantage of the recent powerful computers to execute it, we developped a new algorithm of n-grams of large text data for arbitrary n and calculated successfully, within relatively short time, n-grams of some middle size Japanese text data containing between two and thirty million characters. From this experiment it became clear that the automatic extraction or determination of words is possible by mutually comparing the n-gram statistics for different values of n.

1　はじめに

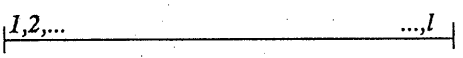
シャノンが1948年に情報理論を確立し，言語をマルコフ過程としてとらえる立場が提案された(1)．そして一つの言語におけるアルファベット文字がどのような現れ方をするかが関心の的となった．すなわち，あるn個の文字の組み合せ（nグラム）がどのような頻度で生じるかを調べることで，言語の持つ性質が明らかにできるという期待であった．

しかし今日までこの考え方を具体的に実現することには成功していない．それは1つには大量の言語データを集め，計算機に記憶させることが難しかったこと，もう1つは，nグラム統計をとるとき$n > 3$となると，必要とする記憶容量が膨大となって計算機で扱えなくなってしまうというところにあった．

われわれは，今日のテキストデータ量と計算機の記憶容量からすれば，この困難をかなり軽減し，いろいろと面白い結果が得られるのではないかと考え，nグラム統計を取ることを試みた．本論文では，長さnに依存しないnグラム統計を取る新しい方法を示し，いくつかの中規模テキストデータに対してnグラム統計をとった結果，文章中に現れる単語というものを新しい観点から定義できる可能性があることを示した．

2　nグラム統計における必要記憶容量の推定

今日ではかなりの量の言語データが計算機の処理対象として利用可能となっている．言語統計を取ろうとするとき，これをどの様に記憶するのがよいかを検討することは大切なことである．われわれは，オクスフォード英語辞典（約3.5億文字）を計算機に入れて検索可能としたウォータルー大学の取った方式を採用することにした(2)．これは図1に示すように，テキスト全体を1列の文字列と見なすものである．これをストリング記憶と名付けておく．オックスフォード英語辞典の場合は，長さが約$l = 350,000,000$の1つの文字列である．これに要する記憶容量は日本語の場合$2l$バイトである．



l文字からなる文章．
途中に句読点やスペースなども含まれている．

図1　テキストのストリング記憶

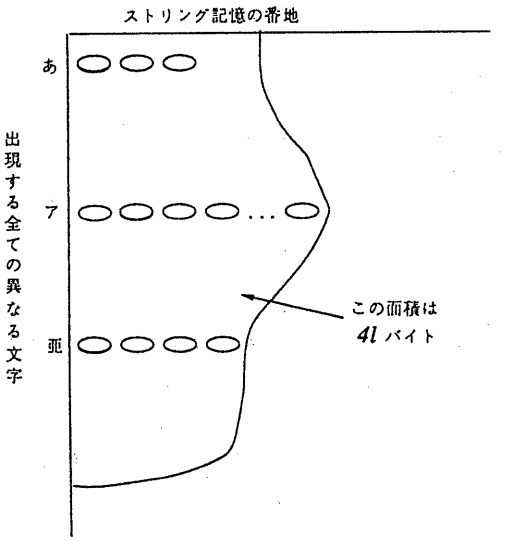


図2　文字の出現位置と頻度を表わす表

次にこの文字列に現れる各文字の出現頻度（1グラム）を求めることを考えよう．図2に示すような2次元マトリックスを考える．縦軸には出現する可能性のあるあらゆる文字を取る．そしてある文字（例えば「あ」）が図1のストリング中のあちこちに現れるとして，それらの位置を表わす数値を並べる．このようにすると横軸方向の長さは不定となるが，マトリックス中に書き込まれる数値の個数はストリングの長さ（文字数）lに等しい．即ち，このマトリックスの所要面積はlで押さえられる．我々の場合，取り扱うべきテキストデータの量は数億文字以上を考えねばならないので，文字の位置を表現するためには少なくとも30ビット必要となり，4バイト（32ビット～40億文字）を取ることにした．従って，図2の出現頻度を調べるための必要記憶容量は$4l$バイト程度となる．2次元の矩形のマトリックスを用意せず，ポインターをうまく用いることによって実質的にこの程度の記憶容量で押さえることが可能である．

次に2グラム統計を取ることを考えよう．この場合は図3に示すようにマトリックスの縦軸は出現する全ての隣接2文字組をとる．ある2文字組がストリング中に現れる位置をその2文字組の行に記入して行く．この場合も横方向の長さは不定であるが，重要なことはマトリックス全体として必要な面積は$4l$バイトとなることである．同様のことは一般のnグラム統計のための表についても

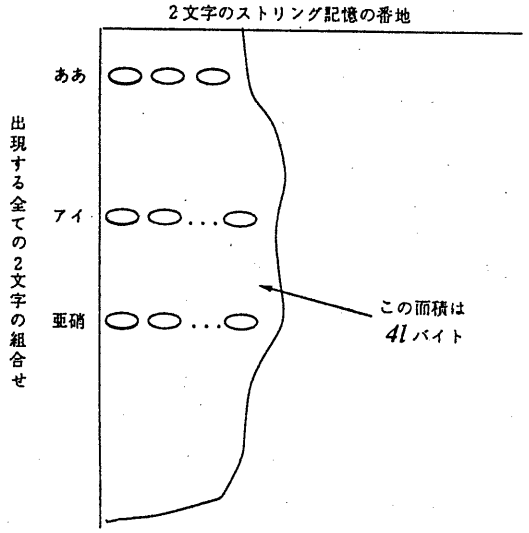


図3　2グラムの出現位置の表

言うことができる.

以上の議論から言えることは，長さ l の日本語原テキストを記憶するのに $2l$ バイト，nグラムまでの全ての統計を取るのに $4l \times n$ バイトを必要とし，合計で $2l + 4l \times n = 2l(1 + 2n)$ バイトが必要となるということである．例えば，1000万文字の文章の場合に10グラム統計までを取ることにすると，420MBの容量で済むことになる．これは今日ワークステーションで可能な数値であり，かなりの量の詳しい言語統計をこのクラスの計算機で計算できることを意味する．

我々は，この考え方をさらに発展させ，さらに少ない容量で任意の n に対するnグラム統計を比較的簡単に作り出す方法を考案した．これを次節に述べる．

3　nグラム統計を取るアルゴリズム

長さ l の1本の長い文字列(ストリング)の中に特定の n 個の隣接した文字列が何回現れるか，どこに現れるかを知るために，その n 文字列でストリングを走査するというのは，あらゆる可能な n 文字列についてそれを行なわねばならないということを考えると，得策ではない．そこで次のような新しい方法を考案した．

まず，ストリングの i 文字目から最後の文字(l 文字目)までを1つの文字列と見なし，$i = 1$ から $i = l$ までの l 個の文字列の存在を仮定する（実際は図1のように1本の文字列である）．i 文字目から始まる文字列を i 番目

ポインター	1	2	3	4	5	6	7	8	9	10	11	12	13
原ストリング	あ	る	所	へ	あ	る	日	会	い	に	ゆ	く	。
ソート結果	13	5	1	9	12	10	4	11	6	2	7	8	3

図4　文字列のソート例

部分文字列	A 欄(一致文字数)
ある日ある所に…	3
ある日会いにゆく。	6
ある日会いにいった。	2
あるのは…	1
あれは…	

図5　一致文字数の図

文字列と呼ぶことにする．この l 個の文字列を辞書順でソートする．その大小比較は i でポイントされた図1の同一のデータを用いる．ソートにはバブルソートの改良版であるコムソートを用いた．そのソーティングの時間は $O(l \log l)$ である．ソートされた結果は文字列の先頭番地の配列順序によって示されている．その簡単な例を図4に示す．各文字のストリング記憶中の位置を示す為に4バイト取っているので，このソートされたデータの記憶容量は $4l$ バイトである．

このようにソートされ辞書式順序に並んだ文字列において，隣接する2つの文字列が先頭から何文字目まで同じ文字であるかを調べ，これを2つのうちの前の文字列の情報として付ける．例を図5のA欄に示す．この一致文字数を示すために1バイト用意し最大255文字までの同一性(nグラムで $n = 255$ まで)を表現できるようにした．このA欄には文字列全体で l バイト必要となる．

A欄の1バイトの中に書かれている数字を見ることによって，任意の n に対してnグラム表は簡単に作れる．n をある値に固定し，ソートされた文字列の表(辞書式配列になっている)を上から見て行く．まず最初の文字列について先頭から n 文字を切り出し，それが以下に連続する幾つの文字列に共通に存在するかを一致文字数を示すA欄を見て行くことによって得ることができる．即ち，A欄の値が初めて n より小になるまでの文字列の数を計数すればよい(図5参照)．そこでA欄が初めて n より小になった文字列の先頭から n 文字を取り出すと，これが新しい n 文字列となっており，その文字列の出現回数は

同様にA欄を見て行くことによって容易に計数することができる．このようにしてソートされた文字列の表から任意のnに対するnグラムを簡単に作り出すことができるのである．我々のプログラムでは$n=255$までが可能となっている．

以上の処理に必要な記憶容量は，(1)長さlの原文字列の記憶のために$2l$バイト，(2)ソートされた結果の記憶のために$4l$バイト，(3)先頭からの一致文字数を表示するためにlバイト必要となり，ここまでの処理で$7l$バイトが必要となる．

現在のところnグラムの表を作って記憶しておいてもあまり意味がないので，nグラムの表が必要となったときに，その都度上記したように一致文字数の欄を見ながらnグラム表を作るのがよいと考えている．後に述べるように我々は日本語テキストデータの特徴を調べるために，とりあえず$n=12$までのnグラムの主要な部分を打ち出して検討した．

nグラム統計の対象となるテキストの量が膨大になると1度にテキストデータ全体の処理を行なうことは不可能となる．我々が現在使っているワークステーションは64MBの主記憶容量なので，$64MB/7B \approx 600$万文字が一度に処理できる限度である．そこでそれ以上のテキスト量の時は600万文字以下のテキスト量に分割し，それぞれをソートしたあと，これらをマージすることが必要となる．ここでは2つのテキストデータのソート結果をマージすることにし，3つ以上のデータ群を一挙にマージすることは考えなかった．マージのステップは次のようになる．

- 第2のテキストのソートされた結果のデータで，文字位置を表わしている数値に第1のテキストの大きさを表わす数を加える．
- 第1，第2のソートされた文字列データの先頭から1つずつ取り出し，比較をし，どちらを先に持って来るかを決定する通常のマージの仕方でマージを行なう．
- ソート結果において隣接する2つの文字列が何文字目まで一致するかを調べる．これによって2つのテキスト文字列を合わせた文字列のソートされた結果が得られる．

4　出現頻度の高い文字列の抽出

前節で既に述べたようにソートされた文字列の表から任意のnに対するnグラムは直ちに得られるが，我々はその方向の研究よりは，出現頻度の高い文字列と我々が通常単語と称している文字列との関係がどうなっているか，出現頻度の高い文字列を取り出せば単語というものを機械的に決定できたことにはならないか，といったことに興味を持ち，次のようなことを考えた．

長さnのある特定の文字列の頻度が高く，かつその文字列の直接左，直接右に来る文字の種類が多く変化に富んでいるという場合は，その特定の文字列は，それが1つのユニットとして使われている可能性が高いので，これを抽出する．

これは，前節の処理の結果得られたソートされた文字列集合に対して比較的簡単に適用して抽出することができる．まずあるnを固定したとする．これでソートされた文字列集合の一致文字数の欄を見て行き，文字列の先頭からn文字までが同じもののうち，$n+1$番目の文字で異なった文字が何種類あるか調べる．また，文字列の先頭の文字の直前の文字についても同様に異なった文字の種類を調べる．これらがある値以上であるもののみを打ち出すのである．この時の異なった文字が幾つ以上あるべきかという閾値は経験によって決めざるをえないが，100万 ～ 200万 文字の原テキストデータの場合，表1に示すようにいくつかの閾値の組を考えた．例えば，閾値の組Aについては，$n=2$の場合は2文字列の左または右，あるいはその両方ともに10種類以上の違った文字が来る場合，その2文字列は1つの単語である可能性が高いとする．$n=3$の場合は3文字列の左右に来る文字がそれぞれ8種類以上であるという時に，この3文字列は1つの単語である可能性が高いとして取り出す．$n=6$以上のnについては閾値は$n=6$の場合と同じものに固定した．閾値の組A，B，Cの違いによる効果などは次節で説明する．

表1　実験に用いた閾値

n	2	3	4	5	6	7	...
閾値の組A	10	8	6	4	3	3	...
閾値の組B	18	13	9	6	5	5	...
閾値の組C	27	19	13	9	7	7	...

5　テキストデータの処理と結果

5.1　用いたテキスト

実験に用いたテキストは次の3種類である．

(i) 岩波情報科学辞典の全テキスト (182.5 万文字, 3.7MB)

(ii) 朝日新聞天声人語 (402 万文字, 8MB)

(iii) 研究室にある種々の電子版資料 (2936 万文字, 59MB)

(i)(ii) のテキストは比較的小規模のもので, ソートに問題はなかったが, (iii) はかなり大きな量なので, これを 4 つに分割し, 2 組ずつマージすることを 3 回繰り返して 1 つに統合した.

前節 4 に述べた単語の可能性の高い文字列の抽出を表 1 の閾値 A, B, C に対して行なってみると, その抽出語数の比は約 7.8 : 2.3 : 1 となり, 閾値 A によるものが圧倒的に多かった. 閾値 A と C の場合を比較すると意味のある文字列の抽出はどちらも同じ程度であり, C を用いた方が雑音性が少ないということができる. 閾値 A を用いた時, 岩波情報科学辞典の場合 61,408 語, 研究室資料の場合 1,054,107 語が抽出された. これらはあまりにも膨大な量であり, さらに工夫をして有効な文字列だけが出てくるようにする必要がある.

5.2 結果の検討

以上のようにして得たデータはいずれにしても膨大なものであり, それらの全てについて言語学的立場から検討し, ここで紹介することはできない. ここでは言語的に面白いいくつかの典型的な場合について述べる.

新語が浮かび上がって来た

ワープロ, 朝シャン, 過労死, ＪＲ, …

連続した活用語尾が取り出されてきた

なければならない, ざるをえなくなった, どうしたらいいのか, …

複合語

自然言語による対話システム, 湾岸戦争, バレンタインデー, …

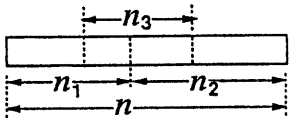
単語間の強い共起性を示すもの

「影響を受ける」, 「影響を与える」の 2 つは出て来たが, その他の結合は閾値以上の頻度で出て来なかったので, 「影響を」とくればかなり高い確率で次が「受ける」か「与える」であることが推定される. 「入退院を繰り返している」という句が成句として天声人語の文章から取り出され「入退院を」に続いては多く「繰り返す」が来ることが推定される. 「洋の東西を問わず」についても同様で「問わず」がほとんどの場合である.

表 2　抽出語句の例

出典	抽出語句の例	出現頻度
岩波情報科学辞典	シンプレックス法	44
	チューリング機械	138
	トランザクション	124
	意味ネットワーク	65
	有限オートマトン	59
	マイクロプログラム	82
	ワークステーション	69
	エキスパートシステム	100
	オペレーティングシステム	293
朝日新聞天声人語	税制改革	58
	ヒキガエル	19
	真珠湾攻撃	20
	大型間接税	79
	中曽根首相	121
	コマーシャル	18
	ゴルバチョフ氏	76
	フセイン大統領	31
	リクルート事件	35
研究室資料*	テュービンゲン大学	
	ネアンデルタール人	
	ハイデルベルク大学	
	フランス系カナダ人	
	ラテンアメリカ諸国	
	国際通貨基金ＩＭＦ	
	オーストリア継承戦争	
	アデノシン三リン酸ＡＴＰ	
	マサチューセッツ工科大学	

* このテキストの頻度については出していない.



$n_1+n_2=n$

図6　部分文字列への分割

典型的な付属語がつく文字列

研究室資料から「ローマ・カトリック教会」という語が取り出されるが，その後に続く語は「と」，「に」，「の」，「を」，「では」，「との」であった．従って，この種の付属語が後に続く共通の文字列があれば，これらは「ローマ・カトリック教会」と類似した性質の語と考えられよう．このようにして抽出した語の例を表2に示す．

6　mグラム，nグラムの対比

2つの異った長さの文字列mグラムとnグラムの現れ方を対比することによって多くのことが明らかとなる．その典型的な場合を以下に述べる．

複合語についての判断

nグラムの表の中で長さnの語がかなりの頻度で生じた時，図6に示すような長さ$n_1, n_2, (n_1+n_2=n)$の文字列がそれぞれn_1グラム表，n_2グラム表に頻度高く現れ，かつn_1の右端の文字とn_2の左端の文字を含む長さ$n_3(\geq 2)$の文字列がn_3グラム表にあまり現れない場合には長さnの語は長さn_1とn_2の2つの語からなる複合語であるとみなせる．例えば情報科学辞典では表3に示すような例が多く得られている．

特に長い文字列についての判断

特に長い文字列で意味のあるものがある．これは当然出現頻度はあまり大きくなく，上に述べた方法では見過ごされてしまう可能性が高い．これらは外来語の片仮名表記のものが多いが，「AのB」のようにある種の句をなすものもある．これらについては当然のこととはいえ，次のような面白いことが分かった．

例えば，「インプリケーショングラフ」という$n=12$の長さの文字列は4回岩波情報科学辞典に出て来るが，11グラムにおいては「インプリケーショングラ」と「ンプリケーショングラフ」がそれぞれ4回という同じ回数で出て来る．また12グラムにおいて，「カリフォルニア大学バーク」，「リフォルニア大学バークレー」，「ォルニア大学バークレー校」が全て16回ずつ出て来ている．我々の現在の実験では12グラムまでしか印字出力しなかったが，ここから「カリフォルニア大学バークレー校」という15文字列が15グラムの表に頻度16で出て来ることは確実であると言える．

例えば，12グラムの中には「言語で書かれたプログラム」(12回)，「人工知能における問題解決」(12回)，「であるための必要条件」(18回)「ページ置換えアルゴリズム」(9回)，「拡散方程式の有限要素解法」(8回)，「プログラムの部分的正当性」(7回)などがあり，11グラムにおいてこれらのほとんど全ての文字列の前後1文字を削った文字列が同じ頻度で現れている．

このようなことから，図7に示すようにかなり長い文字列の成句(長さn)については，たとえ頻度が少なくても，長さ$n-1$，あるいは長さ$n-2$までの文字列もその成句からのみのものとなってしまい，これらの出現頻度は全て同じとなる．長さがあまり長くない($n=4,5$あたり)文字列においては，前または後1文字を削ったn-1グラムの出現頻度がn文字での出現頻度よりわずかに多くなる．例えば「アドレス」(693回)に対して「アドレ」(695回)，「ドレス」(693回)などである．従って，この現象をとらえてプログラムすることによって専門用語を抽出することが可能となる．

長い一般用語の出現頻度

専門用語でなく，一般の語のならび，特に長い付属語列については次のような考察が可能であろう．まず図8に例をあげよう．これを概念的に書いたものが図9である．いま1つの慣用的によく使われる文字列xがあったとして，それより短い長さlの文字列を考える．その場合，これが文字列xの中にある時にはその出現頻度が文字列xの出現頻度より高く，その両端からはずれて行くとその出現頻度が急速に少なくなって行くことになる．即ち図9に示すような出現頻度の曲線が得られる．逆にこのような曲線が得られる文字列の範囲がよく現れる慣用的文字列であるということになる．このような考え方で調べてみると，

しなければならない，ことが知られている，行なうことができる，求めることができる

といった表現が岩波情報科学辞典でよく現れる慣用的文字列ということになった．これらはいくつもの単語か

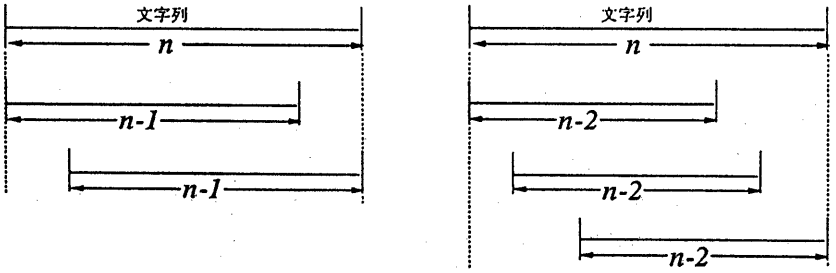


図7　n 文字列と $n-1$ 又は $n-2$ 文字列の関係

らなる複合文字列であり，1つの単語と見ることはもちろんできないが，しかし英語で考えるとこれらは can, be known, can(do), can(ask) といった語に対応し，それぞれ1つのまとまった概念であると考えられなくもない．

7　処理時間に関する検討

このnグラム統計の計算に必要な時間についての検討をここで簡単に行なっておく．この処理で最も時間がかかるのはテキストデータのソートであり，これはデータの大きさを l としたとき $O(l \log l)$ である．その他の処理，マージやnグラムデータの抽出等は全てデータ量 l に比例する．従って，l が大きくなればなるほど全体の時間はソートの時間に大きく依存することになるが，これが $l \log l$ のオーダですむからそれほど深刻な問題ではない．我々が行なった200万文字，3000万文字のテキストデータの処理には Sun SPARC Station 10 を用いて，それぞれ約1時間，1昼夜を必要としたが，これが長時間すぎるかどうかは個人の主観によるだろう．10億文字くらいのデータを処理するためには今日の最高速度のワークステーションで200日位を必要とすることになるだろうが，並列処理計算機の利用も間近であるから夢ではない．この程度の言語データを処理すれば言語的に面白いことが色々と分かって来ると思われる．

8　おわりに

この研究は大規模なテキストデータに対してnグラム統計を(しかも任意のnについて)比較的容易に今日の計算機で作ることができること，これを日本語テキストに対して実行した時，統計的立場からの単語というものをある程度決定できるということを示すことにあった．そして180万文字，400万文字，3000万文字の3種の日本語テキストに対して $n = 12$ グラムまでの統計を出し，種々の言語的な性質を具体的に検討した．

l	文字列	頻度
	することができる	188
8	現することができ	22
	することができる	188
	ることができるが	12
7	現することがで	22
	することができ	223
	ることができる	452
	ことができるが	18
6	用することが	41
	することがで	223
	ることができ	528
	ことができる	665
	とができるが	18
5	定することと	101
	することが	494
	ることがで	582
	ことができ	784
	とができる	665
	ができるよ	25
4	定することこ	101
	することと	1689
	ることが	1310
	ことがで	784
	とができ	784
	ができる	770
	できるよ	147

図8　隣接する部分文字列の頻度

表3　複合語の部分文字列の出現頻度

複合語		妥当な分割	その他の部分文字列
記憶装置 (280)	=	記憶 (1545)・装置 (1540)	記憶装 (280), 憶装置 (280), 憶装 (280)
情報処理 (166)	=	情報 (2058)・処理 (2698)	情報処 (166), 報処理 (166), 報処 (166)
集積回路 (188)	=	集積 (242)・回路 (1350)	集積回 (188), 積回路 (188), 積回 (188)
計算機システム (168)	=	計算機 (1797)・システム (3209)	算機システム (168), 計算機システ (168), 機シ (173)

( )内は岩波情報科学辞典での出現頻度

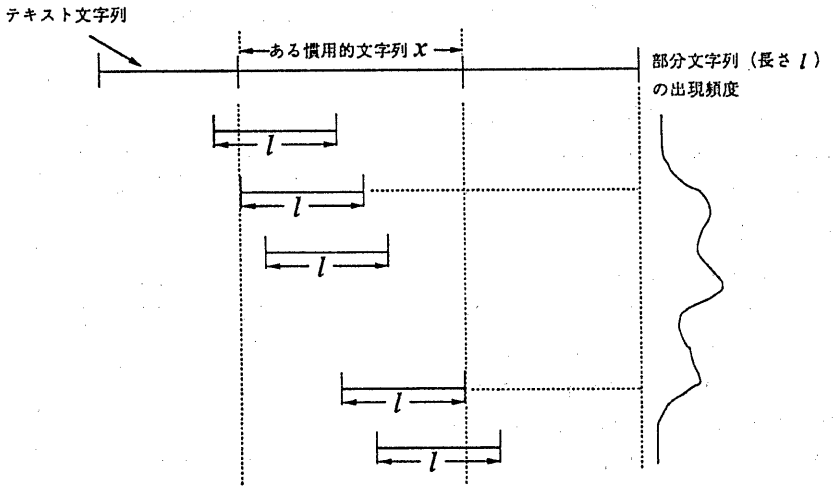


図9　部分文字列の出現頻度曲線

本論文中にそれら全てを記載することはできないし，またデータ量も十分とはいえないので，格別に言語学的に面白く画期的なことがみつかったというところまでは行かなかった．しかし，この程度のデータ量でも色々と考えさせられる事が存在し，そのいくつかについては本文中に述べた．このような立場からのテキスト解析は特にこれからの新しい日本語辞書作成にとって役立つ方法であると考えている．

参考文献

(1) C.E.Shannon: A mathematical theory of communication, Bell System Tech.J., Vol.27, pp.379-423, pp.623-656, (1948).

(2) D.R.Raymond, F.W.Tompa: Hypertext and the Oxford English Dictionary, Comm. ACM, 31(7), pp.871-879, July (1988).