人　工　知　能　88－3
（1993．6．23）

自然言語からのパターン学習による知識獲得

大須賀　勝美、黒川　一夫

東京理科大学

本研究では、自然言語で記述された文書内で繰り返し用いられる文字列のパータンを調べることにより、あらかじめ用意された辞書や文法などの知識を用いることなく、そのパターン学習から文書に関する知識を獲得する方法を考察する。獲得した文字列情報の記憶方法として、階層的な木構造による表現方法を用いる。木構造の同じパターンをまとめて枝の数を最小にすることにより学習を行い、文字の並びによって記述されている文書情報を文書ごとに整理を行う。

シミュレーションにより学習のアルゴリズムに対して比較を行い、人間の判断に近く自然な形で文字列の解釈を行う方法を検討した。また、木構造を用いることにより、効率よく記憶をすると共に、効果的に情報を利用できるようになる。文書から辞書的な情報や文法的な規則を獲得する、学習モデルを報告する。

Knowledge Acquisition by Pattern Learning from Natural Language

Katsumi OSUGA, Kazuo KUROKAWA

Science University of Tokyo

1-3, Kagurazaka, Shinjuku-ku, Tokyo 162, Japan

This paper discusses a method of knowledge acquisition from documents written by natural language. A pattern learning is used in this method and arrangements of characters or words used in documents repeatedly are investigated. The information of their arrangements is expressed by hierarchical tree structure. The learning is performed by combining the same pattern and decreasing the number of branches in this tree.

Some algorithms are compared in their performance by computer simulation. It is possible to obtain information from documents without dictionary data or grammatical rules which are prepared in advance. In using this tree structure, relations of characters or words are memorized efficiently.

1．はじめに

現在一般的に行われている言語処理は、語彙情報として辞書データを持ったり、文法規則をプログラムとしてあらかじめ記述しておき、これらの情報を利用することによって処理を行っている。しかし、事前に登録されていない未知語や文法に従わない文章の場合には処理できないことになる。幅広く多くの分野での文書の処理を行おうとすれば大きな辞書データと文法規則を事前に登録しておく必要がある。そのために、学習によって新しい情報を獲得して追加する能力が必要となり、どの様にして辞書や規則を作り、どの様にして登録しておくかが問題となる。

本研究では、自然語による文書情報から、辞書情報の自動作成と文法規則の自動獲得を行うことを目的とする。そのために、文章の中で使われた文字の並び方のパターンを利用して学習を行うことにより言語に対する知識を獲得するモデルを考える。文字の使われ形を学習する方法と、獲得した知識を整理して格納する記憶方式について検討したのでここに報告する。

2．木構造による文書情報の表現

2.1 日本語文書の特徴

まず最初に、我々の普段用いている自然語である日本語文書の特徴について簡単に考察しておく。文書は文字の並びによって表現されるが、文章が意味を持って情報を伝えるためには、言語としてある規則に従った文字の並び方が必要である。局所的な観点からすれば、単語や熟語といったレベルでの文字の並び方が制約され、大きな観点からすれば文法的に文構造がどうなっているかが問題となる。そして、日本語文書の大きな特徴は表意文字ある漢字を用いていることである。また、漢字は表意文字であるから、文字そのものが意味を持ち制限を与えるので、その並び方には極めて特徴がある。文章の意味を表現するのは大部分が漢字の部分であり、漢字の文字情報を利用することにより文章の意味を獲得することも考えられる。

もう1つの特徴としては、日本語の文書はべた書きでわかち書きされておらず、切り分けを行う必要がある。しかし日本語の場合、漢字の他に平仮名や片仮名、更に数字や記号やアルファベットのようなものも用いられる。文字の種類によってその役割は異なり、文章を切り分ける際にこの文字種類のパターン情報が役に立つ。

次に、漢字文字列の特徴をみてみると、実際の文書中で用いられている漢字文字列について調査を行った結果、平均は約2文字となっている。大部分の漢字熟語は2文字で構成され、3文字以上のものについては単語や熟語に分解することが可能となっている。文字が並んで単語や熟語となり、それらが更に組み合わされて文章となるので、その接続情報を図1のような木構造で表現することができる。

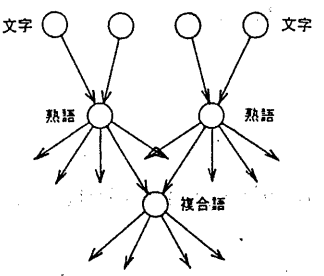


図1　文字列の木による表現

2.2 木構造による文書情報の表現

本研究では文書の文字列情報を図2に示すような木構造によって表現することを考える。この木は文字や文字列の間の結びつきを表しており、文字や文字列がどのように並んで新しい文字列を構成するかを表し、枝を先に向かってたどることによって文字の並び方を知ることができる。ノードは階層的な構造になっており、階層が深くなるほど長い文字列を表現できるようになる。ここで、木の枝の分かれる節点をノードと呼び、そのノードとノードの間を結ぶ木の枝をリンクと呼ぶことにする。

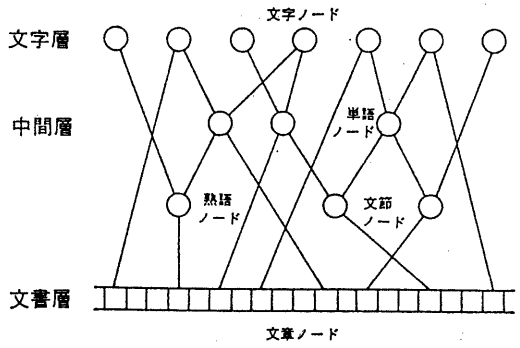
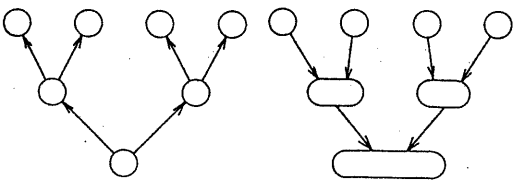


図２　文書情報の木による表現

最上層である第１層目のノードでは文字情報を表し、これを文字層と呼ぶ。第２層目以降の中間層は単語や熟語や複合語などの文字列情報を取り扱う。最下層では文書層として、句点までをひと区切りとした文章単位でノードを作成する。

ノードを一番上の文字層までたどることにより文字列情報となり、これは記憶内容を取り出す際に利用される。また、下層へ向かうリンクはそのノードがどこで使用されているか表現し、文字列や文章の検索の際に利用することができる。図３参照。リンクの持つ情報はこのように双方向性であり、ノードの使用回数をノード間を結ぶリンクの接合強度として表現する。

このように、日本語の文章を階層的な木構造で表現することにより、効率的にデータを蓄積するばかりでなく、効果的に関連性のあるデータを検索することが可能となる。



(a)　読み出し　　(b) 検索

図３　文字列情報の取り扱い

３．パターン学習による階層的な木構造の構築

3.1 パターンによる学習方法

本研究で取り扱う学習の基本的な考え方は、繰り返し用いられる文字の並びを単語や熟語として切り出して認識することである。繰り返し回数の多い文字列パターンから単語と見なしてノード化していくことにする。このとき文字間やパターン間のつながり回数の多い順に組み立てると階層的な木構造になる。文書中で使われる熟語については、専門書などのようにある程度同じ内容を取り扱っている範囲の中では、その内容に関する熟語が多く使われ、同じものが繰り返し用いられる。この繰り返し用いられる文字列の情報を集めることによりその構造を知ることができる。

実際に情報を記憶するのはノード間を結ぶ各リンクであり、同じ文字列を１つにまとめてノードとすることにより、階層を構築していく。また、このように文字列を組み合わせてノードを作成することによりリンクの数が減少し、これを最小化することにより最適化を図る。

3.1 階層木の構築方法

文書情報を表現する階層木構造の構築方法について、２つの場合が考えられる。１つは既に辞書となる文字列の木情報がある場合。もう１つは全く何も情報がないところから始めて、繰り返し用いられる文字列を調べて、新しく階層木を作成していく場合が考えられる。

本研究では辞書情報を一切用いず、与えられた文書から得られた情報に対して最適な階層構造を構築することを目指す。そして、各文書から得られた情報を併合して蓄積することにより、辞書を作成することができる。

ここでは、同じ文書を何度も繰り返し読むことによって、一層ずつ階層を重ねて、リンクの数を最小にするように木を構築していく方法について検討する。何も文字列に関する情報のないところから始めるため、一つの層を作るのに何回か処理を繰り返すことによって不要なノードをなくしてリンク数を減らしていく。

初期状態としては文書情報はノードがなくすべて文字層から直接結ばれ、リンクによって単純に文字の並びによって表現される。したがって、この状態では中間層は存在せず、図４の一番上の状態に相当する。同じパターンをまとめることによりリンク数は減少していく。

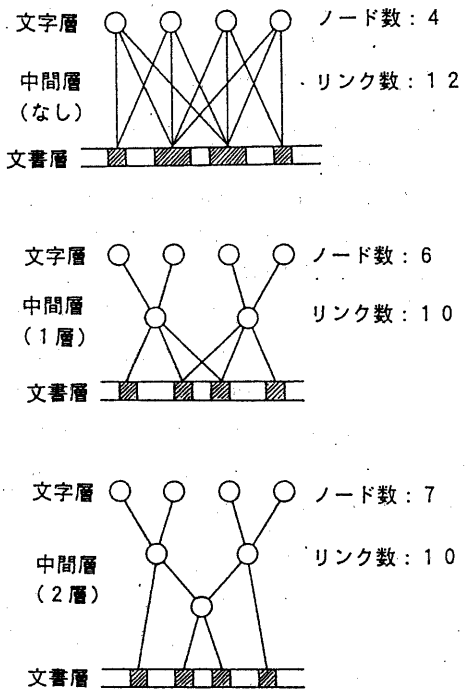


図４　ノード作成によるリンク数の変化

４．学習のアルゴリズム

4.1 二種類の学習方法

木構造を構築する方法としては、全パターンを見て一番使用回数の多い文字列からノードとしてまとめる。次に新しいパターンを含んで再び全パターンを見て一番使用回数の多いものを選ぶ。以下この動作を使用回数が１のものだけになるまで繰り返せば最小になる。これを一括法と呼ぶことにする。

また、図５のように前後の文字や文字列とのつながり方を調べて、接合の度合いが大きい方と結び付けてノードを生成していく。そのときの判断としては図６のように接合強度の変化が上に凸の場合に結び付け、下に凸の場合には結び付けないようにする。すなわち、前後のどちらのノードと結びついた方が良いかを判定することになる。こちらを逐次法と呼ぶ。

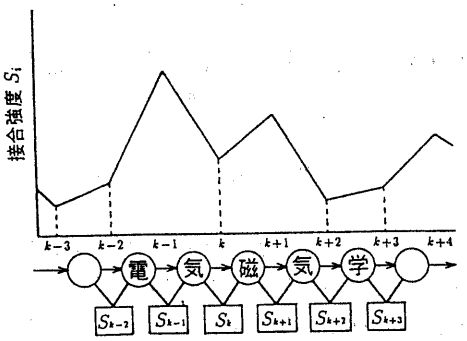
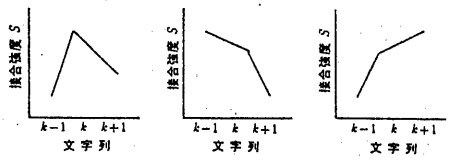


図５　逐次法におけるノード化



(a) 接続の場合

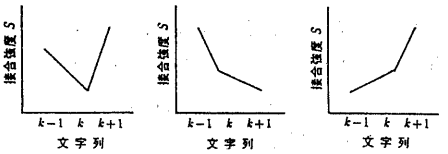


図６　ノードの切り分けの判定

4.2 一括法における学習アルゴリズム

ステップ１：文書を一度全部読んで二文字間の並び方のパターンをすべて調べて、その種類とそれぞれの使用回数を数える。

ステップ２：　使用回数の多いパターンから順番に、一括的に新しいノードとして決定していく。

ステップ３：ステップ１からステップ２までの作業を更に繰り返して層を重ねて、階層構造にする。

このアルゴリズムにおいて、１つのノードを決定するたびにすべてのパターンを調べ直しているのでは大変であるので、ステップ２の作業を何回か繰り返してから再び使用回数を調べ直すようにする。また、ステップ２において既にノードとなった部分を含むノードが候補として現れた場合。そのノードの部分が重複して他のノードに属さないように保留しておき、次の階層においてノードとなるようにして、階層化が行われるようにする。

4.3 逐次法における学習アルゴリズム

ステップ１：文書を一度全部読んで二文字間の並び方のパターンをすべて調べて、その種類とそれぞれの使用回数を数える。

ステップ２：文書を前からもう一度読み直して順番に調べて、各文字間の使用回数から、その部分をつなげて熟語としてノードにするかしないかを判断していく。次の判定式を用いて、前後４文字間でのパターンの使用回数を調べる。

二階差分式

$$(S_i-S_{i-1})-(S_{i+1}-S_i)>0 \quad : \text{接続} \quad (1)$$

$$(S_i-S_{i-1})-(S_{i+1}-S_i)<0 \quad : \text{分割} \quad (2)$$

判定式

$$kS_i-(S_{i+1}+S_{i-1})>0 \quad : \text{接続} \quad (3)$$

$$kS_i-(S_{i+1}+S_{i-1})<0 \quad : \text{分割} \quad (4)$$

ステップ３：ステップ２の判定結果によりステップ１で算出した使用回数に対して、接続なら増加、分割なら減少と修正を加えて学習を行わせる。

ステップ４：文書の最後まで判断が終わったら、再びステップ２に戻り何度か繰り返した後にノードを決定していく。１層分の処理の終了。

ステップ５：ステップ１からステップ４の作業を更に何回か実行して層を積み重ねて、階層構造を構築する。

判定式を発展させて、ここでつなげるか次でつなげるかを比べて判定を行う。５文字間での使用回数を利用する。

二階差分値の比較

$$P_1=(S_i-S_{i-1})-(S_{i+1}-S_i) \quad (5)$$

$$P_2=(S_{i+1}-S_i)-(S_{i+2}-S_{i+1}) \quad (6)$$

判定式

$$P_1=kS_i-(S_{i+1}+S_{i-1}) \quad (7)$$

$$P_2=kS_{i+1}-(S_{i+2}+S_i) \quad (8)$$

$$P_1>0 \text{ かつ } P_1>P_2 \quad : \text{接続} \quad (9)$$

$$P_1<0 \text{ または } P_1>P_2 \quad : \text{分割} \quad (10)$$

５．シミュレーションによる検討

実際の文書を用いて、文字列のパターン学習のシミュレーションを行い比較を行う。この際に用いる文書として、理工系専門文書２種類、理系用語集、人文系専門書、法律文書、の５種類を用意した。実験では特に表意文字である漢字と平仮名によって構成される文字列の部分に着目し、漢字文字列、漢字＋平仮名文字列、平仮名文字列について検討を行った。それ以外の部分は１つの文字列として扱い無視している。また、文章の切り分けの際には、文字の並び方だけに着目しており、文法的な情報は文字種類の区別と句読点による処理だけである。ノード作成の際の判定方法を変えて実験を行い、そのリンク数の減少のしかたを調べて、最適な階層の構築方法を決定する。

5.1 階層化によるリンク数の変化

(1) 一括法

逐次法は、学習によりだんだんと最適な階層構造に近づいていくが、一括法は使用回数の多いものからノードとして採用していくため、繰り返さなくてもリンク数は最小値になるはずである。しかし、この考えでは多いノードだけに着目しており、不要ノードを減らすということを考える必要もある。また、この方法では前から順番に処理していくことはできない。

(2) 逐次法

逐次法について、判定式の係数kを変えた場合について実験を行った。結果として係数kが1.50の場合が1番ノード数が少なくなり、不要ノードも生成されにくいことが分かった。また増加量と減少量についてはその差が大きい程学習速度は速く急激に減少していが、最終的なノード数はそれほど少なくならない。一方、増加量と減少量の差が小さいと、ノード数は緩やかに減少するが、最終的なノード数はより小さくなることが分かった。また、パラメータの違いによる影響は、ほとんどないことがわかるが、増加量＝1、減少量＝1、係数を1.5とした場合が、学習により一番リンク数が少なくなっていることがわかる。階層は3層以降ほとんど変化がなく、辞書情報としては5層分の情報を持っていればよいと考えられる。

(3) 両方法の比較

一括法の長所は同じパターンはすべて同じようにつながるということである。同じパターンでも場所によっては前後関係によっはなれることも必要である。また、判断を間違うとすべてそのパターンは誤りになってしまう。逐次法の場合は前後の関係を見てその都度判断しているので比較的よくあっている。

使用回数の多いものをノードにすることよりも使用回数の小さいものがノードとならないようにする方がよいことがわかる。そのためには繰り返しによる学習によって、使用回数の少ないノードを前後の部分とうまくつなぎ換えて、不要ノードをなくすようにする。

また、パラメータを変えて実際の文書を用いていろいろと実験をして、リンク数の変化と正解率を比較した結果、逐次法にしたときがリンクの数も減少し、切り分け方も人間の判断に近く自然に行うことができた。人間の判断との比較を一致率によって評価すると、漢字部分についてはほとんど間違いなく切り分けられることも確認できた。

5.2 繰り返しによる変化

学習の繰り返しによるリンク数の変化を見てみる。中間層と文書層を合わせた全リンク数を見ると、そのノード数の変化はほとんど差がないが、リンク数と中間ノード数の変化を見てみると増加量＝1、減少量＝1の場合が良いことがわかる。また、係数についても階層を重ねるときとは逆に、繰り返しの時は係数が2.0の時の方が変化量は大きくなっている。ただし、これは第1層目を作成するときのことであって最終的には階層化するので、総合的には階層構造を構築した後の効率が良くなるようにするためには、係数を1.5にした方がよい。

繰り返し回数による学習効果であるが、繰り返しによる全ノード数の変化と使用回数の変化を調べてみると、繰り返していくと使用回数の多いパターンはより強くなり、不要ノードの使用回数が減るので、全ノード数は減少し、その分ノード当たりの使用回数は増えるわけである。結果からも分かるように繰り返し回数は大体5回で収束している。これは文書の種類によらずほぼ一定の傾向があることから、日本語の文書ならば一般的に言えることと思われる。サンプル数は5つと少ないが、内容は異なっているにもかかわらず、大きく異なるデータがないことからそう考えられる。

5.3 文書の違いによる比較

文書の種類や長さに対するノード数の変化であるが、ノード数だけを調べるとその増加はある程度文書が長くなると収束してくるが、文書を再現するためのリンク情報まで含めると約半分程度となる。5つの文章での比較では、全リンク数はどの文書についてもそれほど変わらず、最初の約半分以下にまで減少している。文書の内容によらず日本語の文書ならば同じような階層化の効率が得られると考えられる。また、あまり小さな文書だと学習は困難であるが、ある程度の長さの文書があれば十分であることがわかった。

表1．　一括型　　基本アルゴリズム

階　層	文書ﾉｰﾄﾞ数	中間ﾉｰﾄﾞ数	リンク数	全リンク数	縮小率［%］
0	25898	—	—	25898	100.0
1	14579	1439	3013	17592	67.9
2	9717	1258	2630	15360	59.3
3	8180	548	1161	14984	57.9
4	7972	89	187	14963	57.8
5	7960	6	12	14963	57.8

表2．　一括型　　修正アルゴリズム

階　層	文書ﾉｰﾄﾞ数	中間ﾉｰﾄﾞ数	リンク数	全リンク数	縮小率［%］
0	25898	—	—	25898	100.0
1	17906	784	1620	19526	75.4
2	13140	880	1846	16606	64.1
3	10297	682	1474	15237	58.8
4	8833	398	900	14673	56.7
5	8257	172	400	14497	56.0

表3．　逐次型　　係数 = 1.5　　増加量 = 1　　減少量 = 0

階　層	文書ﾉｰﾄﾞ数	中間ﾉｰﾄﾞ数	リンク数	全リンク数	縮小率［%］
0	25898	—	—	25898	100.0
1	16380	818	1677	18057	69.7
2	11677	868	1773	15127	58.4
3	9469	614	1241	14160	54.7
4	8714	275	551	13956	53.9
5	8511	82	165	13918	53.7

表4．　逐次型　　係数 = 1.5　　増加量 = 1　　減少量 = 1

階　層	文書ﾉｰﾄﾞ数	中間ﾉｰﾄﾞ数	リンク数	全リンク数	縮小率［%］
0	25898	—	—	25898	100.0
1	17001	588	1197	18198	70.3
2	11967	763	1562	14726	56.9
3	9438	695	1397	13594	52.5
4	8478	359	721	13355	51.6
5	8151	143	286	13314	51.4

表5．　文書の違いによるリンク数の変化の比較

階　層	理系文書1	理系文書2	理系用語集	人文系文書	法律文書
0	100.0	100.0	100.0	100.0	100.0
1	69.3	69.1	69.3	67.8	66.3
2	53.8	54.6	54.0	53.5	49.2
3	47.9	49.5	48.1	48.8	40.0
4	46.6	48.5	46.5	48.1	39.4
5	46.4	48.5	46.2	48.1	39.3

6．学習方法の検討

最終的な結果としては逐次型の方が繰り返しによる学習効果により効率をよくできるために、一括型よりも効果が良くなるし、実際に処理する場合にも逐次型の方が望ましい。判定式は５文字間の結合度を調べて、前後のどちらでつなげた方がよいかを比較して結合を判断する。また、この時の評価式の係数は1.5とし、繰り返しによる学習の際の変化は増加量＝１、減少量＝１で行う。

本報告では日本語文章の情報を格納する階層的な木構造による表現方法を述べ、それによるパターン学習による知識の獲得方法を提案した。本方式を用いることにより、あらかじめ辞書を用意して置かなくても、かなりの正解率で文章の切り分けが行えることが確認された。特に専門的な分野の文書においては、専門語が繰り返して多く使用されるのでかなり効果的である。

また、文章をこのようなデータ構造で記憶しておくことにより、事前に文書を読んでその中の文字情報が整理されていることになるので、検索や法則性の獲得などのさまざまな応用に関して効果的である。本システムの記憶装置では文字情報だけを管理しているが、更に各ノード間に意味的な知識を表すリンクを結び付けることにより、意味情報の処理へと展開も可能であると考えられる。

前後に共通のノードがつながる置換可能なノードを探し出ことにより、接頭語、接尾語の識別や送り仮名の付け方など、木構造の解析などの応用方法が考えられるが、この点については現在検討中である。

7．むすび

今回は、ソフトウェアによるシミュレーションによって学習モデルのアルゴリズムの検証を行ったが、各ノードの持つ情報や機能は単純なので、これを小さなプロセッサによって実現して、多数組み合わせて並列的に動作させることによって、ハードウェア化することによって更に効率化を図ることも考えられる。

将来的にはコンピュータシステムに組み込み、言語処理用の機能を装備した専用機の開発へと結び付けていきたい。

参考文献

[1] 高橋延匡：“日本語情報処理”，
近代科学社（1986.7).

[2] 大須賀勝美，黒川一夫：
“日本語を基礎とした計算機システム”，
信学技報，CPSY 90-12～37，Vol.90，No.143，
pp.35-40（1990.7).

[3] 大須賀勝美，黒川一夫：
“日本語情報処理用の計算機システム”，
情報処理学会 第42回全国大会講演論文集(6)，
pp.106-109（1991.3).

[4] 福島俊一：“形態素抽出ハードウェアアルゴリズムとその実現”，情報処理論文誌，
Vol.32，No.10，pp.1259-1267（1991.10).

[5] 高橋直人，板橋秀一：“ニューラルネットワークを用いた日本語解析の試み”，
情報処理学会論文誌，Vol.32，No.10，
pp.1330-1337（1991.10).

[6] 高橋直人，板橋秀一：“相互結合型ニューラルネットによる日本語解析”，
システム／制御／情報，Vol.36，No.7，
pp.441-446（1992.7).

[7] 大須賀勝美，黒川一夫 ほか：
“日本語情報処理用の計算機システム”，
情報処理学会　第44回全国大会講演論文集(6)，
pp.75-78（1992.3).