形態素構文解析

内元 清貴　　馬 青

郵政省 通信総合研究所

〒 651-2401 神戸市西区岩岡町岩岡 588-2

tel:078-969-2186　fax:078-969-2189

{uchimoto,qma}@crl.go.jp

要 旨

言語処理研究においては自然言語の解析としておおよそ「形態素解析」「構文解析」「意味解析」「文脈解析」の四つの段階が考えられている。本稿ではこのうち前の二つの「形態素解析」「構文解析」について、代表的な解析のアプローチと現在の動向を中心に解説する。形態素解析とは、与えられた文を形態素と呼ばれる語の並びに分解し、それぞれの語がもつ品詞、活用などの文法的属性を決定する処理のことである。構文解析とは、文法規則および種々の優先規則に基づいて、語と語の係り受け関係などの文の構造を明らかにする処理のことである。現在、これらの処理は機械翻訳、情報検索、自然言語インターフェース (ワープロの仮名漢字変換など) などに幅広く用いられている。

1　はじめに

言語処理研究においては自然言語の解析としておおよそ以下のような四つの段階が考えられている。

形態素解析 ⇒ 構文解析 ⇒ 意味解析 ⇒ 文脈解析

本稿ではこのうち前の二つの「形態素解析」「構文解析」について説明する。残る「意味解析」「文脈解析」については、本稿の一つ後の『意味文脈解析』(1) で解説している。

2　形態素解析

形態素解析の形態素とは、単語や接辞など、文法上、最小の単位となる要素のことである。形態素解析とは、与えられた文を図 1の一番左の列のような形態素の並びに分解し、それぞれの形態素に対し文法的属性 (品詞、活用、数、性、時制、人称、格など) を決定する処理のことである。

見出し語	読み	基本形	品詞	活用型	活用形
先生	(せんせい)	先生	普通名詞		
に	(に)	に	格助詞		
なった	(なった)	なる	動詞	子音動詞ラ行	タ形

(日本語形態素解析システム JUMAN(2) による解析結果)

図 1: 形態素解析の例:「先生になった」

英語のように分かち書きする (単語と単語の間に空白を入れる) 習慣がある言語と、日本語のように分かち書きをしない言語とでは形態素解析において問題となる箇所が異なる。日本語の形態素解析では、語の区切りを認識すること (単語分割) が問題となる。一方、英語の場合、最初から分かち書きされているため単語分割の必要はないが、英語のほとんどの名詞は動詞としても使うことができるというように品詞の曖昧性が非常に多いという問題がある。例えば、有名な例として “Time flies like an arrow.” という文がある。 flies には飛ぶという動詞の他に蝿という名詞もある。 like には動詞と前置詞がある。したがって、英語の形態素解析ではこのような多品詞語に対する品詞の決定 (品詞付け) が問題となる。このように問題となる箇所は異なるが、日本語文における単語分割と英語文における品詞付けの問題は同じアルゴリズムを用いて解決できることが多い。以下、最近の研究のうち代表的な解析アプローチについて説明する。

2.1　ルールベース

ここでは最も基本的な手法の一つであるルールベースによる方法について、日本語の形態素解析を例にあげて説明する。この方法は英語の場合でも同じように使える。ルールベースとは人手で作成した規則を用いる方法のことで、そのような規則としては連接規則と優先規則というものがある。連接規則は、名詞と助詞のように連接可能な二つの属性を指定したもので、連接できない形態素の並びが解析結果として残らないように除くことを目的とする。しかし、連接規則だけだと形態素の並びに複数の可能性が残った場合に、最も適切な形態素の並びを取り出すことはできない。そこで、優先規則を与えることでもっともらしい解だけを選択できるようにする。優先規則としてはコスト最小法と呼ばれる方法がとられることが多い。この方法は、出現しやすい語ほど、また、あり得そうな品詞の連接ほど低い点数 (語や語の連接に付す点数はコストと呼ばれる) を与え、総コストが最小となるものを優先するというものである。

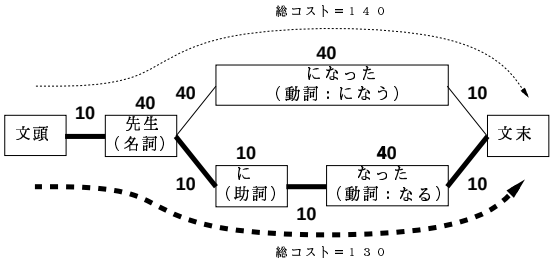
連接規則と優先規則を用いることによって、次のような手順で優先解を得ることができる。入力文として「先生になった」という文が与えられたとする。

1. 語の品詞、読み、活用型などが記載してある単語辞書[1]を参照して、入力文中の各文字位置から始まる語をすべて取り出す。
2. 取り出された語のうち連接規則を満たすものをつないでいく。文頭と文末には特別なノードを設け、それぞれに連接可能な語の属性 (品詞など) についても予め連接規則を用意しておく。
3. それぞれの語、および語と語の連接に語の属性

[1] 単語辞書はトライやパトリシアといった構造を使って、大規模な語彙を高速に検索でき、かつ、非常に小さな空間に格納するように実装される。単語辞書の実装方法については、文献(3) の第 2 章に詳しい説明がある。

や連接可能性を考慮して単語コストおよび連接コストを与える (コストは予め適当に設定しておく)。すると図 2のようなコスト付きのラティス構造が得られる。

4. 文全体で総コストが最小となるような語の並びを優先解とする。図 2の例では、太線のリンクでつながれたノードをたどることによって得られる形態素の並び (下側のパス) がコスト最小解である。



(数字は単語コストと連接コスト)

図 2: 形態素解析の例:「先生になった」

コスト最小解を効率良く求めるために、動的計画法 (Dynamic Programming, DP) という方法が用いられる。これは文頭から文末まで部分的最適解を保持しながら一文字 (英語文の場合一単語) ずつ解析を進めていくという方法で、ビタビ (Viterbi) アルゴリズムとも呼ばれる。このアルゴリズムで文末まで解析して最適解を得た後、今度は文末から A^* アルゴリズムという方法を用いて探索することで、コスト最小解から N 番目にコストが小さい解まで (N-best 解) を順に求めることができる[4], [2]。A^* アルゴリズムは効率良く最適解を求めるために広く用いられている方法である。

コストによって優先解を求めるという方法では、長い語が優先され、分割数が少なくなるように調整される。そのため、形態素解析の精度を上げるためには「形態素解析システム」のように複数の単語からなる複合語や「に関して」などのような定型表現は分割せずにこのままで単語辞書に登録しておく方が良い[6]。また、このような複合語や定型表現などを登録する代わりに、複数の品詞の連接について、決まった品詞の並びには小さなコストを付けるなどして、連接コストの与え方を工夫するような方法もある[7]。

ルールベースの利点は人間にとって理解しやすい規則が作成できることと、規則をコンパクトにできることである。欠点は、規則を変更する際に人間の手間がかかることと、微妙な規則の変更が難しいことである。ある規則を変更することにより、今まで正しく解析していたものを誤って解析するようになるといった副作用が起こりやすい。

現在、日本語の形態素解析では、ルールベースによるシステムが主流である。それに対して、英語の形態素解析では、以下にあげるいくつかのアプローチのようにコーパス (実際の文を集めたもの) から統計的に規則を学習する手法がとられることが多い。人手で作成した規則による方法では、分野・文体が異なるとそれに合わせて規則を追加修正する必要が生じる。その度にコストを人手で与えると、手間、均一性などの問題が生じるが、統計的な手法を用いることによりそれらの問題を解消することができる。

2.2 n-gram(確率正規文法) モデル

n-gram モデルは音声認識の分野で発展してきたモデルで、人が喋る言葉を認識する際、耳にした単語が何であるかはその直前に聞いたいくつかの単語をもとにして予測できると考えるモデルのことである。このモデルではある単語が出現する確率 (生起確率) はその語を含めて先行する n 個の単語によって決まると考える。このように n-gram の n は先行する n 個を考えるということを意味する。このような n-gram モデルを用いて形態素解析を行う場合には、ある単語の品詞はその単語と先行する n 個の品詞をもとにして予測できると考える。これはルールベースによる方法において人間が与えていた単語コスト、連接コストの代わりにコーパスから計算した単語の生起確率、品詞の連接する確率を使うことに相当する。そして、コスト最小法の代わりに、単語の生起確

[2] アルゴリズムについては、文献[5] の第 3章を読んでその章の最後の演習問題を解くと理解しやすい。

率、品詞の連接する確率を掛け合わせたものが一文全体で最大となるものを解として優先するという優先規則を使うことに相当する。

2.2.1 隠れマルコフモデル

ここでは、2-gram の例としてよく使われるモデルの一つである隠れマルコフモデル (Hidden Markov Model, HMM) を紹介する[3]。日本語の形態素解析では、入力文が与えられると、単語列 $W = w_1 \ldots w_n$ に分割し、各単語に品詞を与えて品詞列 $T = t_1 \ldots t_n$ を出力する。このとき、単語列 W と品詞列 T が同時に成り立つ確率 $P(W,T)$ を最大にするような W と T が求めたい単語列、品詞列であると考える。隠れマルコフモデルでは、この同時確率 $P(W,T)$ を次のように近似する。

$$P(W,T) = \prod_{i=1}^{n} P(t_i|t_{i-1})P(w_i|t_i) \quad (1)$$

ここで、$P(t_i|t_{i-1})$ は、ある品詞が現れる確率は直前の一つの品詞だけに依存すると考えたときの品詞の連接確率 (連接コストに対応) であり、$P(w_i|t_i)$ は各品詞に対してある単語が出力される確率が前後の品詞とは独立であると考えたときの単語の生起確率 (単語コストに対応) である。

式 (1) における単語の生起確率や品詞の連接確率は、コーパスに現れる単語や品詞の連接の相対頻度から以下のような式で計算される。

$$P(t_i|t_{i-1}) = \frac{C(t_{i-1},t_i)}{C(t_{i-1})} \quad (2)$$

$$P(w_i|t_i) = \frac{C(w_i,t_i)}{C(t_i)} \quad (3)$$

ここで、$C(x)$ は x の出現頻度を、$C(x,y)$ は x と y が同時に出現する頻度を表す。このようにしてそれぞれの確率 (パラメータ) を計算する作業はパラメータ推定と呼ばれる。

パラメータ推定には一般にタグ付きコーパスが利用される。タグ付きコーパスとは、単語区切りや品詞の情報 (品詞タグ)、構文情報 (構文タグ) などが付与されたコーパスのことをいう。1995 年あたりまでに利用可能となったタグ付きコーパスについては文献[8] が詳しい。それ以降に利用可能となった日本語のコーパスには、新聞記事を対象とした RWC コーパス[9]、京大テキストコーパス[10] がある。

コーパスには必ずしもタグが付いている必要はない。生のコーパスからでも Forward-Backword アルゴリズム[11] という方法を用いてパラメータを推定することができる。しかし、この場合の解は局所的な最適解でしかなく、解析精度はタグ付きコーパスから推定した場合よりも良くないことが報告されている[11][12]。

式 (1) で表されるモデルでは、ある品詞が現れる確率は直前の一つの品詞だけに依存すると考えたが、直前の $n-1$ 個に依存すると考えることもできる。この場合、実際にはあらゆる品詞の連接がコーパスに現れることは期待できない (これはデータが疎らにしかないという意味でデータスパースネスの問題と言われている)。そこで、品詞の n 個の連接 (品詞 n-gram) に対応するパラメータについては、それより少ない連接に対応するパラメータを用いて近似する。これを平滑化 (smoothing) という[4]。例えば、3-gram の確率値を低次の j-gram$(3 \leq j)$ の確率値を用いて近似的に求める方法をとると、3-gram の確率値 $P_{new}(t_i|t_{i-2},t_{i-1})$ は以下のように表される。この方法は線形補間法と呼ばれる。

$$\begin{aligned} P_{new}(t_i|t_{i-2},t_{i-1}) = {} & \lambda_3(t_{i-2},t_{i-1})P(t_i|t_{i-2},t_{i-1}) \\ & +\lambda_2(t_{i-2},t_{i-1})P(t_i|t_{i-1}) \\ & +\lambda_1(t_{i-2},t_{i-1})P(t_i) \end{aligned} \quad (4)$$

ここで、$\lambda_3(t_{i-2},t_{i-1})$、$\lambda_2(t_{i-2},t_{i-1})$、$\lambda_1(t_{i-2},t_{i-1})$ はそれぞれ 3 次、2 次、1 次の確率値、$P(t_i|t_{i-2},t_{i-1})$、$P(t_i|t_{i-1})$、$P(t_i)$ がどの程度 $P_{new}(t_i|t_{i-2},t_{i-1})$ の近似に使えるかを表す重み係数である。これらの重み係数は例えば EM アルゴリズム[13] という方法を用いて、パラメータを推定したものとは別のデータから推定される。

[3] 隠れマルコフモデルは非決定性有限状態オートマトンとして定義される。単語の系列が与えられても、品詞の遷移の系列は唯一には決まらない。観測できるのは単語の系列だけであることから「隠れ」マルコフモデルと呼ばれる。日本語では 2-gram、英語では 3-gram を扱うことが多い。

[4] 平滑化の基本的な考え方は文献[3] でやさしく解説されている。

2.2.2 n-gram モデルの拡張

前節までは品詞の連接を考えていたが、より正確に品詞付けを行うためには品詞より細かい情報を扱う方が良い。このようにより細かい情報を扱ったモデルとして、単語と品詞の組の n 個の連接を扱う形態素 n-gram と呼ばれるモデルがある。例えば、形態素 2-gram を考えるとき、式 (1) で表される隠れマルコフモデルは次のように変更される。

$$\begin{aligned} P(W,M) &= \prod_{i=1}^{n} P(m_i|m_{i-1})P(w_i|m_i) \\ &= \prod_{i=1}^{n} P(m_i|m_{i-1}) \end{aligned} \quad (5)$$

ここで、m_i は単語 w_i と品詞 t_i の組である。m_i が決まれば必ず w_i も決まるため $P(w_i|m_i) = 1$ となる。しかし、品詞だけでなく単語も区別することで、これらの組のコーパスにおける出現頻度は極端に少なくなり、推定されたパラメータの信頼性が低下するという問題が生じる。そこで、この問題に対処するために、予め単語をいくつかのクラスと呼ばれるグループに分類しておき、クラスと品詞の組の n 個の連接 (クラス n-gram) を考える方法が提案されている[14, 5]。例えば、クラス 2-gram を考えるとき、式 (1) で表される隠れマルコフモデルは次のように変更される。

$$P(W,C) = \prod_{i=1}^{n} P(c_i|c_{i-1})P(w_i|c_i) \quad (6)$$

ここで、c_i は単語 w_i が属するクラスと品詞 t_i の組である。推定しなくてはならないパラメータの数は品詞 n-gram、クラス n-gram、形態素 n-gram の順に多くなる。このようなクラスというものを設けたのは、品詞だけでは分類が粗過ぎて正確に品詞付けできないことが多いためである。同じような問題意識から、品詞を誤り駆動で細分類する試みもなされた[17]。これは推定を誤ることが多い品詞を細分類することによって粗過ぎる分類を細かくしていこうというものである。先に述べた研究ではクラスがボトムアップに作られていたのに対して、この方法ではクラスに相当するものをトップダウンに自動獲得する。

ここまでは、常に n 個の長さに固定された連接を考えてきた。しかし、パラメータを推定するためにどんな単語でも常に n 個前の情報が必要な訳ではない。単語によって参考にするべき文脈 (先行する過去の系列) の長さが異なるはずである。このような考え方から、文脈木というものを用いる方法が提案された[18][19]。これは可変記憶長マルコフモデル (Variable Memory Markov Model, VMM) とも呼ばれる。この方法を用いることで確率の予測に最適な長さの文脈を選択することが可能となり、少数のパラメータで長い範囲の接続を記述できる[18]。さらに春野らは、品詞の推定に誤りが多い部分に着目して、一般的なモデルから少しずつ例外的な現象に特化したモデルまで複数の確率モデルを学習し、それらのモデルをブースティングと呼ばれる方法を用いて混ぜ合わせることで日本語形態素解析の精度を向上させている[19]。この方法では、まず一つモデルを作り、そのモデルを用いてコーパスを解析する。そして解析を誤った部分を取り出してそこに特化したモデルを学習する。さらにそのモデルを用いて前の解析誤りを解析し、それでも解析を誤った部分があればそれを取り出してそこに特化したモデルを学習する。これを繰り返すことにより複数のモデルを学習する。

ここまでは主に、すべての単語が辞書に記載されている、あるいはコーパスに現れることを前提としていた。しかし、そのような前提を仮定するのは現実的ではない。このように辞書にもコーパスにもない単語が存在するという問題は未知語の問題と呼ばれ、その対策としていろいろな方法が考えられてきた。Weischedel らは未知語に含まれる頭文字、ハイフン、語尾の文字列に着目し、それらの文字列を含む単語の品詞が、コーパスではどのように分布しているかを調べて、未知語の品詞の確率分布を求めようとしている[20]。永田は文字の n-gram の連接確率を求め、単語の生起

[5] クロスエントロピーが最小になるようにクラス分類をすることによって形態素解析の精度が上がるという報告もある[15]。クロスエントロピーはモデルがどれだけ正確に自然言語を近似しているかを表す尺度として用いられる。クロスエントロピーの定義については文献[16]を参照。

確率をその単語に含まれる文字の n-gram の確率で近似することで、未知語を扱っている[4]。

2.3 決定木モデル

決定木モデルでは、式 (1) の同時確率 $P(W,T)$ を決定木 (Decision Tree) を用いて求めることによって形態素解析を行う。これはすでに知っている (記憶している) 事例の中から、決定木を用いて入力文と最も似た属性をもつものを取り出し、そこに付与されている品詞タグを答とすることに相当する。ここで属性とは品詞付けに用いる情報のことで、例えば「一つ前の品詞が名詞であるかどうか」などが一つの属性として用いられる。決

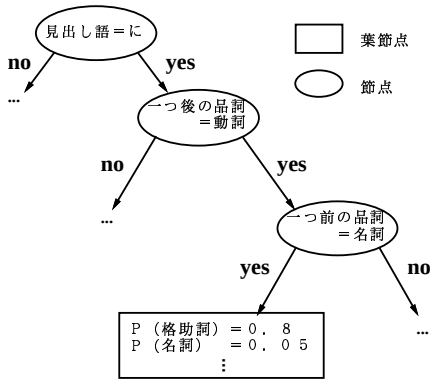


図 3: 決定木の例

定木とは、図 3のように、各節点に属性が設けられており節点から出た二本の枝はその属性のあるなしを表すような二分木のことである。木の根から順番に葉節点に向けて各節点の属性に関する質問に答えていくと、行き着いた葉節点に最終的な答が書いてある。例えば、図 3のような決定木を作ったとする。入力として、「先生／に／なった」という単語列を考え、「先生 (名詞)」と「なった (動詞)」については辞書を引いて一意に決まったと仮定する。このとき、「に」については格助詞や数量詞など品詞の候補が複数考えられるが、決定木をたどり最終的に行き付いた葉節点に書いてある確率を参照することにより品詞付けができる。例えば、図 3で確率値の一番大きなものをとることにすると「に (格助詞)」と品詞付けを行うことになる。

決定木は属性のあるなしによってコーパスから集めた事例を分割していくことによって作られる。その際、どの属性から順に分割していくかが重要となる。分割は次の二つの条件が満たされるようになされるのが望ましい。

1. 同じ品詞タグが付与されている事例ができるだけ同じ事例集合に含まれる。
2. 分割された個々の事例集合に含まれる事例の数が極端に少なくならない。(この条件が満たされないと確率を求めたときに信頼性が低下してしまう。)

これらの条件を満たすために、事例を分割してみて、分割する前と後とで事例集合全体の平均情報量の変化が最も大きくなくような属性から順に分割するという方法がとられる。ここで、平均情報量とは情報の不確かさの度合を表す量のことであり、この量の変化が大きいほど情報の不確かさが減る、つまり、いろいろな品詞タグが混在する事例集合が減ることを表している。また、すべての属性を使って分割すると、分割された個々の事例集合が小さくなってしまうので、分割は適度なところで止める。分割が終ったとき、葉節点はコーパスから集めた事例の集合からなる。葉節点のラベルは、そこに付与されている品詞のうち最も多いもの (例えば「格助詞」など) とすることもできるし、複数の品詞がある場合にはそれぞれの占める割合 (確率) で表す (例えば、図 3) こともできる。ラベルをそれぞれの品詞の確率で表す場合、その確率は式 (1) の確率 $P(t_i|t_{i-1})P(w_i|t_i)$ の代わりとして用いることができ、 n-gram モデルと同じように解析を行うことができる。

決定木モデルが n-gram モデルより優れている点は二つある。一つは、決定木の各節点にはどんな属性でも用いることができる点である。例えば、連接する品詞列だけでなく、離れた単語の品詞、文字列なども考慮することができる。もう一つの優れている点は、属性として何を選んでおいてもよいという点である。極端なことをいえば、思いつく限りの属性をすべて用意しておいてもよい。これは、決定木は重要そうな属性から順に分岐するように作られるためである。したがって、

形態素解析を行うときには決定木を根から順にたどりながら属性に関する質問に答えていくだけでタグ付けするべき品詞を予測するのに必要な文脈の長さが自動的に決まることになる。このとき、必要な文脈の長さ、つまり着目している単語の品詞を予測するのに必要な情報の多さは、木のたどる深さで表される。

Daelemansらは既知語と未知語ごとに考慮するべき文脈が異なるだろうとの考察からそれぞれに別の決定木を用意し、属性として単語の語尾や語頭の文字列、前後の単語などを考慮することによって未知語に対しても高い品詞タグ付けの精度を得ている(21)。

2.4 最大エントロピーモデル

最大エントロピー (Maximum Entropy) モデルでは、式 (1) の同時確率 $P(W,T)$ を最大エントロピー法を用いて求めることによって形態素解析を行う。最大エントロピー法とは、素性[6]が与えられたとき学習データ中に観測された素性の頻度などからそのデータに特徴的な素性を重み付けする仕組みのことである。この方法の一つの特徴は品詞タグの違う事例をうまく区別できるような素性の場合には大きな重みが与えられるという点である。しかし、素性全てが高い頻度でコーパス中に現れるわけではない。最大エントロピー法のもう一つの特徴は、観測されにくい素性についても考慮する点である。エントロピー[7]とは情報の不確かさの度合を表す量のことで、確率分布 $P(W,T)$ のエントロピーを最大にするという手法によって、観測されにくい素性に対しては $P(W,T)$ の確率値がどの品詞に対しても等確率になるように重み付けされる。このため最大エントロピーモデルはデータスパースネスに強いとされている。

このモデルの確率は、以下のような式で表される。

$$p(h,t) = \pi \prod_{j=1}^{k} \alpha_j^{f_j(h,t)} \quad (7)$$

$$= \pi e^{\lambda_1 f_1(h,t)+\ldots+\lambda_j f_j(h,t)+\ldots+\lambda_k f_k(h,t)}$$

$$(0 \leq \alpha_j \leq \infty, \pi\text{は正規化定数})$$

ここで、 $p(h,t)$ は文脈が h でかつ、今、付与される品詞タグが t であるような確率を表す。この確率は、式 (1) の確率 $P(t_i|t_{i-1})P(w_i|t_i)$ の代わりとして用いることができる。式 (7) で、 λ_j は素性 f_j の重みを表す。 $f_j(h,t)$ は素性 f_j が観測されたときに 1、それ以外のときに 0 を返すような関数で、 n-gram モデルでいうと低次の j-gram の補間による補間係数に対応させることができる。 n-gram モデルとの違いは、 h として何を考えてもよいという点である。例えば、単語 w_i に対して、前後二単語ずつと前二つの品詞を文脈 $h_i(=\{w_i, w_{i+1}, w_{i+2}, w_{i-1}, w_{i-2}, t_{i-1}, t_{i-2}\})$ とするとき、 $f_j(h,t)$ の一例として以下のようなものを考えることができる。

$$f_j(h_i,t_i) = \begin{cases} 1 & if\ \text{語尾}(w_i,h_i) = \text{「た」} \\ & \quad \& \ t_i = \text{動詞} \\ 0 & otherwise \end{cases} \quad (8)$$

これは h_i における単語 w_i の語尾が「た」で、かつその単語の品詞が動詞であるときのみ 1 を返す関数を表している。

素性としては決定木モデルと同様に様々な属性を利用することができる。決定木モデルとの違いは、決定木モデルでは重要な属性から順番に調べていくため、必ずしもすべての属性からの影響を考慮する訳ではないのに対し、最大エントロピーモデルではすべての素性の影響を考慮する点にある。現在、単独のモデルでは最大エントロピーモデルを用いた Ratnaparkhi の品詞付けの方法(22)が最も精度が高い。

2.5 ニューラルネットワーク (神経回路網) モデル

ニューラルネットワークモデルでは、品詞付けにニューラルネットワークを用いる。ニューラルネットワークはある特定のパターンの刺激に対してある特定の出力をするように学習させたシステムのことで、その働きが人間の神経回路網の働きと似ていることからそう名付けられた。

ニューラルネットワークを形態素解析に応用すると次のようになる。まず、図 4のような三層か

[6] この方法では属性の代わりに素性という言葉が使われる。

[7] エントロピーと 2.3節の平均情報量は同じものである。

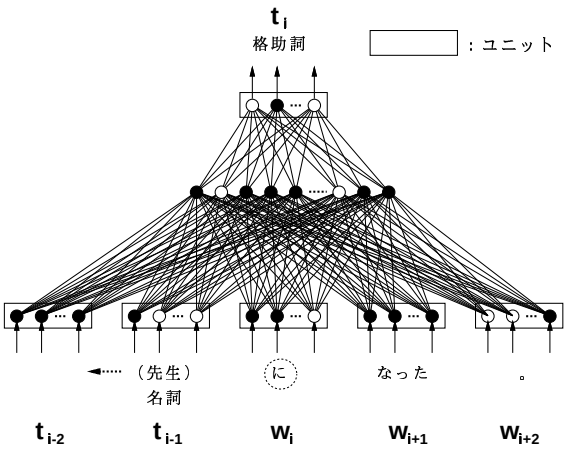


図 4: ニューロタガーの例

らなるニューロンと呼ばれるノード群を用意する。このとき、最下層および最上層のノードは品詞の種類の数だけのノード (品詞が 50 種類ある場合には 50 個のノード) をまとめて一つのユニットとする。この一つのユニットに対してそれぞれ一つの単語あるいは品詞を対応させる。例えば、図 4では最下層に同じユニットを五つ用意しており、単語「に」に対応するユニットには格助詞や数量詞など「に」に付与され得るすべての品詞からなるパターンを与える。中間層のノードは上下の層のすべてのノードとリンクでつながっている。次に、最下層から各単語に対応するパターンを刺激として各ユニットに与え、そのときに最上層のユニットが特定の品詞に対応したパターンで発火するようにリンクの重みを調節する。ここまでが学習の過程に当たる。解析するときには、最下層のユニットごとに各単語に対応するパターンを与える。すると、そのパターンがリンクの重み付けのときに受けた刺激と似ている場合に、最上層のユニットで特定の品詞に対応するパターンが発火する。図 4にあげたニューラルネットワークは、ある語 w_i の品詞 t_i を推定させるために、前二つの品詞 t_{i-2}、 t_{i-1} とその語 w_i および後二つの語 w_{i+1}、 w_{i+2} を刺激として学習したものの例である。図 4では、入力の単語列が「先生／に／なった／。」のときに、真中の単語「に」の品詞を推定している様子が表されている。次の語の品詞を推定するときは、単語列を全体に一つずつ左へずらす。

Schmid は英語の品詞タグ付けに、馬らはタイ語の品詞タグ付けにニューラルネットワークを応用した[23][24]。ニューラルネットワークモデルでは、 n-gram モデルに比べると、推定するべきパラメータの数が格段に少ない。例えば、品詞が 50 種類ある言語の場合、 3-gram モデルでは、$50^3 = 1.25 \times 10^5$ 個のパラメータを推定しなければならない。それに対し、図 4のようなニューロタガーの場合、 5-gram モデルに相当する場合を考えたとしても、中間層のノードの数を最下層のノード数 (50×5) の半分 (125) とすると、リンクの数は $50 \times 5 \times 125 + 125 \times 50 = 37,500$ 個である。パラメータの数が少なければ、それらを正しく同定するのに必要な訓練データの数も少なくてよい。そのために、ニューラルネットワークモデルの品詞タグ付けの性能は n-gram モデルに比べて訓練データの少なさに影響されにくい。また、n を大きくとることも可能になる。馬らはさらに最下層のユニットの数が違ういくつかのニューラルネットワークモデルを混ぜ合わせ、状況に応じて適切な長さの文脈を自動的に選ぶようにすることで、精度を向上させている[25]。

2.6 いくつかのモデルの混合

形態素解析の精度を向上させるために、いくつかのモデルを混合するアプローチの仕方がある。混合の仕方には二通りある。一つはモデルを逐次的に利用する方法で、もう一つはモデルを並列に利用する方法である。

逐次的な方法には Brill の提案した誤り主導の変換に基づく学習によるものがある。 Brill は人手により与えられた正解データとシステムの解析結果を比較し、解析誤りから自動的に書き換え規則を獲得する、誤り主導の変換に基づく学習の方法を提案した[26]。これは、よく同じ間違いをするところを学習するという方法である。

書き換え規則の獲得のために、予め図 5のようなテンプレートを用意しておく。学習の段階では、最初に簡単な統計手法などを用いて品詞付

次の条件を満たす場合タグ **a** をタグ **b** に書き換える。

1. 一つ左 (右) の単語のタグが **z** である。
2. 二つ左 (右) の単語のタグが **z** である。

⋮

図 5: テンプレートの例

けを行い、解析を誤ったもののうち図 5のテンプレートによく当てはまる書き換え規則を獲得する。実際のタグ付けでは、簡単な統計手法などを用いて品詞付けを行った後、学習で得られた書き換え規則を用いて品詞を付け直す。

この方法は、他の解析システムの後処理として使われ効果をあげている。久光らは、ルールベースの形態素解析システムの後処理として利用した[27]。単語境界の書き換え規則[8]も含めることによって、 Brill の方法を日本語文に適用し、解析精度を向上させている。馬らはニューラルネットワークモデルの後処理として利用した[28]。ニューラルネットワークモデルでは扱いにくい単語そのものの情報などを書き換え規則を利用することで反映させている。

並列にモデルを利用する方法には、複数のモデルからそれぞれ得点付きの制約を取り出し同じレベルで扱うもの、複数のモデルから出力された結果を使って多数決を採るものなどがある。Màrquez らは人手で作成した規則、 n-gram モデル、決定木モデルから取り出した得点付きの制約をひとまとめにして、得点の高い制約から適用することによって品詞付けを行う方法を提案した[29]。 Halteren らは、誤り主導の変換に基づく学習モデル、 n-gram モデル、決定木モデル、最大エントロピーモデルを用いた四つのシステムに出力結果を投票させることによって、品詞タグ付けの信頼性を高める方法をとった[30]。いくつかの投票の仕方で投票させた結果、それぞれのシステム単独の結果より良い 98% 程度の精度を得たと報告している。これは上にあげた四つのモデルはそれぞれに性質が異なっており、お互いに短所を補い合うことができたということを示している。

[8] 例えば、単語境界の書き換え規則として以下のようなものが考えられている。以下で「／」は単語の境界を表す。
助詞／ “C_1” : 普通名詞／ “C_2” : 未登録語／助詞
⇒ 助詞／ “C_1C_2” : 普通名詞／助詞

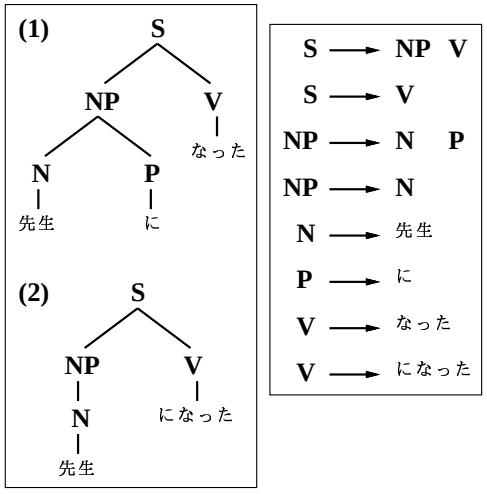
3 構文解析

構文解析とは、文法規則および種々の優先規則に基づいて文の構造を明らかにする処理のことである。構文解析に用いられる文法の枠組には、主に句構造文法、依存文法の二つがある。以下、それぞれの文法に基づく解析方法について説明する。

3.1 句構造文法に基づく解析方法

3.1.1 文脈自由文法に基づく解析

句構造文法の枠組では文脈自由文法 (context free grammar, CFG) を仮定する方法が一般的である。図 6の (b) が文脈自由文法の文法規則の例で、生成規則あるいは書き換え規則と呼ばれる[9]。この規則を用いて、文 (S) から出発して順次書き換え規則を適用することによって例えば図 6の (a) のような構文木を作り出すことができる。このように各々の規則を適用していくことを導出と呼ぶ。

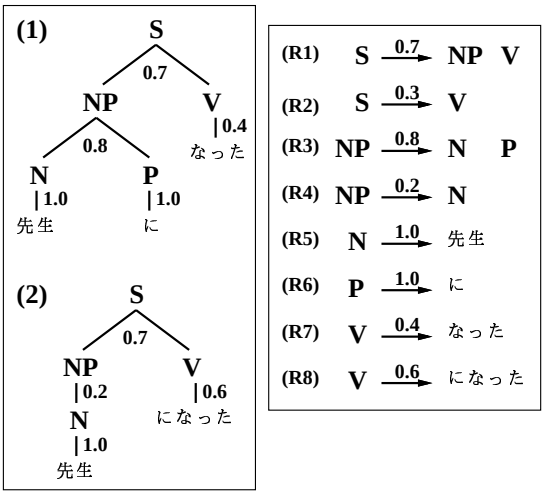


（ａ）構文木　　（ｂ）生成規則

図 6: 文脈自由文法と構文解析の例

導出の仕方にはいくつかあり、図 6の (1) のような構文木だけでなく、同じ規則から図 6の (2) のような構文木も導出できる。どちらの構文木を

[9] 基本的なことについては文献[5] の第 2 章、第 4 章を参照のこと。



（a）構文木　　（b）生成規則

図 7: 確率文脈自由文法と構文解析の例

優先するべきかを決めるために、図 7の (b) のように各規則に確率を与える方法がある。確率は、コーパスから各規則が使われる頻度を調べることによって計算する。確率値は、規則の左辺が同じものをすべて足すと 1 になるように定める。このような確率付きの文脈自由文法を確率文脈自由文法と呼ぶ。今、開始記号を S_0 とするとき、文 S が S_0 から始めて、m 個の規則を用いることで導出されるとする。

$$S_0 \stackrel{r_1}{\Rightarrow} t_1 \stackrel{r_2}{\Rightarrow} t_2 \stackrel{r_3}{\Rightarrow} \ldots \stackrel{r_i}{\Rightarrow} t_i \stackrel{r_{i+1}}{\Rightarrow} \ldots \stackrel{r_m}{\Rightarrow} t_m = S \quad (9)$$

ここで、t_i は i 番目の導出により導かれた木構造を表す。例えば、図 7の (1) の構文木は、

$$S_0 \stackrel{R1}{\Rightarrow} NP\ \ V \stackrel{R3}{\Rightarrow} N\ \ P\ \ V \stackrel{R5}{\Rightarrow} \text{先生}\ P\ \ V$$
$$\stackrel{R6}{\Rightarrow} \text{先生 に}\ V \stackrel{R7}{\Rightarrow} \text{先生 に なった}$$

のように導出される。ここで矢印 $(\Rightarrow)$ の上に書いてある記号は各導出で用いた生成規則に対応する。例えば二つ目の導出では、NP に対して規則 (R3) が適用され N と P が生成されている。

確率文脈自由文法では、各々の導出が独立であると考え、文 S が構文木 T として生成される確率 $P(T,S)$ を

$$P(T,S) = \prod_{i=1}^{m} P(r_i|t_i^-) \quad (10)$$
$$= \prod_{i=1}^{m} P(r_i) \quad (11)$$

のように表す。ここで、t_i^- を i 番目までの導出により導かれた木構造とし、i 番目の導出は t_i^- には依存しないとして、$P(r_i|t_i^-) = P(r_i)$ のように近似している。式 (11) を用いると、例えば、図 7の (1) の確率は、$0.7\times0.8\times1.0\times1.0\times0.4 = 0.224$、(2) の確率は、$0.7 \times 0.2 \times 1.0 \times 0.6 = 0.084$ のようになる。したがって、図 7の文法で「先生になった」を解析する場合には (1) の構文木が優先解となる。解析の方法としては、アーリー法などのトップダウン型のアルゴリズム、CYK 法、LR 法などのボトムアップ型のアルゴリズムなどがある[10]。

文脈自由文法の文脈自由とは、木を導出するそれぞれの段階で、以前に適用した規則とは無関係 (自由) に次の規則を適用できることを意味する。しかし、例えば動詞がどのような単語であるかによって主語や目的語になり得る単語が制限されたり、前の文や節で使われていた単語が省略あるいは代名詞化されることが多いというように、実際の文には単語そのものの振舞いやもっと広い範囲の文脈が反映されているわけであるから、各々の規則は独立に適用されるべきではない。このような観点から、共起する単語の統計的な情報などを用いて語の振舞いを考慮できるようにしたり、もっと広い範囲の情報を考慮できるように工夫することによって文脈自由文法を拡張しようという試みがなされてきた。以下の節では、それらの拡張方法について説明する。

3.1.2 語彙情報の利用による拡張

文脈自由文法に基づく解析では基本的に品詞間の関係しか考慮されていないが、近年、統計的な情報として語彙 (単語、形態素) が使えるように様々な工夫がなされるようになった。これまでに、単語の出現頻度や単語の共起関係といった語彙的な統計情報をうまく利用して解析精度を向上させた研究が数多く報告されている[31][32][33]。

日本語については以下のような研究がある。2.2.2節でも説明したように、形態素解析に用いる情報としては、品詞では分類が粗過ぎ、語彙では細か過ぎた。そこで、その中間くらいの細かさの分類としてクラスというものが設けられた。

[10] それぞれのアルゴリズムは、文献[5] 第 4 章でやさしく解説されている。

これにより、形態素解析の精度を向上させることができる。構文解析についても同様にクラスを解析に利用した研究がある。森らは、クロスエントロピーが最小になるように形態素をクラスタリングすることにより、係り受けモデルの精度を上げることに成功した[34]。係り受けモデルは確率文脈自由文法で表現している。係り受けについては3.2節で述べる。

白井らは、統計情報の種類によって学習に要する言語資源の質・量が大きく異なることに着目し、構文的情報と語彙的情報をそれぞれ独立に学習して組み合わせる手法を提案した[35]。そこでは、動詞と助詞が共起する確率を調べて利用することにより、それぞれの動詞がどのような格をとるのかといった語の振舞いを考慮できるようにしている。

3.1.3 文脈依存への拡張

我々人間が作る文の構造はそれまでに生成された文や節 (文脈) によってある程度制限されている。ところが、文脈自由文法では、それぞれの導出は過去にどのような導出がなされたかには依存しないと仮定しているため、それまでの文脈を反映させるのが難しい。そこで、文脈を反映させるためにいくつかの方法が提案された。ここでは代表的な方法である履歴に基づく文法による方法と複数の規則を統合した方法について説明する。

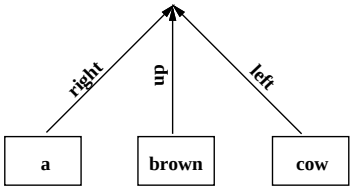
[履歴に基づく文法]

Black らはある規則を適用する際にそれまでの導出の履歴を文脈情報として用いる履歴に基づく文法 (History Based Grammar) という考え方を提案した[36]。これは、前の文や節で使われていた語が何であるか、前の文が質問文であったかどうかなどによって、次の文の導出が制限されるだろうとの考え方に基づく。この文法では、各生成規則の確率をその直前に適用されたいくつかの生成規則の条件付き確率とする。導出される順番が重要だと考えるのである。したがって、文 S が構文木 T として生成される確率 $P(T,S)$ は先にあげた式 (10) のように表される。この式における確率 $P(r_i|t_i^-)$ は 2.3節で説明した決定木モデルを用いて求められる。ここで、もし t_i^- (i 番目までの導出により導かれた木構造) を得たときと全く同じ導出をコーパスに求めるとデータスーパスネスの問題が生じるが、決定木モデルでは、t_i^- と同じ属性をもつ木構造はすべて同じクラス $E[t_i^-]$ に属するものとして扱えるためあまり深刻な問題とはならない。このとき決定木モデルにおいて、確率 $P(T,S)$ は次のような式で表される。

$$P(T,S) = \prod_{i=1}^{m} P(r_i|E[t_i^-]) \qquad (12)$$

導出は、各導出過程で得られる木構造において常に一番左にある節点に規則を適用する最左導出のみを考えている。

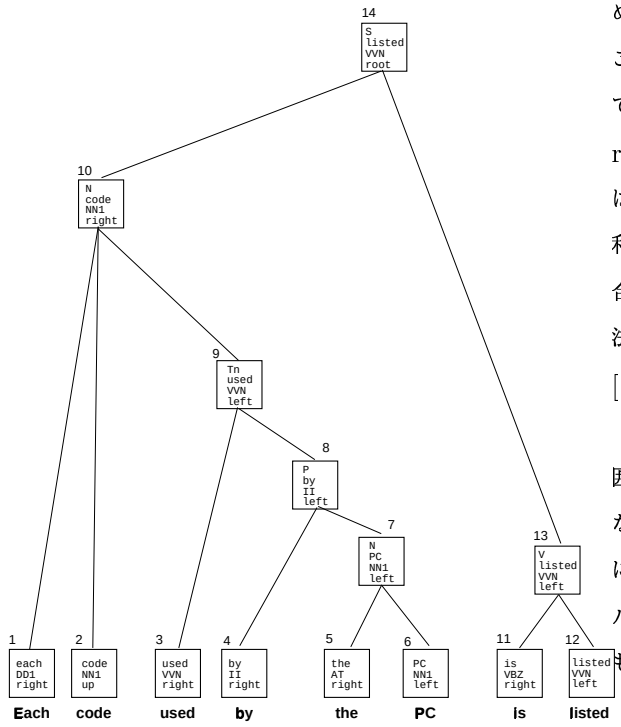
Magerman は Black らの考え方をさらに発展させ、導出過程そのものをモデル化する方法をとった。この方法は文法規則を仮定しないのが特徴で、導出によって作られる節点のラベル (NP などの非終端記号) および構文木の展開の仕方を表すラベルをそれぞれ別の決定木を用いて学習する。この後者のラベルは left, up, right, unary, root の五つからなり、どのように構文木を作り上げていくかということを表す。 left, up, right のラベルがこの順に付与された場合、例えば、図 8のように一つの親ノード (一段階上のノード) にまとめられる。 unary は子ノード (一段階下のノード) を一つしかもたず、 root は構文木の根のノードを表す。 例えば、図 9の構文木の導出



(文献[31] より)

図 8: left, up, right のラベル付与により作られる構文木の例

では、 5 番目の単語 “the” が right に、 6 番目の “PC” が left にラベル付けされ、その次の導出過程として 5 と 6 番目の単語がまとめられて left というラベルが付与されている。それぞれの過程でラベルを付与する確率は、式 (12) の $P(r_i|E[t_i^-])$



(文献[31]より)

図 9: Magermanの方法による決定木を用いた構文解析の例

と同様にそれまでの導出の履歴に基づいて決められる。そして、それぞれの確率を掛け合わせたもの、つまり式 (12) の $P(T,S)$ が最大になるように全体の構文木が作られる。構文木は左から右へ、ボトムアップにすべての可能性を保ちながら構成される。すべての可能性を考えると組み合わせが爆発するという問題が生じるが、スタックデコーダアルゴリズムという方法により解消している[31]。このMagermanの方法では付与するべきラベルを学習するだけなので品詞付けも同じように行うことができ、形態素解析と構文解析は同時に行われる。柏岡らはこの方法を日本語に適用した[37]。

Ratnaparkhiは履歴に基づく文法の考え方を2.4節で説明した最大エントロピーモデルを用いて実現した[38]。現在英語の構文解析では最も精度が高い。この方法もMagermanの方法と同様に文法規則を仮定せず、構文木を組み上げていく過程を学習する。Magermanの決定木モデルとの顕著な違いは、Magermanの方法では語を予めクラスタリングしておく必要があるのに対してこの方法ではその必要がない点にある。この方法で学習しているのは、shift-reduce法のshiftとreduceのアクションに相当する。shift-reduce法は一般化LR法の基礎となる方法で、スタックを利用し、shiftとreduceという二つの操作を組み合わせることによってボトムアップに解析する方法である[11]。

[**複数の規則の統合**]

局所的にみると決まらないが、文全体の広い範囲を見たときに構造が確実に決まるような定型的な表現は、予め数多く集めておきそれらを優先的に使うようにするのがよさそうである。これらのパターンは個々の規則を統合したものとみることもできる。

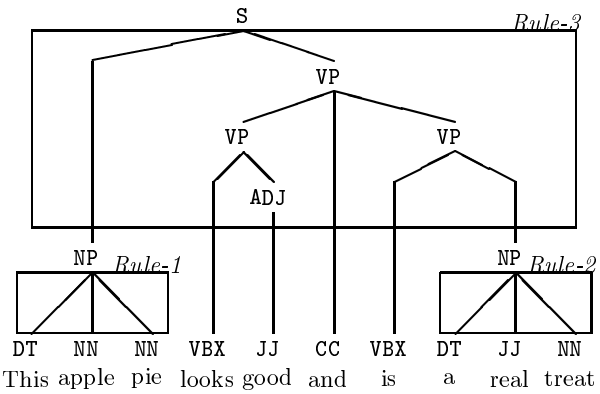
Bodは構文情報が付与されたコーパスである括弧付きコーパスから部分木を集め、コーパスによく現れたものを優先的につなぎ合わせることによって構文木を求める方法を提案した[39]。部分木の組み合わせの数は膨大なものとなるので、準最適な解を求めるためにモンテカルロ法と呼ばれる近似解法が用いられる。

一般に広い範囲の情報を考慮した規則はコーパス中に多くは出現しないが、関根らは規則に用いる非終端記号 (生成規則の左側に現れ得る記号) を S と NP の二つだけとすることによって有意に現れる長い規則を獲得することを可能とした[40]。図 10は、この方法により獲得された規則とその規則を用いたときの解析木の例である。規則は木構造を表現するためにリスト構造で表されている。

3.1.4 文法の枠組の拡張

文脈自由文法で扱われる非終端記号では、英語にみられる数や人称の一致を扱うことができない。そこで、非終端記号に数や人称などを表す素性を付与し、生成規則を適用する際に適用条件としてそれらの値の一致具合を単一化という方法を用いて調べるように拡張した単一化文法[12]と

[11] shift-reduce法、一般化LR法については、文献[5]第4章でやさしく解説されている。



Rule-3:

```
S -> NP VBX JJ CC VBX NP
 :structure"(S<1>(VP(VP<2>(ADJ<3>))<4>(VP<5><6>)))";
```

(文献[40] より)

図 10: 構文解析木と規則の例

いう枠組が考えられた。この考え方を採り入れた文法には HPSG(Head-driven Phrase Structure Grammar, 主辞駆動句構造文法) をはじめ様々なものがあるが、単一化という作業には膨大な時間が掛かるため、実用的に解析できるようなシステムは作られなかった。近年、単一化の作業の部分を分離するなどして高速に処理できるようになり[41][42][43][44]、言語学との連携もはかられるようになってきた。

他に、記号列を書き換える文法規則ではなく、木構造を書き換える文法規則をもった文法の枠組がある。この文法は TAG(Tree-Adjoining Grammar) と呼ばれ、各々の規則は木の形で与えられる。 TAG は導出において、以前に適用した規則の導出過程を導出過程木として考慮している。したがって、文脈自由文法と文脈依存文法の間の生成力をもつ弱文脈依存文法という文法のクラスに属する[45]。

3.2 依存文法に基づく解析方法

3.1節の句構造文法に基づく解析方法は、英語を中心とする言語を対象として考えられてきた方法であり、日本語のように語順の自由度が高く、省略の多い言語を句構造文法で解析するのは困難である。日本語を解析する方法としては、依存文法 (dependency grammar) を仮定する方法が有効である。これは、文節間の依存関係を決定する方法であるため、語順の変化や省略現象に強い。

日本語の係り受けには、主に以下の特徴があるとされている。

(1) 係り受けは前方から後方に向いている。

(2) 係り受け関係は交差しない。(非交差条件)

(3) 係り要素は受け要素を 1 つだけもつ。

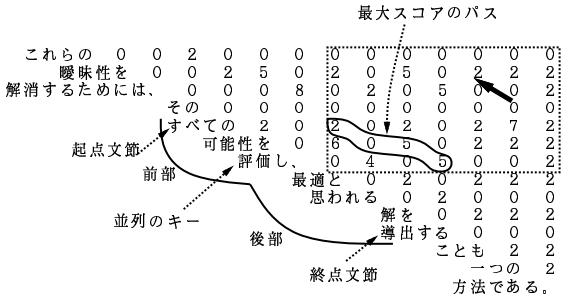
この特徴から、長い文ほど各文節の係り先の候補数が増え、解析が難しくなることが分かる。この問題に対処するために様々な工夫がなされてきた。

3.2.1 ルールベース

長い文では並列構造が現れることが多く、従属節が複数含まれることが多い。並列構造の範囲を特定したり、従属節間の係り受け関係を決定するのは簡単な規則で解決できる問題ではなく、構文解析における大きな問題となっている。

黒橋らは、一文中の並列する部分は何らかの意味において類似していると考え、類似性という観点から並列構造を検出しその範囲を特定する方法を提案した[46]。ここで、簡単にその方法を説明する。この方法ではまず、入力文中で助詞「と」や連用形の動詞 (並列キーと呼ぶ) がある部分を探し、この前後に並列構造があると仮定する。次に、その前後にあるすべての文節について二文節間の類似度を計算し、その類似度を要素とする図 11のような三角行列を考える。図 11で例えば「評価し、」という文節と「導出する」という文節の類似度は 5 であることを示している。文節間の類似度は品詞の一致、文字列の一致、シソーラスによる意味的な近さなどによって定義する。並列構造の範囲は、動的計画法を用いて図 11の行列の右下から左上にかけてスコアが最大となるような範囲を見つけることによって検出できる。並列キーを挟んでこの範囲に含まれる二つの文節列が類似した文節列である。図 11の例では「すべての可能性を評価し、」という部分と「最適と思われる解を導出する」という部分が並列であると推定され

[12] 単一化文法については、文献[5] 第 4 章を参照。



(文献[47] より)

図 11: 並列構造の推定の例

たことを表す。この方法により高い精度で並列構造の範囲を特定できるようになり、構文解析の精度も向上した[47]。現在、この方法を実装した構文解析システムが公開されている[48]。

白井らは南[49]の提案した従属節の三階層の分類に基づいて、以下のような三階層の従属節分類を提案している[50]。

A類 「同時」の表現。「～とともに」、「～ながら」、「～つつ」など7種類。

B類 「原因」、「中止」の表現。連用形単独、「～て」、「～ため」など46種類。

C類 「独立」の表現。「～が」の一種類。

これらは新聞記事に現れた従属節を元に分類されたものである。そして、A類<B類<C類のように優先度を決め、係り受け関係の決定には

- 優先度の低いものは、高いものに係る。
- 優先度の高いものは、低いものに係らない。

という優先規則を用いることによって、従属節の係り先を絞り込む。その他に、読点の付与された従属節の方が、読点の付与されていない従属節よりも優先度が高いなど読点の有無に関するものを含めてさらに四つの詳細な分類を行い、従属節の間に詳細な優先度を設定している。これらの分類と優先規則を用いると、例えば、図12の例のように係り先を決めることができる。この方法によって従属節の係り先を高い精度で正しく決定することができるようになった。

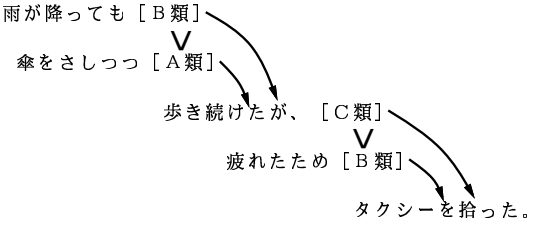


図 12: 従属節間の係り受けの例

3.2.2 係り受け確率モデル

Collinsは依存文法の考え方を統計的構文解析に導入した[51]。英語文を句の列に分解した後、二つの句の係り受け確率を計算することにより高い解析精度を得ている。係り受け確率の計算には、句の主辞にあたる語の共起確率、句と句の間の距離などの属性を用いる。この手法の成功の秘訣は、言語的なまとまりとして句を選びその係り受け関係を考えた点にあると考えられている。

藤尾らは、Collinsの方法を句の代わりに文節を用いて日本語に適用した[52]。文節は、前の主辞にあたる部分と後ろの助詞や活用形にあたる部分に分けて考え、それぞれの属性とともに文節間の距離や句読点のあるなしなども属性として考慮した。

春野らは、藤尾らが用いた属性に加えて括弧のあるなしや文節間の「は」のあるなしなど様々な属性を考慮した。様々な属性を扱うことができるように、係り受け確率の計算には2.3節に説明した決定木モデルを用いている[53]。さらに決定木をブースティング法で混合することによってデータスパースネスの問題を緩和させ、良い精度を得ている[54]。

関根らは、3.2節の冒頭にあげたような日本語の係り受けの特徴に加えて、

(4) ほとんどの場合、係り先決定には前方の文脈を必要としない。

という特徴があることに着目し、統計的手法と文末から文頭に向けて解析する方法を組み合わせることにより高い解析精度を得た[55]。(4)の特徴はあまり議論されてはいないが、人間に対する実験で90%以上の割合で成立することが確認されてい

る[55]。統計的手法としては最大エントロピーモデルを用いており、属性としては、春野らの属性に加えて属性間の組み合わせや、係り側の文節と同じ助詞や活用形が文節間にもあるかどうかなどの属性も考慮し、その有効性を示している[56]。この方法では他の手法に比べて学習データの大きさが 10 分の 1 程度であるにも関わらず、同程度以上の精度が得られている。このようにデータスパースネスに強いのは最大エントロピーモデルの特徴の一つである。

ここで、文末から解析する手法の手順について説明する。入力文は形態素解析、文節区切認定まで終っていると仮定する。解析は次の手順で行う。

手順

1. 一番最後の文節から係り先を考える。最後の文節には係り先はない。
2. 次に一つ前の文節を考える。図 13の一番左の図のように最後から二つ目の文節は最後の文節にしか係り得ない。
3. 次に最後から三つ目の文節について考える。この文節の係り先の候補は、最後から二つ目か、最後の文節かのいずれかである。最大エントロピーモデルを用いて計算される係り受け確率をスコアとし、図 13の真中のように両方の解析結果を取っておく。(スコアは図の上からそれぞれ 0.9、 0.1 だったとする。)
4. 次に最後から四つ目の文節について考える。図 13の真中の上の解析結果を基にすると、非交差条件により、この文節は文節 $(N-2)$ か最後の文節かの二通りの係り先しかもたない。それぞれの解析のスコアとしては文節 $(N-3)$、文節 $(N-2)$、文節 $(N-1)$ に関して最大エントロピーモデルを用いて計算される確率を与える。一方、図 13の真中の下の解析結果を基にすると係り先は 3 つ考えられる。
5. このような解析を文頭まで繰りかえす。文頭まで解析が終ったら、一番良いスコアの結果を解とする。

このように文末から文頭に向けて解析した場合に

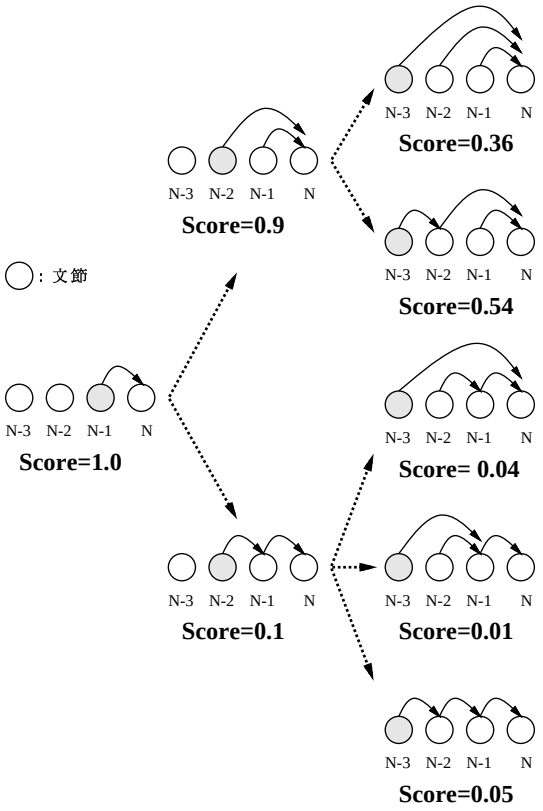


図 13: 文末から解析するときの解析例

効率良く解析ができるのは以下の二つの利点が考えられるためである。今、文節長 N の文の解析において M+1 番目の文節まで解析が終了していると仮定し、現在 M 番目の文節の係り先を決定しようとしているとする (M<N)。まず、一つ目の利点は、 M 番目の文節の係り先は、すでに解析を終了している M+1 番目から N 番目の文節のいずれかであるということである。したがって、未解決な解析状態を積み上げておく必要がない。別の利点は、 M 番目の文節の解析を開始する時点には、 M+1 番目から N 番目の係り受け解析はなんらかの形式において終了しており、可能な係り先は、非交差条件を満足する文節だけに絞られるということである。

統計的な手法では、ルールベースに比べて並列構造や従属節間の係り受け関係に対する解析誤りが多い。西岡山らは、この後者の問題を取り上げ、 Yarowsky の決定リスト (Decision List)[57] という方法を用いて解析することによってルール

ベースを上回る結果を得ている[58]。決定リストというのは、決定木の根から葉節点に至るまでのすべてのパスを展開したものを規則として優先度の高いもの順に並べ、優先度の高いものから優先的に適用する方法である。例えば、2.3節にあげた図3の決定木の例からは、次のような規則が得られる。

(見出し語 = に)&(一つ後の品詞 = 動詞)
&(一つ前の品詞 = 名詞) → P(格助詞) = 0.8

これは形態素解析の例であるが、構文解析についても同様にして規則を得ることができる。

係り受け確率を計算するモデルとしては、二つの文節の関係を係るか係らないかの二カテゴリとして学習し、係り元と係り先の文節の間の関係のみ考慮しているものが多い。しかし日本語の係り受けでは、まわりの文節との兼ね合いで係り先が決まるということが多いため、それぞれの文節から得られる情報を総合的に判断して係り先を決定したい。西岡山らは二つの文節の関係が係るか係らずに越えるかを学習するモデルを提案した[58]。このモデルを用いることにより、二つの文節だけでなくその間にある文節との関係も扱えるようになる。内元らは、(A) 二つの文節 (前文節と後文節) の関係を「前文節が後文節を越える」か「前文節が後文節に係る」か「前文節が後文節より手前の文節に係る」かの三カテゴリとして学習し、(B) 着目している二つの文節が係るか係らないかという情報に加えて、前文節が、後文節より前方にある文節に対しては越え、後文節より後方にある文節に対しては係らずにそれより手前にある文節に係るという情報も扱うことによって、前文節より後方にあるすべての文節を考慮した形で係り受け確率を求めるモデルを提案した[59]。二分類を学習する係り受け確率モデルと同じ素性を用いた実験で比較し、このモデルの方が良い解析精度が得られることを示している。

4 おわりに

本稿では言語解析の前半部分の「形態素解析」「構文解析」について解説した。これらの処理は機械翻訳、情報検索、自然言語インターフェース (ワープロの仮名漢字変換など) などに幅広く用いられている。

現在の研究の動向から考えて、今後発展する方向についてキーワードをあげるとすれば、それは以下の三点であろう。

- 分野依存

  どの分野に対しても高精度で解析できることが望ましいが、分野が変わると解析精度が下がることが多いため、容易にそれぞれの分野に適応できるシステムが期待される。コーパスベースのシステムは、分野に応じて学習し直した場合に解析精度が上がるという報告[60]があり、その期待に答えることができる有力候補である。その場合、分野が変わるとすべてを学習し直すのは効率が悪いので、どの分野にも普遍な規則と、分野に依存する部分を分けて、分野依存の部分はコーパスから学習するようにするのが良さそうである。

- 統合処理

  本稿で解説した二つの処理は、それぞれ独立に処理を行うより、二つの処理を統合して同時に行う方が整合性がよくなるため望ましい。さらに、様々な分野へ利用する場合を考えると、どのような分野に対しても精度良く解析できる汎用的なシステムが望まれる。このような汎用的なシステムは、それぞれの分野に依存した複数のモデルを統合することで実現されそうである。

- 部分解析

  精度を重視すれば、現在の解析精度では物足りない面がある。しかし、このような場合には部分解析という立場をとることにより、部分的ではあるが高精度の解析結果を得ることが可能である[61]。例えば、情報検索などで係り受け関係を利用する場合など、部分的であったとしても高精度の解析結果を利用できれば、より良い検索結果が期待できる。

本文の脚注にあげたもの以外に教科書として、文献[62][63][64]をお薦めする。

参考文献

(1) 村田真樹, 井佐原均. 意味文脈解析. 人文学と情報処理, No. 21, pp. 30–36, 1999.

(2) 黒橋禎夫, 長尾眞. 日本語形態素解析システム JUMAN 使用説明書 version 3.5. 京都大学大学院工学研究科, 1998.

(3) 松本裕治, 影山太郎, 永田昌明, 齋藤洋典, 徳永健伸. 単語と辞書, 岩波講座言語の科学, 第 3 巻. 岩波書店, 1997.

(4) Masaaki Nagata. A Stochastic Japanese Morphological Analyzer Using a Forward-DP Backward-A^* N-Best Search Algorithm. In *Proceedings of the 15th International Conference on Computational Linguistics (COLING94)*, pp. 201–207, 1994.

(5) 長尾眞, 佐藤理史, 黒橋禎夫, 角田達彦. 自然言語処理, 岩波講座ソフトウェア科学, 第 15 巻. 岩波書店, 1996.

(6) 山地治, 黒橋禎夫, 長尾眞. 連語登録による形態素解析システム JUMAN の精度向上. 言語処理学会 第 2 回 年次大会 発表論文集, pp. 73–76, 1996.

(7) 北内啓, 山下達雄, 松本裕治. 日本語形態素解析システムへの可変長連接規則の実装. 言語処理学会 第 3 回 年次大会 発表論文集, pp. 437–440, 1997.

(8) 竹沢寿幸, 末松博. 音声・テキストコーパスとその構築技術, 標準化動向. 人工知能学会, Vol. 10, No. 2, pp. 4–16, 1995.

(9) 新情報処理開発機構. RWC テキストデータベース第二版. 1998.

(10) 黒橋禎夫, 長尾眞. 京都大学テキストコーパス・プロジェクト. 言語処理学会 第 3 回 年次大会 発表論文集, pp. 115–118, 1997.

(11) Doung Cutting, Julian Kupiec, Jan Pedersen, and Penelope Sibun. A Practical Part-of-Speech Tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing*, pp. 133–140, 1992.

(12) 竹内孔一, 松本裕治. 隠れマルコフモデルによる日本語形態素解析のパラメータ推定. 情報処理学会論文誌, Vol. 83, No. 3, pp. 500–509, 1997.

(13) L. E. Baum. An inequality and associated maximization technique in statistical estimation for probabilistic functions of markov process. *Inequalities*, Vol. 3, pp. 1–8, 1972.

(14) Peter F. Brown, Vincent J. Della Pietra, Peter V. deSouza, Jenifer C. Lai, and Robert L. Mercer. Class-based n-gram models of natural language. *Computational Linguistics*, Vol. 18, No. 4, pp. 467–479, 1992.

(15) 森信介, 長尾眞. 形態素クラスタリングによる形態素解析精度の向上. 自然言語処理, Vol. 5, No. 2, pp. 75–103, 1998.

(16) 北研二, 中村哲, 永田昌明. 音声言語処理 — コーパスに基づくアプローチ —. 森北出版, 1996.

(17) 北内啓, 宇津呂武仁, 松本裕治. 誤り駆動型の確率モデル学習による日本語形態素解析. 情報処理学会 自然言語処理研究会 NL124-6, pp. 41–48, 1998.

(18) Hinrich Schütze and Yoram Singer. Part-of-Speech Tagging Using a Variable Memory Markov Model. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 181–187, 1994.

(19) Masahiko Haruno and Yuji Matsumoto. Mistake-Driven Mixture of Hierarchical-Tag Context Trees. In *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 230–237, 1997.

(20) Ralph Weischedel, Marie Meteer, Richard Schwartz, Lance Ramshaw, and Jeff Palmucci. Coping with ambiguity and un-

known words through probabilistic models. *Computational Linguistics*, Vol. 19, No. 2, pp. 359–382, 1994.

(21) Walter Daelemans, Jakub Zavrel, Peter Berck, and Steven Gills. MBT: A Memory-Based Part-of-Speech Tagger-Generator. In *Proceedings of the 4th Workshop on Very Large Corpora*, pp. 1–14, 1996.

(22) Adwait Ratnaparkhi. A Maximum Entropy Model for Part-Of-Speech Tagging. In *Conference on Empirical Methods in Natural Language Processing*, pp. 133–142, 1996.

(23) Helmut Schmid. Part-Of-Speech Tagging with Neural Networks. In *Proceedings of the 15th International Conference on Computational Linguistics (COLING94)*, pp. 172–176, 1994.

(24) Qing Ma, Hitoshi Isahara, and Hiromi Ozaku. Automatic part-of-speech tagging of thai corpus using neural networks. *Artificial Neural Networks - ICANN 96, Lecture Notes in Computer Science 1112*, pp. 275–280, 1996.

(25) 馬青, 井佐原均. 長さ可変文脈を用いたマルチニューロタガー. 自然言語処理, Vol. 6, No. 1, pp. 29–42, 1999.

(26) Eric Brill. Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part-of-Speech Tagging. *Computational Linguistics*, Vol. 21, No. 4, pp. 543–565, 1995.

(27) 久光徹, 丹羽芳樹. 書き換え規則と文脈情報を用いた形態素解析後処理. 情報処理学会 自然言語処理研究会 NL126-8, pp. 55–62, 1998.

(28) 馬青, 内元清貴, 村田真樹, 井佐原均. ニューラルネットとルールベース手法を統合した品詞タグ づけシステム. 言語処理学会 第 5 回 年次大会 発表論文集, pp. 293–296, 1999.

(29) Lluís Màrquez and Lluís Padró. A Flexible POS Tagger Using an Automatically Acquired Language Model. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 238–252, 1997.

(30) Hans van Halteren, Jakub Zavrel, and Walter Daelemans. Improving Data Driven Wordclass Tagging by System Combination. In *Proceedings of the COLING-ACL '98*, pp. 491–497, 1998.

(31) David M. Magerman. Statistical decision-tree models for parsing. *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 276–283, 1995.

(32) Eugene Charniak. Statistical parsing with a context-free grammar and word statistics. *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, pp. 598–603, 1997.

(33) Michael Collins. Three generative, lexicalized models for statistical parsing. *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 16–23, 1997.

(34) Shinsuke Mori and Makoto Nagao. A Stochastic Language Model using Dependency and Its Improvement by Word Clustering. In *Proceedings of the COLING-ACL '98*, pp. 898–904, 1998.

(35) 白井清昭, 乾健太郎, 徳永健伸, 田中穂積. 統計的構文解析における構文的統計情報と語彙的統計情報 の統合について. 自然言語処理, Vol. 5, No. 3, pp. 85–106, 1998.

(36) Ezra Black, Fred Jelinek, John Lafferty, David M. Magerman, Robert Mercer, and Salim Roukos. Towards history-based grammars: Using richer models for

probabilistic parsing. *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 31–37, 1993.

(37) 柏岡秀紀, 河田康裕, 金城由美子, Andrew Finch, Ezra Black. 確率付き決定木を用いた日本語構文解析. 言語処理学会 第 4 回 年次大会 発表論文集, pp. 213–216, 1998.

(38) Adwait Ratnaparkhi. A Linear Observed Time Statistical Parser Based on Maximum Entropy Models. In *Conference on Empirical Methods in Natural Language Processing*, 1997.

(39) Rens Bod. Using an annotated corpus as a stochastic grammar. *Proceedings of the Sixth Conference of the European Chapter of the Association for Computational Linguistics (EACL'93)*, pp. 37–44, 1993.

(40) Satoshi Sekine and Ralph Grishman. A corpus-based probabilistic grammar with only two non-terminals. *Proceedings of the 4th International Workshop on Parsing Technology*, pp. 216–223, 1995.

(41) Kentaro Torisawa and Jun'ichi Tsujii. Computing phrasal-signs in hpsg prior to parsing. *Proceedings of the 16th International Conference on Computational Linguistics (COLING96)*, pp. 949–955, 1996.

(42) Takaki Makino, Minoru Yoshida, Kentaro Torisawa, and Jun'ichi Tsujii. Lilfes—towards a practical hpsg parser. *Proceedings of the COLING-ACL '98*, pp. 807–811, 1998.

(43) Yutaka Mitsuishi, Kentaro Torisawa, and Jun'ichi Tsujii. Hpsg-style underspecified japanese grammar with wide coverage. *Proceedings of the COLING-ACL '98*, pp. 876–880, 1998.

(44) Takashi Ninomiya, Kentaro Torisawa, and Jun'ichi Tsujii. An efficient parallel substrate for typed feature structures on shared memory parallel machines. *Proceedings of the COLING-ACL '98*, pp. 968–974, 1998.

(45) The XTAG Research Group. A lexicalized tree adjoining grammar for english. http://linc.cis.upenn.edu/ xtag/tech-report/tech-report.html.

(46) 黒橋禎夫, 長尾眞. 長い日本語文における並列構造の推定. 情報処理学会論文誌, Vol. 33, No. 8, pp. 1022–1031, 1992.

(47) 黒橋禎夫, 長尾眞. 並列構造の検出に基づく長い日本語文の構文解析. 自然言語処理, Vol. 1, No. 1, pp. 35–57, 1994.

(48) 黒橋禎夫. 日本語構文解析システム KNP 使用説明書 Version 2.0b6. 京都大学大学院情報学研究科, 1998.

(49) 南不二男. 現代日本語文法の輪郭. 大修館書店, 1993.

(50) 白井諭, 池原悟, 横尾昭男, 木村淳子. 階層的認識構造に着目した日本語従属節間の係り受け解析の方法とその精度. 情報処理学会論文誌, Vol. 36, No. 10, pp. 2353–2361, 1995.

(51) Michael Collins. A new statistical parser based on bigram lexical dependencies. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 184–191, 1996.

(52) 藤尾正和, 松本裕治. 統計的手法を用いた係り受け解析. 情報処理学会 自然言語処理研究会 NL117-12, pp. 83–90, 1997.

(53) 春野雅彦, 白井諭, 大山芳史. 決定木を利用した日本語係り受け解析. 自然言語処理シンポジウム'97「実用的な自然言語処理に向けて」, http://www.csl.sony.co.jp/person/nagao/nlsym97/, 1997.

(54) 春野雅彦, 白井諭, 大山芳史. 決定木の混合を利用した日本語係り受け解析. 言語処理学会

第 4 回 年次大会 発表論文集, pp. 217–220, 1998.

(55) 関根聡, 内元清貴, 井佐原均. 文末から解析する統計的係り受け解析アルゴリズム. 自然言語処理, Vol. 6, No. 3, pp. 59–73, 1999.

(56) 内元清貴, 関根聡, 井佐原均. ＭＥによる日本語係り受け解析. 情報処理学会 自然言語処理研究会 NL128-5, pp. 31–38, 1998.

(57) David Yarowsky. Decision Lists For Lexical Ambiguity Resolution: Application to Accent Restoration in Spanish and French. In *32th Annual Meeting of the Associtation of the Computational Linguistics (ACL)*, pp. 88–95, 1994.

(58) 西岡山滋之, 宇津呂武仁, 松本裕治. コーパスからの日本語従属節係り受け選好情報の抽出. 情報処理学会 自然言語処理研究会 NL126-5, pp. 31–38, 1998.

(59) 内元清貴, 村田真樹, 関根聡, 井佐原均. 日本語係り受け解析に用いるＭＥモデルと解析精度. 言語処理学会 第 5 回 年次大会 併設ワークショップ, pp. 41–48, 1999.

(60) Satoshi Sekine. The domain dependence of parsing. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pp. 96–102, 1997.

(61) 乾健太郎, 白井清昭, 田中穂積, 徳永健伸. 統計に基づく部分係り受け解析. 言語処理学会 第 4 回 年次大会 発表論文集, pp. 386–389, 1998.

(62) James Allen. *Nutural Language Understanding*. Benjamin/Cummings Publishing Company, 2nd edition, 1994.

(63) Eugene Charniak. *Statistical Language Learning*. MIT PRESS, 1993.

(64) 中川聖一. 確率モデルによる音声認識. 社団法人電子情報通信学会, 1988.