自然言語処理 112－5
（1996．3．14）

文脈木を利用した形態素解析

春野 雅彦* 松本 裕治†
*NTT コミュニケーション科学研究所
†奈良先端科学技術大学院大学

本稿では文脈木を利用した形態素解析手法を提案する。文脈木はデータ圧縮の一手法である算術符号において、次に現われる記号の出現確率を正確に評価するために導入されたものである。我々が日本語の形態素解析に文脈木を利用するのは以下の2つの理由からである。(1) 文脈木を利用することで確率の予測に最適な長さの文脈(過去の系列)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出来る。これは情報量が異なる複数の文字種を持つ日本語の形態素解析に特に有効な特徴である。(2) 文脈木を利用することで、ある品詞の中で特殊な振舞いをする単語を自動的に見つけ出すことが可能となる。これらの単語を別の形態素として分離することで、適切に語彙化された接続規則を自動的に生成出来る。

Japanese Morphological Analyzer Using Context Tree

Masahiko Haruno* Yuji Matsumoto†
*NTT Communication Science Laboratories
†Nara Institute of Science and Technology

This paper describes a context-tree based approach to Japanese morphological analysis. The context tree is originally introduced for the use in arithmetic coding and accurately evaluates the next symbol probability. Compared to conventional methods, our context tree based approach has the following advantages: (1) The optimal length of context is selected by virtue of the context tree, resulting in the reduction of probability parameters. (2) Lexicalized rules can be automatically constructed. Context tree makes it possible information-theoretically to identify the words that behave differently from others of the same part-of-speech. Such words are to form lexicalized rules.

1 はじめに

近年テキストの電子化に伴い、コーパスに基づく自然言語処理の研究が盛んになっている。中でも英語の品詞付けや日本語の形態素解析では tri-gram モデルに基づく統計的方法で95%程度の精度を達成しており、人手で作成した規則を凌ぐようになっている [1, 5]。しかし形態素解析の精度が検索、機械翻訳等のあらゆる現実的応用に影響することを考えると、この精度は十分なものではない。統計に基づく手法でこれ以上の精度を達成するためには、主として以下の2つの可能性がある。

モデルの次数を増やす tri-gram のモデルの次数は2であるが、これ3,4と増やせば後続形態素の予測確率はより正確になると考えられる。

例外的な単語の接続を規則化する ある一つの形態素の中でも他の単語と違った振舞いをする例外的な単語が存在する。これらも接続規則として登録出来れば形態素解析の精度が上がると考えられる。

しかしながら、既存の n-gram 統計に基づく方法でこれらの拡張を行なうことは困難である。

モデルの次数を大きくすればパラメータ数が指数的に増加するとともに、学習に必要なデータの量も著しく増加する。この問題は主に n-gram 統計では全ての文脈に対して同じ n を仮定していることに起因する。実際には、言語現象に応じて、大きい n を必要とするものもあれば、小さくてすむものもある。可変な n を許容するような枠組を導入する必要がある。また、第2の問題に関して tri-gram 以上のモデルでは人手で例外的な単語を列挙することは難しいし、行なったとしてもシステムの保守性が極めて悪くなる。従って、何らかの客観的基準に基づいて例外的単語を見つける手法が必要になる。

本稿では上記2つの問題を解決するため、文脈木を利用した形態素解析手法を提案する。文脈木 [7, 9, 10] はデータ圧縮の一手法である算術符号において、次に現われる記号の出現確率を正確に評価するために導入されたものである。文脈木を用いた算術符号では、個々の文脈に応じて情報理論的に最適なモデル次数が決定されるため、少ないパラメータ数で高精度の予測を行なうことが出来る。この符合化法は理想符合長への高速な収束も証明されている [9]。このような数理的性質を持つため、最近は機械学習の分野でも利用され始めている [8, 3]。

我々が日本語の形態素解析に文脈木を利用するのは以下の2つの理由による。文脈木を利用することで確率の予測に最適な長さの文脈 (過去の系列) を決定することが可能となり、パラメータ数を減らすことが出来る。これは情報量が異なる複数の文字種を持つ日本語の形態素解析では特に有効な特徴である。次に、文脈木を利用することである品詞中で特殊な振舞いをする単語を自動的に見つけ出すことが出来る。これらの単語を別の形態素として分離することで適切に語彙化された接続規則を自動的に生成することが可能となる。

本稿の構成は以下の通りである。2章で確率的形態素解析のモデルを概観し、本稿で取り扱う問題について説明する。3章では基本的な文脈木の構成法と最適文脈の決定法を説明した後、これらの方法が例外的な単語を見つける上でも有効であることを説明する。4章では、効率化のため文脈木を等価な確率オートマトンに変換する方法を述べ、5章でまとめる。

2 確率的形態素解析モデル

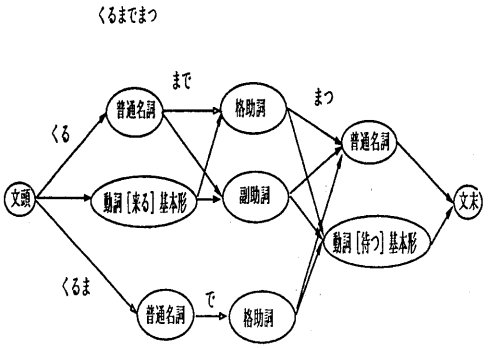

図 1: 確率的形態素解析システムの概要

図 2に入力文字列'くるまでまつ'に対する確率的形態素解析システムの動作を示す。あらゆる位置における形態素の可能候補は図の様に'文頭'で始まり、'文末'で終るラティス上に表現される。形態素と形態素を結ぶパス上にはそれまでの系列に続いて次の形態素が現われる予測確率が付けられる。この予測確率をどのようにモデ

ル化するかによって様々な形態素解析モデルが考えられている。確率的形態素解析とはこのラティス上から'文頭'で始まり'文末'で終る最尤なパスを見つけることに等しい。つまり入力された文字列Lに対してその生起確率を最大化するような形態素の列$t_{1,n}$を求めれば良いことになる。

$$argmax_{t_{1,n}} P(w_{1,n}, t_{1,n}|L) =$$
$$\arg\max_{t_{1,n}} \frac{P(w_{1,n}, t_{1,n}, L)}{P(L)}$$
$$\Leftrightarrow argmax_{t_{1,n}, w_{1,n} \in L} P(t_{1,n}, w_{1,n})$$

$P(t_{1,n}, w_{1,n})$を更に変形すると

$$P(t_{1,n}, w_{1,n}) = P(w_{1,n})P(t_1|w_1)P(w_2|t_1, w_1)P(t_2|t_1, w_{1,2})$$
$$\cdots P(t_n|t_{1,n-1}, w_{1,n})P(w_n|t_{1,n-1}, w_{1,n-1})$$
$$= \prod_{i=1}^{n} P(w_i|t_{1,i-1}, w_{1,i-1})P(t_i|t_{1,i-1}, w_{1,i})$$

を得る。

通常は単語の出現確率はその形態素だけで決まり、形態素の出現確率は過去の形態素の系列だけで決まると近似し、最大化すべき式を以下のようにする。

$$argmax \prod_{i=1}^{n} P(w_i|t_i)P(t_i|t_{1,i-1}) \qquad (1)$$

(1)式において$P(t_i|t_{1,i-1})$の評価に1次、2次のモデルを用いるのが各々bi-gram、tri-gramモデルである。我々が形態素解析に文脈木Tを用いるは、$P(t_i|t_{1,i-1}, w_{1,i})$を$P(t_i|T)$として効率良く評価するためである。

bi-gram、tri-gramモデルでは(1)式の最大化問題に、動的計画法に基づくViterbiアルゴリズム[11]が利用される。しかしながら、文脈木を含む木情報源は一般にユニフィラー情報源ではないため、入力記号から次状態が一意に決まらない。従ってそのままではViterbiアルゴリズムを適用することが出来ない。そのため、4章で文脈木を等価な確率オートマトンに変換する手法について述べる。この変換によって効率的な実装が可能となる。

3 文脈木を用いた確率予測

この章では文脈木の構成法、最適モデル次数の決定法を説明した後、例外的な単語を発見しながら文脈木を構成するアルゴリズムを説明する。

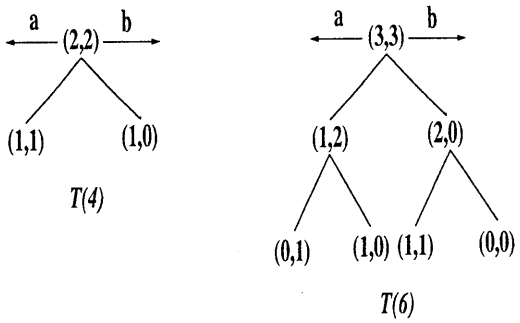

図2: 'baabab'に対する文脈木の成長

3.1 文脈木の構成法

文脈木は過去の系列に対する記号の出現頻度表(カウント)を木の状態で蓄えたものである。文脈木中で深さ1の節点は1個前までの文脈を考慮した記号のカウントを表し、深さ2の節点は2個前までの文脈を考慮した記号のカウントを表す。同様にして木の根から深さd辿った節点はd個前まで文脈を考慮した場合の記号のカウントである。

図2は文字列'baabab'によって生成される記号が2種類('a'と'b')の文脈木の例で、$T(4)$は記号列'baab'まで読み込んだ状態の文脈木である。$T(4)$は以下の出現頻度表と等価である。深さ1の節点で文脈'a'を見ると'a'に続いて'a'が1回、'b'が1回出現していることが分かり、文脈'b'を見ると'b'に続いて'a'が1回出現したことが分かる。

節点の深さ	文脈	aの出現回数	bの出現回数
0	なし	2	2
1	'a'	1	1
1	'b'	1	0

一般に記号数が予め分かっている場合に、データ系列から文脈木を構成する方法は以下の通りである。

1. 根だけからなる木を用意し根には全ての記号に対してカウント0を付与する。

2. 以下の手続きを再帰的に適用する. $T(t\text{-}1)$を最も最近に構成された木とする。次のシンボル$x(t)$が入力されると以下の操作に従って木T(t)を構成する。$T(t\text{-}1)$を根から辿り始めて$x(t\text{-}1)$,$x(t\text{-}2)$, $\cdots$に従っ

て進む。途中訪れた節点に対して記号 $x(t)$ のカウントをインクリメントする。この操作を最も深い節点に到達するまで行なう。

3. 最後にインクリメントした節点の記号 $x(t)$ に関するカウントが少なくとも2になれば、新たな子節点を作り、$x(t)$ のカウントを除き(このカウントは1)全ての記号のカウントを0にする。こうして出来る木を $T(t)$ とする。

3.2 最適文脈の決定法

前節で説明した文脈木は、あらゆる文脈における各記号の出現回数をカウントとして集計しただけのもので、特定の文脈に対して何次のモデルを使うのが良いのかは分からない。この節では情報理論的に最適なモデル選択法を説明する。

記号の集合を A とする。ある節点 s において(木の根から s 至る文脈で)、記号 a が出現する回数、確率を各々 $n(a|s)$、$\hat{P}(a|s)$ とすると

$$\hat{P}(a|s) = \frac{n(a|s)}{\sum_{b \subseteq A} n(b|s)}$$

となる。

ここで親ノード節点 s の代わりに、s の子節点 $sb, b \subseteq A$ を用いた場合に得られる利得関数 $\Delta(sb)$[9] を次の様に定義する。

$$\Delta(sb) = \sum_{a \subseteq A} n(a|sb) log \frac{\hat{P}(a|sb)}{\hat{P}(a|s)} \qquad (2)$$

この $\Delta(sb)$ は文脈 sb で出現した全ての記号を sb の確率分布を使って符合化した場合の理想符合長と s での確率分布を使って符合化した場合の理想符合長との差であり、節点を1つ展開したことの利得を表現するのに適切なものである。

更に文脈木 $T(t)$ に対して $T(t)$ の最適な節点集合 S_t を以下の様に定義する。ただし C は定数である。

{以下の条件を満たす$T(t)$の最深節点$w|\Delta(w) \geq Clog(t+1)$}

このように S_t を選んだ時に P を真の分布、$\hat{P}$ を S_t を用いた場合の分布とすれば長さ n の系列 X_n に対して次の定理が成り立つことが知られている[9]。ただし d は記号総数である。

$$\frac{1}{n} log \frac{\hat{P}(X_n)}{P(X_n)} \leq \frac{K(d-1)}{2n} logn + O(\frac{1}{n})$$

この定理は1記号あたりの符合長がデータ系列が長くなるにつれて、右辺に与えられる割合で理想符合長に漸近することを意味する。またこの定理は学習のために必要なデータ量の概略も与える。

このようにして得られた文脈木 T の最適節点集合 S_t を用いた、記号 $a \subseteq A$ の予測確率はラプラス推定量の一種を用いて(3)式の様に得られる[4]。我々の形態素解析システムではこの式で次記号の予測確率を計算する。

$$P(a|T) = \hat{P}(a|s_t) = \frac{n(a|s_t) + \frac{1}{2}}{\sum_{b \subseteq A} n(b|s_t) + \frac{d}{2}} \qquad (3)$$

3.3 文脈木を用いた接続関係の語彙化

この節では例外的な単語を自動的に見つけながら、文脈木を構成する手法について説明する。我々が利用する形態素解析済みコーパス[12]の例を表1に示す。このデータのうち訓練事例として利用するのは $< t_i, p_i >$ のペアである。ただし t_i, p_i は各々標準形態素と単語の表層を表している。標準形態素は接続規則を記述するための標準的な形態素であり、予めユーザによって指定される(もちろん標準形態素が単語の表層であっても良い)。例えば、'減益'や'見直し'の場合には < サ変名詞,減益 >、< 基本連用形,見直し > が訓練事例となる。これらのペアの要素はそれぞれ階層を成しており、どちらの情報を使って文脈木を展開するかかがここでの問題となる。

減益	げんえき	減益	名詞	サ変名詞
決算	けっさん	決算	名詞	サ変名詞
は	は	は	助詞	副助詞
経営	けいえい	経営	名詞	サ変名詞
見直し	みなおし	見直す	動詞	基本連用形
の	の	の	助詞	名詞接続助詞
好機	こうき	好機	名詞	普通名詞

表 1: 形態素解析済みコーパスの例

この問題を統一的に扱うために、(2)式を(4)式のように変形する。式中で $n(s)$ は節点(文脈)s で出現した記号ののべ総数、D_{KL} は Kullback-Leibler 情報量と呼ばれる非負数である。Kullback-Leibler 情報量は2つの確率分布の距離として頻繁に用いられる[2]。

$$\Delta(sb) = n(sb)\sum_{a\subseteq A}\frac{n(a|sb)}{n(sb)}log\frac{\hat{P}(a|sb)}{\hat{P}(a|s)}$$

$$= n(sb)\sum_{a\subseteq A}\hat{P}(a|sb)log\frac{\hat{P}(a|sb)}{\hat{P}(a|s)}$$

$$= n(sb)D_{KL}(\hat{P}(\cdot|sb),\hat{P}(\cdot|s)) \qquad (4)$$

標準形態素、表層のどちらを用いるかという問題に対して、(4)式を以下のように解釈することが出来る。$n(sb)$はsに付加する記号bが一般的、即ち標準形態素である場合に大きくなる。それに対して、$D_{KL}(\hat{P}(\cdot|sb),\hat{P}(\cdot|s))$は節点$sb$と節点$s$の確率分布としての距離を表しているのであるから、$b$が特殊な記号、即ち表層語である方が大きくなると考えられる。文脈木の展開に関する決定は両者のトレードオフに依存している。この考察から標準形態素、表層のどちらを用いて文脈木を展開するかという問題も前節で述べたのと全く同じ枠組で処理出来ることが分かる。

$< t_i, p_i >$を一つの事例であるとすると、式(4)を用いて漸次的に語彙化された文脈木を得るアルゴリズムは図3で与えられる。

形態素集合Aを標準形態素の集合とする
do
  t番目の事例$x_t(< t_t, p_t >)$を読み
  $x_{t-1}, x_{t-2}, \cdots, x_{t-(i-1)}$の文脈を辿り節点$s$に到達
  if$(max(\Delta(st_{t-i}), \Delta(sp_{t-i})) \geq Clog(t+1))$
    if$(\Delta(st_{t-i}) \leq \Delta(sp_{t-i}))$
      表層を用いて文脈木を展開
    else 通常の標準形態素で文脈木を展開
  $t = t+1$
while(x_tが存在する)

図3: 語彙化された文脈木を得るアルゴリズム

このアルゴリズムでは事例を1つ読み込むたびに$max(\Delta(st_i), \Delta(sp_i))$を計算するが、この計算は事例数の2乗のオーダーの時間を要する。展開する階層レベルを動的に決定する場合には、予めコーパスから文脈木を構成しておくことが出来ず、1つの事例を読み込むたびに全事例を調べなければならないからである。したがって現実的には、上記のアルゴリズムで予め表層レベルまで考慮する標準形態素を指定しておく方法が有効である。

4 木情報源から確率オートマトンへの変換

ここでは文脈木を確率オートマトンに変換する方法[8]を述べる。確率オートマトンを利用すると入力記号から次状態を一意に決定出来るため、Viterbiアルゴリズムによる効率的な実装が可能となる。

基本的に文脈木の各節点をオートマトンの状態に対応させる。まず問題となるのは状態sに記号σが入力された後、$s\sigma$に対応する節点が文脈木上に存在しない場合である。この場合にはオートマトンの次状態を決定することが出来ない。この問題を避けるためには、以下の条件が成り立つまで図3のアルゴリズムで得られた文脈木Tを拡大し、T'を構成すれば良い。

条件 T'の全ての葉sに対して以下が成り立つ。形態素集合Aの全ての記号σを入力した時、$s\sigma$のあるsufixがT'を根$\rightarrow\sigma\rightarrow\cdots$と辿った$T'$の節点である。

次に状態間の遷移確率について考える。状態$s1$で記号σが入力され状態$s2$に移ったとすると、その状態遷移確率$P(s_1 \rightarrow s_2|\sigma)$は(3)式を用いて

$$P(s_1 \rightarrow s_2|\sigma) = \frac{n(\sigma|s_1)+\frac{1}{2}}{\sum_{b\subseteq A} n(b|s_1)+\frac{|A|}{2}}$$

で求められる。最後に状態の初期確率としては根の確率を1にしておけば良い。2記号の場合の文脈木から確率オートマトンへの変換例を図4に示す。ここでは簡単のため状態遷移確率にラプラス推定量は用いなかった。

5 まとめ

本稿では文脈木を使った形態素解析について主に情報理論の応用という観点から述べた。本文で述べたように文脈木を形態素解析に利用することの利点は以下のようにまとめられる。

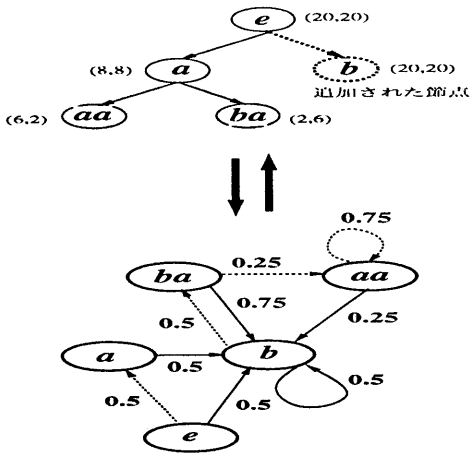

図 4: 文脈木から確率オートマトンへの変換例

1. 必要に応じて適切なモデル次数を選ぶことが出来るため、既存の方法に比べて少ないパラメータ数で長い範囲の依存関係を捉えることが可能となる。

2. 情報理論に基づく客観的な基準により、ある形態素中で特殊な振舞いをする例外的な単語を見つけることが可能となる。

我々は現在本稿に述べた手法を実装し、評価を行なっている最中である。今回は自然言語処理固有の問題については殆んど触れなかったが、実装に関する技術等については別の機会で報告する予定である。

我々がここで述べた手法は単に形態素解析に留まらず、階層が存在する状態系列に関する推測問題全般に有効であると思わる [6]。

謝辞 本稿の様々な点に関し御討論頂いた NTT の向内 隆文研究員、ならびに文脈木からオートマトンへの変換に関して詳細を教えて頂いた AT&T ベル研究所の Yoram Singer 博士に感謝致します。

参考文献

[1] Eugene Charniak, Curtis Hendrickson, Neil Jacobson, and Mike Perkowits. Equations for Part-of-Speech Tagging. In *Proc. 11th AAAI*, pages 784–789, 1993.

[2] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 1991.

[3] H.Schutze and Y.Singer. Part-of-speech tagging using a variable markov model. In *the 32th Annual Meeting of ACL*, pages 181–187, 1994.

[4] R.E. Krichevskii and V.K. Trofimov. The performance of universal encoding. *IEEE Transaction on Information Theory*, 27(2):199–207, 1981.

[5] Masaaki Nagata. A Stochastic Japanese Morphological Analyzer Using Forward-DP Backward-A^* N-Best Search Algorithm. In *Proc. 15th COLING*, pages 201–207, 1994.

[6] N. Reithinger and E. Maier. Utilizing statistical dialogue act processing in verbmobil. In *the 33th Annual Meeting of ACL*, pages 116–121, 1995.

[7] Jorma Rissanen. A universal data compression system. *IEEE Transaction on Information Theory*, 29(5):656–664, September 1983.

[8] Dana Ron, Yoram Singer, and Naftali Tishby. The power of amnesia: Learning probabilistic automata with variable memory length. (to appear) Machine Learning Special Issue on COLT94, 1996.

[9] M J. Weinberger, J J. Rissanen, and M. Feder. A universal finite memory source. *IEEE Transaction on Information Theory*, 41(3):643–652, May 1995.

[10] F M J. Willems, Y M. Shtarkov, and T J. Tjalkens. The context-tree weigting method: Basic properties. *IEEE Transaction on Information Theory*, 41(3):653–664, May 1995.

[11] 中川 聖一. 確率モデルによる音声認識. 電子情報通信学会, 1988.

[12] 松本 裕治他. 日本語形態素解析システム *JUMAN* 使用説明書 *2.0*. TR94025, 奈良先端科学技術大学院大学, 1994.