社団法人 情報処理学会　研究報告
IPSJ SIG Technical Report

2004－ＦＩ－ 76　(15)
2004－ＮＬ－163　(15)
2004／9／17

PCFGと分岐HMMを用いた構文解析

松崎 拓也 †　宮尾 祐介 †　辻井 潤一 †‡
†東京大学大学院 情報理工学系研究科
‡CREST JST
{matuzaki,yusuke,tsujii}@is.s.u-tokyo.ac.jp

概要:　本論文では、構文木の生成モデルである分岐 HMM という確率モデルを定義し、構文解析済みのコーパスを訓練データとした EM アルゴリズムを用いて分岐 HMM のパラメータ推定を行うことで、詳細化された CFG 規則を自動的に学習する手法を提案する。また、PCFG を用いて得た複数の解析候補を分岐 HMM によってリランキングする実験を行い，分岐 HMM の学習によって自動的に得た詳細化された CFG 規則が構文解析の精度を向上させることを確かめた。

Parsing with PCFG and Branching HMM

Takuya Matsuzaki†　Yusuke Miyao†　Jun'ichi Tsujii†‡
†Graduate School of Information Science and Technology, University of Tokyo
‡CREST, JST
{matuzaki,yusuke,tsujii}@is.s.u-tokyo.ac.jp

Abstract:　This paper defines a generative probability model of parse trees, which we call the branching HMM. A branching HMM can automatically learn fine-grained CFG rules from a parsed corpus by an EM-style estimation algorithm. The effectiveness of automatically learned CFG rules in parsing is demonstrated through experiments in which tentative parse results by a PCFG is re-ranked by a branching HMM.

1　はじめに

構文解析済みのコーパスから得た PCFG (Treebank-PCFG) を用いて構文解析を行う手法 [1] は Collins[2]、Charniak[3] などの高カバレッジ・高精度な統計的構文解析手法のベースとなっている。しかし、例えば Johnson[4] などが指摘しているように，確率モデルとしての Treebank-PCFG が仮定している、構文木の生成 (導出) における条件付独立性 (文脈自由性) は強すぎる仮定である。この仮定の悪影響を避けるため、PCFG ベースの構文解析に関するいくつかの既存研究では、終端・非終端記号に構文木内でその記号が出現する環境、例えば親ノードや主辞の終端記号などの情報を付加することで Treebank-CFG の規則を詳細化している [2, 3, 4, 5]。

本論文では、分岐 HMM という確率モデルを定義し、これを用いて、詳細化された CFG 規則を自動的に学習する方法を提案する。分岐 HMM は隠れ変数を終端・非終端記号とする確率文脈自由規則によって隠れ変数をノードとする木を生成し、各ノードの隠れ変数が構文木内の終端・非終端記号を生成するモデルである。分岐 HMM の学習を、解析済みコーパスを訓練データとした EM アルゴリズムで行うことにより、既存研究では主に人手で終端・非終端記号に付加する情報を選択して行っていた CFG 規則の詳細化を自動的に行うことができる。

分岐 HMM に類似した既存の確率モデルや学習の手法としては、複数の特徴的な時間スケールをもつ記号列を生成するモデルである階層 HMM[6] や、単語列あるいは部分的に括弧がついた単語列から PCFG の教師なし学習を行う手法 [7, 8] がある。これらの研究と異なり、本論文で提案する手法では非終端記号を含む構文木を訓練データとして用いる。

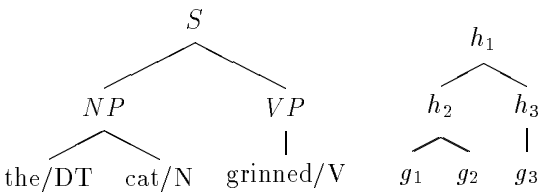

図 1: 構文木の生成の例

左 : 構文木 T

右 : T を生成した隠れた木 HT

また、宇津呂ら [9] による CFG 規則の自動的な詳細化・汎化に関する研究は本研究と共通の目的を持つものであるが、確率モデルとしての意味付けは明らかでなく、自動的な詳細化・汎化が構文解析精度に与える影響については報告されていない。

分岐 HMM $\mathcal{M}$ による学習が構文解析に対してどの程度有効であるかを最も直接的に調べるには、入力文 s について、終端記号列が s である全ての解析木 $T_1^s, T_2^s, \ldots$ のうち $\mathcal{M}$ による生成確率 $P(T_i^s)$ が最も大きい T_{max}^s を選び、T_{max}^s の精度を評価すればよい。しかし、$P(T_i^s)$ を真に最大にする T_{max}^s を求めるのは計算量が大きく困難である。そこで、本論文では PCFG による複数の解析結果を分岐 HMM でリランキングすることで高い $P(T)$ をもつ T を選ぶ実験を行い、この手法によって単純な PCFG に比べ解析精度が向上することを示す。解析候補のリランキングに関する既存研究には SVM や perceptron などの判別型のモデルによってリランキングを行うもの [10, 11, 12] がある。

2 分岐 HMM

分岐 HMM は PCFG と同様に構文木を生成する確率モデルである。但し、分岐 HMM では構文木 T を不完全な観測データと考え、T と同じ木構造をもつ、観測できない隠れた木 HT が存在すると仮定する。HT の各内部ノードのラベル $X_1, X_2, \ldots$ および各葉ノードのラベル $Y_1, Y_2, \ldots$ を隠れ変数と考え、これらをまとめて $\mathbf{X} = (X_1, X_2, \ldots)$、$\mathbf{Y} = (Y_1, Y_2, \ldots)$ と書く。また、隠れた木 HT がノードのラベルとして $\mathbf{X}$、$\mathbf{Y}$ をもつことを $HT[\mathbf{X}, \mathbf{Y}]$ と表すことにする。

図 1 の例は、観測データ (学習の訓練データ) T が、隠れた木 $HT[\mathbf{h}, \mathbf{g}]$、但し $\mathbf{h} = (h_1, h_2, h_3)$、$\mathbf{g} = (g_1, g_2, g_3)$ から生成されたことを表している。T と $HT[\mathbf{h}, \mathbf{g}]$ を合わせた完全データ $\langle T, HT[\mathbf{h}, \mathbf{g}]\rangle$ は以下のように生成される:

1. 根ノード位置の隠れ変数 X_1 が確率 $\pi(h_1)$ で値 h_1 をとる
2. 隠れ変数値を終端および非終端記号とする文脈自由規則 $h_1 \to h_2 h_3$、$h_2 \to g_1 g_2$、$h_3 \to g_3$ が、確率 $\beta(h_1 \to h_2 h_3)$、$\beta(h_2 \to g_1 g_2)$、$\beta(h_3 \to g_3)$ で適用され、HT が完成する
3. HT の各内部ノードにおいて、隠れ変数値 h_1, h_2, h_3 が非終端記号 S, NP, VP ををそれぞれ確率 $\delta(S|h_1), \delta(NP|h_2), \delta(VP|h_3)$ で生成し、葉ノードにおいて隠れ変数値 g_1, g_2, g_3 が終端記号 the/DT, cat/N, grinned/V をそれぞれ確率 $\delta(\text{the/DT}|g_1)$, $\delta(\text{cat/N}|g_2)$, $\delta(\text{grinned/V}|g_3)$ で生成する。

以上より、完全データ $\langle T, HT[\mathbf{h}, \mathbf{g}]\rangle$ の同時生起確率は

$$\begin{aligned} & P(T, HT[\mathbf{h}, \mathbf{g}]) \\ &= \pi(h_1)\ \beta(h_1 \to h_2 h_3)\ \beta(h_2 \to g_1 g_2)\ \beta(h_3 \to g_3) \\ &\times \delta(S|h_1)\ \delta(NP|h_2)\ \delta(VP|h_3) \\ &\times \delta(\text{the/DT}|g_1)\ \delta(\text{cat/N}|g_2)\ \delta(\text{grinned/V}|g_3) \end{aligned}$$

となる。不完全データである構文木 T の出現確率 $P(T)$ は、$P(T, HT[\mathbf{X}, \mathbf{Y}])$ を $HT[\mathbf{X}, \mathbf{Y}]$ に含まれる隠れ変数 $\mathbf{X} = (X_1, X_2, X_3), \mathbf{Y} = (Y_1, Y_2, Y_3)$ に関して周辺化したものになる。つまり、非終端記号位置の隠れ変数が取り得る値の集合を H_{nt}、終端記号位置の隠れ変数が取り得る値の集合を H_{t} とするとき

$$P(T) = \sum_{X_1, X_2, X_3 \in H_{\text{nt}}} \sum_{Y_1, Y_2, Y_3 \in H_{\text{t}}} P(T, HT[\mathbf{X}, \mathbf{Y}])$$

である。

訓練コーパスに現れる非終端記号よりも多い数の隠れ変数値 $h_1, h_2, \ldots$ を考えることで、$h_i \to \eta$ (η は隠れ変数値の列) という形の詳細化された CFG 規則の適用確率を学習することができる。

Treebank-PCFG に対応する分岐 HMM を単純に構築すると、観測データ T 内の一段の部分木 T' に対して、隠れた木 HT 内の T' に対応する部分木で適用された可能性のある隠れ変数の CFG 規則の数

が膨大になるため、パラメータ推定のコストが非常に大きくなる。そのため、本研究では現実的な時間で推定が行えるように、モデルにいくつかの制約を加えた。

以下この節では、最初に分岐 HMM を一般的な形で定義し、次に動的計画法と EM アルゴリズムを用いたパラメータ推定について説明し、最後にモデルに加える制約について説明する。

2.1 分岐 HMM の定義

本論文でいう分岐 HMM $\mathcal{M}$ とは、以下のような 8 つ組である:

$$\mathcal{M} = \langle N_{\rm nt}, N_{\rm t}, H_{\rm nt}, H_{\rm t}, R, \pi, \beta, \delta \rangle$$

$N_{\rm nt}$: 非終端記号 (観測変数) の集合
$N_{\rm t}$: 終端記号 (観測変数) の集合
$H_{\rm nt}$: 非終端記号を生成する隠れ変数値の集合
$H_{\rm t}$: 終端記号を生成する隠れ変数値の集合
R : 隠れ変数値を終端・非終端記号とする文脈自由規則の集合
$\pi(h)$: 根ノード位置での隠れ変数値 h の生起確率
$\beta(r)$: 規則 $r \in R$ の適用確率
$\delta(n|h)$: 隠れ変数値 h から観測変数 n が生成される確率

上の定義で、$N_{\rm nt} \cap N_{\rm t} = \emptyset$、$H_{\rm nt} \cap H_{\rm t} = \emptyset$ とし、$N = N_{\rm nt} \cup N_{\rm t}$、$H = H_{\rm nt} \cup H_{\rm t}$ とする。また、R は $h \to \eta$ (但し $h \in H_{\rm nt}$ かつ η は隠れ変数値の列、即ち $\eta \in H^+$) という形の、文脈自由形の生成規則の集合である。

各生成規則 $h \to \eta \in R$ にはその適用確率である条件付確率 $\beta(h \to \eta) = P(h \to \eta | h)$ が付与されているものとする。便宜上、R に含まれないような規則 $h \to \eta \notin R$ については $\beta(h \to \eta) = 0$ とする。$\pi(h)$ は開始記号 (構文木の根ノード) の位置での隠れ変数値 $h \in H_{\rm nt}$ の生起確率である。また、δ は通常の HMM における、いわゆる emission probability にあたるもので、ある隠れ変数値 $h \in H$ から観測変数 $n \in N$ が生成される条件付確率 $\delta(n|h) = P(n|h)$ を表す。隠れ変数値 $h_{\rm nt} \in H_{\rm nt}$ は非終端記号のみを生成し、隠れ変数値 $h_{\rm t} \in H_{\rm t}$ は終端記号のみを生成するものとする。即ち、$h_{\rm nt} \in H_{\rm nt}$、$w \in N_{\rm t}$ ならば $\delta(w|h_{\rm nt}) = 0$、また $h_{\rm t} \in H_{\rm t}$、$n \in N_{\rm nt}$ ならば $\delta(n|h_{\rm t}) = 0$ である。

β、π および δ は以下の正規化条件を満たす:

$$\sum_{h \to \eta \in R} \beta(h \to \eta) = 1 \qquad \text{for } \forall h \in H_{\rm nt}$$
$$\sum_{h \in H_{\rm nt}} \pi(h) = 1$$
$$\sum_{n \in N_{\rm nt}} \delta(n|h) = 1 \qquad \text{for } \forall h \in H_{\rm nt}$$
$$\sum_{w \in N_{\rm t}} \delta(w|h) = 1 \qquad \text{for } \forall h \in H_{\rm t}$$

次に、長さ k の文 $w_1 w_2 \ldots w_k$ に対する構文木 T の生起確率 $P(T)$ を定義する。T は根の位置に非終端記号 n_1 を持ち、その他の内部ノードの位置に非終端記号 $n_2 \ldots n_l$ を持つものとする。T を生成した隠れ変数の木を $HT[\mathbf{X}, \mathbf{Y}]$ とし、T 内の非終端記号 $n_1, \ldots, n_l$ の位置に対応する隠れ変数を $\mathbf{X} = (X_1, \ldots, X_l)$、終端記号 $w_1, \ldots, w_k$ の位置に対応する隠れ変数を $\mathbf{Y} = (Y_1, \ldots, Y_k)$ とまとめて表記する。$HT[\mathbf{X}, \mathbf{Y}]$ に含まれる、隠れ変数が作る一段の部分木を文脈自由規則の形で表したものの集合を $D_{HT} = \{X_i \to Z_1 \ldots Z_m | X_i \in \mathbf{X}, Z_1 \ldots Z_m$は X_i の娘ノードの隠れ変数列 $\}$ とする。まず、完全データ $\langle T, HT[\mathbf{X}, \mathbf{Y}] \rangle$ の同時生起確率は以下のように定義される:

$$P(T, HT[\mathbf{X}, \mathbf{Y}]) = \pi(X_1) \prod_{r \in D_{HT}} \beta(r) \times \prod_{X_i \in \mathbf{X}} \delta(n_i | X_i) \prod_{Y_i \in \mathbf{Y}} \delta(w_i | Y_i).$$

不完全データ (即ち、構文木) T の生起確率は隠れ変数 $\mathbf{X}$、$\mathbf{Y}$ について $P(T, HT[\mathbf{X}, \mathbf{Y}])$ を周辺化したものになる:

$$P(T) = \sum_{\mathbf{X} \in H_{\rm nt}^l, \mathbf{Y} \in H_{\rm t}^k} P(T, HT[\mathbf{X}, \mathbf{Y}]). \quad (1)$$

2.2 前向き・後ろ向きアルゴリズム

式 (1) の周辺化の計算の際、HMM に対する後ろ向きアルゴリズムと同様の動的計画法を用いることで効率的に $P(T)$ を求めることが可能である。以下でこれについて説明する。

木 T の各ノード (葉ノードを含む) には根ノードの番号を 1 として番号 $i \in ID(T) = \{1, 2, \ldots, m\}$ が振られているものとする。木 T を生成した隠れ

た木を HT とし、T のノード i に対応する HT のノードも同じ番号 i をもつものとする。番号 i のノードの観測記号および隠れ変数をそれぞれを n_i, Z_i とする。

ノード i、$h \in H$ に対する後ろ向き確率 $b_T^i(h)$ を以下のように再帰的に定義する:

- i が葉ノードのとき

$$b_T^i(h) = \delta(n_i|h)$$

- i が内部ノードのとき、HT 中の i の娘ノードの隠れ変数列を ζ として

$$b_T^i(h) = \delta(n_i|h) \sum_{\zeta} \beta(h \to \zeta) \prod_{Z_j \in \zeta} b_T^j(Z_j)$$

但し、右辺の ζ に関する総和は ζ に含まれる隠れ変数に対する値の割り当ての全ての組み合わせについて和を取るものとする。

根ノードに対する後ろ向き確率を求めた後、

$$P(T) = \sum_{Z_1 \in H} \pi(Z_1) b_T^1(Z_1)$$

として $P(T)$ が求まる。$P(T)$ を求める際の計算量は娘の数が最大のノード i_{max} に対する後ろ向き確率の計算量に支配され、i_{max} の娘の数を d とすると、計算量のオーダーは上の定義より $O(|H|^{d+1})$ 以下となる。

パラメータ推定アルゴリズムを説明するための準備としてノード i、$h \in H$ に対する前向き確率 $f_T^i(h)$ を以下のように定義しておく:

- i が根ノードのとき

$$f_T^i(h) = \pi(h)$$

- それ以外のとき、HT で Z_i を娘に含む一段の部分木が $Z_{i'} \to \zeta_1 Z_i \zeta_2$ であるとして

$$f_T^i(h) = \sum_{Z_{i'}, \zeta_1, \zeta_2} f_T^{i'}(Z_{i'}) \delta(n_{i'}|Z_{i'}) \times \beta(Z_{i'} \to \zeta_1 h \zeta_2) \prod_{Z_j \in \zeta_1 \cup \zeta_2} b_T^j(Z_j).$$

ただし、右辺の総和記号は $Z_{i'}$ および ζ_1, ζ_2 に含まれる隠れ変数に対する値の割り当ての全ての組み合わせについて和を取るものとする。

2.3 推定アルゴリズム

ここでは、分岐 HMM のパラメータ $\theta = (\beta, \delta, \pi)$ を推定する EM アルゴリズムについて説明する。

訓練データである構文木の集合を $\mathbf{T} = \{T_1, \ldots, T_K\}$ とし、T_i に対応する隠れ変数の木を $HT_i[\mathbf{Z}_i]$ とする。$\mathbf{Z}_i = (Z_1, \ldots)$ は HT_i に含まれる隠れ変数を全てまとめて表したものである。通常の隠れ変数モデルに対する EM アルゴリズムの導出と同様に、パラメータを $\theta = (\beta, \delta, \pi)$ から $\theta' = (\beta', \delta', \pi')$ へ更新する際の更新式は以下の量 $Q(\theta'|\theta)$ の、θ' に関する正規化条件の下での制約付最大化から導出される:

$$Q(\theta'|\theta) = \sum_{T_i \in \mathbf{T}} \sum_{\mathbf{Z}_i \in H^{|\mathbf{Z}_i|}} P_\theta(HT_i[\mathbf{Z}_i]|T_i) \log P_{\theta'}(T_i, HT_i[\mathbf{Z}_i])$$

但し、P_θ および $P_{\theta'}$ はそれぞれパラメータ θ, θ' のもとでの各々の確率を表す。$Q(\theta'|\theta)$ を θ' について偏微分したものをゼロとおき、前向き、後ろ向き確率を用いて整理すると図 2 の更新式が得られる。

2.4 モデルの制約

EM アルゴリズムを用いてパラメータ推定を行う際、E-step で条件付確率 $P_\theta(HT_i[\mathbf{Z}_i]|T_i)$ を求めるときに $P(T_i)$ の計算が必要であり、これに要する計算量は上述のようにオーダー $O(|H|^{d+1})$ である。Penn Treebank のように、比較的平坦な構文木を多く含むコーパス (d が大) を用いて学習を行う場合、学習に要する時間コストは $|H|$ を大きく設定すると非常に大きくなる。

この問題を避けるため、本研究では上で定義した分岐 HMM に 2 つの制約を加え、現実的な時間での学習を可能にした。一つ目の制約は生成する構文木を 2 分木に限ることであり、これによって $d \leq 2$ に制限することができる。二つ目の制約は、各 $h \in H$ について h が生成する観測記号をあらかじめ特定の $n \in N$ に限定することである。つまり $\delta(n|h) = 1$ または $\delta(n|h) = 0$ のどちらか。これによって T の部分木 $n_i \to n_j n_k$ に対応する $HT[\mathbf{Z}]$ の部分木 $Z_i \to Z_j Z_k$ で適用された可能性のある隠れ変数の生成規則の数を大幅に減らすことができる。以下、これら 2 つの制約についてより具体的に説明する。

$$\beta'(h{\to}\eta) = \frac{\sum_{T_i\in\mathbf{T}}\sum_{j\in \mathrm{App}(T_i,h\to\eta)} f^j_{T_i}(h)\,\beta(h{\to}\eta)\prod_{k=1}^{l} b^{\mathrm{dtr}(T_i,j,k)}_{T_i}(h^k)}{\sum_{T_i\in\mathbf{T}}\sum_{j\in ID(T_i)} f^j_{T_i}(h)b^j_{T_i}(h)}$$

(但し、$h{\to}\eta \in R,\ \ \eta = h^1h^2\ldots h^l$)

$$\delta'(n|h) = \frac{\sum_{T_i\in\mathbf{T}}\sum_{j\in \mathrm{Obs}(T_i,n)} f^j_{T_i}(h)b^j_{T_i}(h)}{\sum_{T_i\in\mathbf{T}}\sum_{j\in ID(T_i)} f^j_{T_i}(h)b^j_{T_i}(h)}$$

$$\pi'(h) = \frac{1}{|\mathbf{T}|}\sum_{T_i\in\mathbf{T}}\pi(h)b^1_{T_i}(h)$$

$\mathrm{App}(T_i, h\to\eta)$: $HT_i[\mathbf{Z}_i]$ 内で $h\to\eta\in R$ が適用された可能性のあるノードの集合

$\mathrm{Obs}(T_i, n)$: T_i内で観測記号 n を持つノードの集合

$\mathrm{dtr}(T_i, j, k)$: T_iのノード j の k 番目の娘のノード番号

図 2: パラメータ更新則

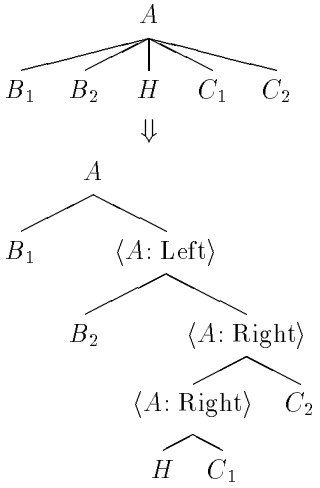

図 3: 主辞を中心にした部分木の 2 分木化 (H : 主辞)

本研究では、訓練およびテスト用コーパスとして Penn Treebank を用い、コーパス中の構文木を以下の手順で 2 分木に変形した。まず、Collins の主辞規則 (head rule)[2] を用いて構文木内の各部分木について主辞である娘ノードを決定し、主辞娘ノードを中心にして図 3 の例のように 2 分木に変形した。分岐 HMM が生成する隠れ変数の木は、図 3 のような形の部分 2 分木および unary transition を組み合わせた構文木の各ノードに隠れ変数が一つずつ配置されたものになる。構文木を 2 分木に変形する方法は他にも考えられるが、変形する方法の違いが分岐 HMM を用いた構文解析の精度に与える影響についてはまだ詳細な比較は行っていない。

モデルに加える二つ目の制約として、非終端記号を生成する隠れ変数値の集合 H_{nt} を

$$H_{\mathrm{nt}} = \bigcup_{\nu\in N_{\mathrm{nt}}} H_{\mathrm{nt}}(\nu),\ \ \nu\neq\nu' \Rightarrow H_{\mathrm{nt}}(\nu)\cap H_{\mathrm{nt}}(\nu') = \emptyset$$

と排他な部分集合に分割し、$h\in H_{\mathrm{nt}}(\nu) \Rightarrow \delta(\nu|h) = 1$ という制約を加えた。この制約には、例えば図 4 のような、対応する n-分木を持たない 2 分木の生成確率を 0 にするという意味もある。また、“the/DT” のように単語 w と PoS タグ τ を組合わせたものを一つの終端記号として扱い、終端記号を生成する隠れ変数値の集合 H_{t} を

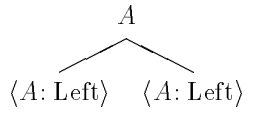

図 4: 対応する n-分木をもたない 2 分木

$$H_{\mathrm{t}} = \bigcup_{\tau\in \mathrm{PoS}} H_{\mathrm{t}}(\tau),\ \ \tau\neq\tau' \Rightarrow H_{\mathrm{t}}(\tau)\cap H_{\mathrm{t}}(\tau') = \emptyset$$

と排他に分割し (PoS は PoS タグの集合)、

$$\textstyle\sum_{w\in W}\delta(w/\tau|h) = 1 \ \text{ for } \forall h\in H_{\mathrm{t}}(\tau)$$
$$h\in H_{\mathrm{t}}(\tau), \tau\neq\tau' \Rightarrow \delta(w/\tau'|h) = 0 \ \text{ for } \forall w\in W$$

という制約を加えた (W は単語の集合)。

以上の二つの制約をモデルに加えると、$H_{\mathrm{nt}}(\nu)$、$H_{\mathrm{t}}(\tau)$ のサイズの最大値を M とするとき $P(T)$ を求める際の計算量は $O(M^3)$ 以下となる。

3 分岐 HMM による構文木のリランキング

分岐 HMM $\mathcal{M}$ を用い、以下のような手順で構文解析を行うことが原理的には可能である。

1. 隠れ変数の文脈自由文法 R と emission probability δ から、$\mathcal{M}$ が生成しうる、観測記号の 1 段の部分木を全て求め、これを文脈自由文法 G とする。

2. 入力文 s に対して可能な、G による全ての解析 $\mathbf{T}_s = \{T_1, T_2, \ldots\}$ の中から、分岐 HMM $\mathcal{M}$ の下での生起確率 $P_{\mathcal{M}}(T)$ が最も大きいもの T_{best} を求め、T_{best} を s の解析結果として出力する。

しかし現実的には、ある程度以上の長さの文に対しては $\mathbf{T}_s$ の要素 T_i を列挙し、それぞれについて $P_{\mathcal{M}}(T_i)$ を計算するのは計算量が大きく困難である。また、観測データである構文木 T が与えられた場合は上で示した前向き・後ろ向きアルゴリズムを用いて効率的に $P_{\mathcal{M}}(T)$ を計算できるが、観測データの一部である単語列 s のみが与えられたときに s を葉ノードとする最適な木構造を効率的に見つけるアルゴリズムは筆者らが知る限り現在のところ存在しない。

そこで、本論文では、PCFG による複数の解析結果を分岐 HMM を用いてリランキングすることで高い生起確率をもつ構文木を探す方法を提案する。具体的には以下のような方法で実験を行った:

Step 1 入力文 s に対して適当な PCFG G' を用いたビームサーチを行い，得られた解析結果のうち、G' の下での生起確率が最も大きい N 個の[1]解析結果 $\mathbf{T}'_s = \{T'_1, \ldots, T'_N\} \subset \mathbf{T_s}$ を得る。

Step 2 $\mathbf{T}'_s = \{T'_i\}$ のうち、$\mathcal{M}$ の下での生起確率 $P_{\mathcal{M}}(T'_i)$ が最も大きい構文木を選び、それを s の解析結果として出力する。

4 実験

この節では、分岐 HMM の構文解析に対する有効性を評価するために、前節で述べたリランキングによる方法で構文解析の実験を行った結果について述べる。

実験データとして Penn WSJ Treebank を用い、section 2-21 (約 4 万文) を訓練に、section 22 (1700 文) をテストに用いた。具体的には、section 2-21 を Step 1 で用いる PCFG の訓練データとして使用し、分岐 HMM の学習には訓練データとして section 2-20 を使用し、section 21 を EM アルゴリズムの繰り返しを止めるタイミングを決めるために用いた[2]。

Step 1 で使用した PCFG は、各非終端記号について親ノードの非終端記号を付加し、文法規則のマルコフ化 [2] を行ったものである[3]。また、Step 1 への入力として単語列のみを与えた場合 (実験 1) と正解の PoS タグ列を与えた場合 (実験 2) の 2 通りの実験を行った。実験 1、実験 2 のいずれの場合もビーム幅 10,000 のビームサーチを行い[4]、各入力文に対し上位 1,000 個までの解析結果を Step 2 への入力とした。Step 2 で使用する分岐 HMM は、図 3 のように 2 分木化した後の各非終端記号 ν および PoS タグ τ について $|H_{\mathrm{nt}}(\nu)| = |H_{\mathrm{t}}(\tau)| = k$ としたモデル $\mathcal{M}_k$、$k = 4, 8, 16$ の 3 つについて実験を行った。

Step 2 で $\mathcal{M}_4$、$\mathcal{M}_8$、$\mathcal{M}_{16}$ をそれぞれ用いた時に出力された解析結果について labeled recall (LR) および labeled precision (LP) を測ったものを表 1、表 2 の「$\mathcal{M}_k$」の行にまとめる。ただし、Step 1 で解析候補がひとつも得られなかった文 (実験 1 で 33 文、実験 2 で 17 文) については評価から除いてある。参考の為、Step 1 の出力 $\mathbf{T}'_s$ のうち、PCFG の下での生成確率が最も大きいものを選んだ場合 (「PCFG」の行) および $\mathbf{T}'_s$ のうち F_1 スコアが最も大きいものを常に選んだ場合 (「Oracle」の行) の結果も併せて載せておく。

また、表 3 に実験に要した計算時間についてまとめる。Step 1 の PCFG による構文解析に要した時間は実験 1 で約 14 時間、実験 2 で約 18 時間であった。また、Step 1 で 1 文あたり得られた構文木の数の平均値は実験 1 で約 630、実験 2 で約 560 であった。

実験結果から、観測記号あたりの隠れ変数の数 k を $4 \to 8 \to 16$ と増やし、より詳細な CFG 規則を分岐 HMM に学習させることで、リランキングの精度が向上していることが分かる。オーバーフィッティングの問題を除けば、k を増やすことでさらに

[1] ビームサーチを用いるため、実際には N 個以下の解析結果しか得られないこともある

[2] section 21 に対する尤度の増加率がある閾値以下になったときに繰り返しを止めた (アーリーストッピング)。

[3] 正確には、Klein ら [5] が $v = 2, h = 1$ のマルコフ化と呼んでいる方法を用いた。

[4] チャートの各セルで内側確率の大きい上位 10,000 個の部分解析結果を残した。

表 1: 単語列を入力とした場合の結果 (section 22)

	40 語以下		全ての文	
	LP	LR	LP	LR
PCFG	79.12	77.93	78.54	77.40
Oracle	92.55	91.85	91.49	90.81
$\mathcal{M}_4$	81.21	81.44	80.38	80.58
$\mathcal{M}_8$	82.79	82.83	81.86	81.93
$\mathcal{M}_{16}$	83.78	83.82	82.89	82.92

表 2: 正解 PoS タグ列を入力とした場合の結果 (section 22)

	40 語以下		全ての文	
	LP	LR	LP	LR
PCFG	80.01	78.55	79.10	77.69
Oracle	94.91	94.11	93.53	92.75
$\mathcal{M}_4$	83.44	83.45	82.42	82.37
$\mathcal{M}_8$	84.74	84.76	83.59	83.59
$\mathcal{M}_{16}$	86.28	86.14	84.92	84.75

表 3: 実験に要した計算時間

		Step 2	
	$\mathcal{M}_k$ の学習時間	実験 1	実験 2
$\mathcal{M}_4$	約 0.5 時間	約 12 分	約 10 分
$\mathcal{M}_8$	約 4 時間	約 50 分	約 42 分
$\mathcal{M}_{16}$	約 36 時間	約 345 分	約 295 分

精度が向上することが期待できそうだが、表 3 から分かるように $\mathcal{M}_k$ の学習およびリランキングに必要な時間はほぼ k^3 に比例しており、現在の学習アルゴリズムで現実的に学習を行える k のサイズには限界がある。これに関しては、近似を入れたアルゴリズムを利用して学習およびリランキングを高速化するなどの改良が考えられる。

5 まとめ

本論文では、構文木を不完全な観測データと考え、細分化された終端・非終端記号を表す隠れ変数をノードとする隠れた木から構文木を生成する分岐 HMM という確率モデルを定義し、パラメータ推定のための EM アルゴリズムを導出した。また、PCFG を用いて得た複数の解析候補を分岐 HMM によってリランキングする実験を行い，分岐 HMM の学習によって自動的に得た詳細化された CFG 規則が構文解析の精度を向上させることを確かめた。

参考文献

[1] Eugene Charniak. Tree-bank grammars. In *AAAI/IAAI, Vol. 2*, pp. 1031–1036, 1996.

[2] Michael Collins. *Head-Driven Statistical Models for Natural Language Parsing*. PhD thesis, University of Pennsylvania, 1999.

[3] Eugene Charniak. A maximum-entropy-inspired parser. Technical Report CS-99-12, 1999.

[4] Mark Johnson. Pcfg models of linguistic tree representations. *Computational Linguistics*, Vol. 24, No. 4, pp. 613–632, 1998.

[5] Dan Klein and Christopher D. Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, 2003.

[6] Shai Fine, Yoram Singer, and Naftali Tishby. The hierarchical hidden markov model: Analysis and applications. *Machine Learning*, Vol. 32, No. 1, pp. 41–62, 1998.

[7] K. Lari and S. J. Young. The estimation of stochastic context-free grammars using the inside-outside algorithm. *Computer Speech and Language*, Vol. 4, pp. 35–56, 1990.

[8] Fernando Pereira and Yves Schabes. Inside-outside reestimation from partially bracketed corpora. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*, pp. 128–135, 1992.

[9] 宇津呂武仁, 小玉修司, 松本裕治. 非終端記号のエントロピーを用いた文脈自由文法の一般化・特殊化. 人工知能学会第 10 回全国大会論文集, pp. 327–330, 1996.

[10] Michael Collins. Discriminative reranking for natural language parsing. In *Proceedings of the 17th International Conference on Machine Learning*, 2000.

[11] Michael Collins and Nigel Duffy. New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 263–270, 2002.

[12] Libin Shen, Anoop Sarkar, and Aravind Joshi. Using LTAG based features in parse reranking. In Michael Collins and Mark Steedman, editors, *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pp. 89–96, 2003.