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

2007－NL－179 (4)
2007－SLP－ 66 (4)
2007／5／24

ギブスサンプリングを用いた係り受け解析

中川 哲治
沖電気工業株式会社
nakagawa378@oki.com

本稿では，ギブスサンプリングを用いた係り受け解析手法を提案する．既存の解析手法ではしばしば変数間に独立性を仮定しており，利用可能な素性が限られているという問題があった．提案手法では，依存構造木全体をモデル化する確率分布を考えることで，依存構造木中の兄弟ノードに関する関係や，子ノードと祖父母ノードに関する関係などの，文中の任意の素性を利用することができる．複数のコーパスで実験を行った結果，提案手法は既存手法と比較して同程度以上の解析精度を持つことを確認した．

Dependency Parsing using Gibbs Sampling

Tetsuji Nakagawa
Oki Electric Industry Co., Ltd.
nakagawa378@oki.com

In this paper, we present a method for dependency parsing with Gibbs sampling. Existing methods for dependency parsing often assume independence among variables, and have limitations in available features. Our method uses a probabilistic model of a whole dependency tree, and allows us to use arbitrary features in a dependency tree, which include relations between sibling nodes and relations between a child and its grandparent nodes. Experimental results on multiple corpora showed that the performance of our method was competitive with other state-of-the-art methods.

1 背景

これまでに，様々な統計的係り受け解析の方法が研究されている．内元らは，最大エントロピー法を用いた日本語の係り受け解析手法を提案した [18]．McDonald らは交差の無い係り受け解析 (projective dependency parsing) のための，マージン最大化に基づくオンライン学習アルゴリズムを用いた方法を提案し [7]，さらにその手法を交差の有る係り受け解析 (non-projective dependency parsing) へも適用した [8]．しかしながらこれらの手法において，文中の各文節や単語の係り先は互いに独立であると仮定されており，利用可能な素性は限られている．

Collins らは構文木中の任意の素性を使用して句構造解析を行うために，リランキングに基づく手法を提案した [4]．係り受け解析においても，そのような大域的な素性を利用する方法がいくつか提案されている．McDonald らは二次の素性 (second-order feature) を利用することができる方法を提案し [9]，Riedel らは言語学的な制約を利用することができる方法を提案したが [12]，そのような大域的な素性を用いることにより解析精度を大きく向上させられることが確認されている．本稿ではこれらのアプローチとは異なった，ギブスサンプリングを用いて大域的な素性を利用する方法を提案する．なお，提案手法は言語には依存せず，依存構造木が交差を含む場合も含まない場合も適用可能である．

以下，2 節ではギブスサンプリングを用いた係り受け解析手法と使用した素性について説明し，3 節では実験結果を報告する．4 節では関連研究について議論し，5 節で結論を述べる．

2 手法

係り受け解析は，与えられた文に対してその依存構造 (係り受け関係) を同定するタスクである [5, 10]．依存構造は，図 1 の例で示されるような依存構造木により表現することができる．この図において，グラフ中の各ノードはトークン[*1]を表し，トークン間の依存関係は係り先 (head) から係り元 (dependent) への矢印により示されている．各トークンはただ一つの係り先を持つが，依存構造木中のルートノードであるトークンは例外であり，そのようなトークンは係り先を持たない．依存構造木は多くの場合交差するエッジを持たないが，チェコ語などの語順が比較的自由な言語では依存構造木中に交差するエッジを持つことがある．以下の節では，係り受け解析のための確率モデルと，それを用いた解析方法，パラメータ推定方法，および使用した素性について説明する．

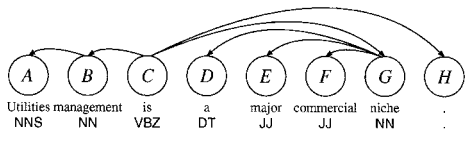

図 1 依存構造木の例

2.1 確率モデル

以下の説明では，入力文を $\mathbf{w}$ で表すことにし，$|\mathbf{w}|$ は文中のトークン数を，w_t は文中の t 番目のトークンを表すことにする:

$$\mathbf{w} = \{w_1, \cdots, w_{|\mathbf{w}|}\}.$$

先頭から t 番目のトークン w_t の係り先のインデックス (何番目のトークンに係るか) を h_t で表すことにし，$\mathbf{h}$ は h_t の集合を表すものとする．文中のルートノードは係り先を持たないが，そのようなルートノードの係り先のインデックスは 0 として考える:

$$\mathbf{h} = \{h_1, \cdots, h_{|\mathbf{w}|}\},$$
$$h_t \in \{0, 1, \cdots, |\mathbf{w}|\} \quad (t = 1, \cdots, |\mathbf{w}|).$$

この場合，入力文 $\mathbf{w}$ に対する係り受け解析は，その文中のトークンに対する係り先の集合 $\mathbf{h}$ を決定するタスクと考えることができる．

本研究では，入力文 $\mathbf{w}$ に対する依存構造 $\mathbf{h}$ の条件付き確率分布を，exponential model を用いて次のように定義する:

$$P_{\Lambda,\mathrm{M}}(\mathbf{h}|\mathbf{w}) = \frac{1}{Z_{\Lambda,\mathrm{M}}(\mathbf{w})} Q_{\mathrm{M}}(\mathbf{h}|\mathbf{w}) \exp\left\{\sum_{k=1}^{K} \lambda_k f_k(\mathbf{w}, \mathbf{h})\right\}, \quad (1)$$

$$Z_{\Lambda,\mathrm{M}}(\mathbf{w}) = \sum_{\mathbf{h}' \in \mathcal{H}(\mathbf{w})} Q_{\mathrm{M}}(\mathbf{h}'|\mathbf{w}) \exp\left\{\sum_{k=1}^{K} \lambda_k f_k(\mathbf{w}, \mathbf{h}')\right\}. \quad (2)$$

[*1] 本稿では，係り受け解析における基本的な単位を「トークン」と呼ぶことにする．日本語では文節が基本的な単位となることが多いが，英語等では単語が基本的な単位として使用されることが多い．なお，各トークンには品詞が付与されているものと仮定する．

ここで，$Z_{\Lambda,\mathrm{M}}(\mathbf{w})$ は正規化定数，$Q_{\mathrm{M}}(\mathbf{h}|\mathbf{w})$ は初期分布，$f_k(\mathbf{w},\mathbf{h})$ は k 番目の素性関数，K は素性関数の数，λ_k は k 番目の素性の重み，である．$\mathcal{H}(\mathbf{w})$ は，入力文 $\mathbf{w}$ に対する全ての可能な係り先の組み合わせを含む集合である．Rosenfeld らは，文中の任意の素性を利用することができる whole-sentence language model を構築するために，これと似たモデルを使用した [13]．この確率モデルは最大エントロピーモデルを一般化したものであり，初期分布を一様分布とした場合に最大エントロピーモデルと等しくなる．ある訓練データに対するこの exponential model の最尤解は，素性の期待値がその訓練データにおける経験的期待値と等しいという制約のもとで，Kullback-Leibler 距離が初期分布に最も近いような分布と等しいことが知られている [11, 13]．式 (1) 中の確率分布 $P_{\Lambda,\mathrm{M}}(\mathbf{h}|\mathbf{w})$ は，入力文に対する各トークンの係り先の同時分布であり，この分布は文中の全ての係り受け関係をモデル化するため，「文単位のモデル」あるいは「大域的モデル」と呼ぶことにする．素性関数 $f_k(\mathbf{w},\mathbf{h})$ は係り先の集合 $\mathbf{h}$ を持つ文全体 $\mathbf{w}$ に関して定義され，各トークンの係り先に対して独立性を仮定せずに文中の任意の情報を利用することができるため，$f_k(\mathbf{w},\mathbf{h})$ を「文単位の素性」または「大域的素性」と呼ぶことにする．この文単位の素性については，2.5 節にて詳しく説明する．

初期分布 $Q_{\mathrm{M}}(\mathbf{h}|\mathbf{w})$ は，各トークンの係り先の分布 $q_{\mathrm{M}}(h|\mathbf{w},t)$ の積として次のように定義する:

$$Q_{\mathrm{M}}(\mathbf{h}|\mathbf{w}) = \prod_{t=1}^{|\mathbf{w}|} q_{\mathrm{M}}(h_t|\mathbf{w},t). \tag{3}$$

ここで $q_{\mathrm{M}}(h|\mathbf{w},t)$ は，文 $\mathbf{w}$ が与えられた場合に t 番目のトークンの係り先が h である確率であり，次のように最大エントロピーモデルを用いて定義する:

$$q_{\mathrm{M}}(h|\mathbf{w},t) = \frac{1}{Y_{\mathrm{M}}(\mathbf{w},t)} \exp\left\{\sum_{l=1}^{L} \mu_l g_l(\mathbf{w},t,h)\right\}, \tag{4}$$

$$Y_{\mathrm{M}}(\mathbf{w},t) = \sum_{h'=0}^{|\mathbf{w}|} \exp\left\{\sum_{l=1}^{L} \mu_l g_l(\mathbf{w},t,h')\right\}. \tag{5}$$

ここで，$Y_{\mathrm{M}}(\mathbf{w},t)$ は正規化定数であり，$g_l(\mathbf{w},t,h)$ は l 番目の素性関数であり，L は素性関数の数であり，μ_l は l 番目の素性の重みである．$q_{\mathrm{M}}(h|\mathbf{w},t)$ は単一のトークンに関する係り先のモデルであり，他のトークンとは独立に計算されるため，$q_{\mathrm{M}}(h|\mathbf{w},t)$ を「トークン単位のモデル」または「局所的モデル」と呼び，$g_l(\mathbf{w},t,h)$ を「トークン単位の素性」または「局所的素性」と呼ぶことにする．トークン単位の素性については，2.4 節にて詳しく説明する．

2.2 ギブスサンプリングを用いた解析

文 $\mathbf{w}$ と，文単位のモデルのパラメータ $\Lambda = \{\lambda_1, \cdots, \lambda_K\}$ と，トークン単位のモデルのパラメータ $\mathrm{M} = \{\mu_1, \cdots, \mu_L\}$ が与えられた場合に，最適な解 $\hat{\mathbf{h}}$ を求めることを考える．$P_{\Lambda,\mathrm{M}}(\hat{\mathbf{h}}|\mathbf{w})$ の値を最大化するような係り先の集合 $\hat{\mathbf{h}}$ を求めようとした場合，可能な解候補の数が非常に多いため，厳密な計算を行うのは困難である．文単位のモデルでは文単位の任意の素性を扱うことを考えており，変数間の独立性は仮定できないため，効率的な動的計画法を用いることはできない．そこで，ギブスサンプリングを用いて近似的に解を求めることにする．

ギブスサンプリングはマルコフ連鎖モンテカルロ法の一種であり，複雑な依存関係を持つ高次元の確率分布から，効率的にサンプルを生成することができる手法である [16]．図 2 にそのアルゴリズムを示す．このアルゴリズムでは，まず最初に初期状態 $\mathbf{h}^{(0)}$ を決める．そして，ある 1 つの確率変数について，それ以外の確率変数を全て固定した条件付き確率からのサンプリングを行い値を更新するという手続きを繰り返して，新しいサンプルを生成していく．ギブスサンプリングにより生成されるサンプルの分布は，もとの確率の定常分布へ収束することが知られている．実験では，確率分布 $Q_{\mathrm{M}}(\mathbf{h}|\mathbf{w})$ を最大化する係り先の集合を初期状態 $\mathbf{h}^{(0)}$ とした．ギブスサンプリング自体は確率分布からサンプルを生成するための手法であり，最適化を行うための手法ではない．そこで最適な解を求めるために，最大全域木 (maximum spanning tree; MST) の探索手法を利用することにする．

```
1 h^(0) を初期化する
2 for r := 1 to R
3   for t := 1 to |w|
4     h_t^(r) ~ P(h_t | w, h_1^(r), ..., h_{t-1}^(r), h_{t+1}^(r-1), ..., h_{|w|}^(r-1))
```

図 2　ギブスサンプリング

McDonald らは，交差の無い係り受け解析の問題を最大全域木問題として解いた [7]．彼らは，文中の各トークン間の係り受け関係を，各トークンをノードとするグラフ中の有向エッジとみなして，各エッジに対するスコアを定義し，このグラフ中の最大全域木 (グラフ中の全てのノードを含む木で，エッジのスコアの和が最大であるもの) を探索することにより最適な依存構造木を求めた．彼らは交差の無い最大全域木を探索するために，Eisner のアルゴリズムを利用した．この手法は，交差の有る最大全域木を探索するために Chu-Liu-Edmonds (CLE) のアルゴリズムを使用する方法 [8] へと拡張された．ここでは，我々もこの最大全域木を用いた枠組みを利用することにする．親ノード (係り先) i から子ノード (係り元) j へのエッジのスコアを $s(i,j)$ で表すことにする．そして，$s(i,j)$ を次のように定義する:

$$s(i,j) = \log P_j(i|\mathbf{w}). \tag{6}$$

ここで $P_t(h|\mathbf{w})$ は，文 $\mathbf{w}$ が与えられた場合における t 番目のトークンの係り先の周辺確率である．上式において周辺分布の対数をスコアとしているのは，最大全域木のアルゴリズムはエッジのスコアの和を最大化するが，ここでは周辺分布の積を最大化したいからである．この周辺確率は，ギブスサンプリングにより確率分布 $P_{\Lambda,\mathrm{M}}(\mathbf{h}|\mathbf{w})$ から得た R 個のサンプル $\{\mathbf{h}^{(1)}, \cdots, \mathbf{h}^{(R)}\}$ を利用して，次のように近似的に計算する:

$$P_t(h|\mathbf{w}) = \sum_{\substack{h_1,\cdots,h_{t-1},h_{t+1},\cdots,h_{|\mathbf{w}|} \\ h_t=h}} P_{\Lambda,\mathrm{M}}(\mathbf{h}|\mathbf{w}),$$

$$= \sum_{\mathbf{h}} P_{\Lambda,\mathrm{M}}(\mathbf{h}|\mathbf{w})\delta(h,h_t) \simeq \frac{1}{R}\sum_{r=1}^{R} \delta(h,h_t^{(r)}). \tag{7}$$

ここで，$\delta(i,j)$ はクロネッカーのデルタである．なお，$P_t(h|\mathbf{w})$ を最大化するように各トークンの係り先を選ぶような方法も考えられるが，そのようにした場合，得られる依存構造グラフ中にはサイクルが含まれる可能性がある．以上のようにして定義されたスコアと Eisner アルゴリズムを用いることで，最適な交差の無い依存構造木を求めることができる．また，CLE アルゴリズムを用いることで，最適な交差の有る依存構造木を求めることができる．提案手法では，最適な依存構造木を選ぶために各エッジに対して互いに独立に定義されるスコア $s(i,j)$ を使用するが，このスコアは文全体を考慮してギブスサンプリングを用いて計算されるものであり，文全体の情報を利用することができる．

ギブスサンプリングでは多数のサンプルを繰り返し生成する必要があるが，サンプリングを行う度に一つのトークンに対する全ての係り先の候補について確率を計算するため，計算量が比較的大きい．そこで，計算時間を削減するた

めにいくつかの手法を用いた．まず，$c(pos_d, pos_h, dir)$ という関数を考える．この関数は，訓練データ中で品詞 pos_d を持ち，その係り先の品詞が pos_h であり，係り先から係り元への方向 (左または右) が dir であるようなトークンの数である．ギブスサンプリングにおいて，あるトークンに対する係り先の候補の確率を計算する際に，もしその係り元と係り先に対するこの関数の値が 0 であるならば，そのような係り先は無視することにする．さらに，トークン単位のモデルにより計算された確率の値がある閾値 θ よりも小さい場合には，その係り先の候補は無視することにする．なぜならば，もしトークン単位の素性のみを用いて計算した確率が十分小さければ，文単位の素性を用いて計算した確率も無視できる程度に小さいと考えられるからである．実験では，θ の値は 0.5% とした．ギブスサンプリングにおける確率の計算 (図 2 の 4 行目) において，文中のあらゆる大域的な素性を生成する必要はない．現在考慮している変数 h_t に関する大域的な素性しか確率分布に影響を及ぼさないので，それらの素性のみを生成すればよい．

2.3 モデルパラメータ推定

N 個の事例からなる学習データ $\{\langle \mathbf{w}^1, \mathbf{h}^1\rangle, \cdots, \langle \mathbf{w}^N, \mathbf{h}^N\rangle\}$ が与えられた場合に，モデルのパラメータを推定することを考える．

初めに，トークン単位のモデルのパラメータ $\mathrm{M} = \{\mu_1, \cdots, \mu_L\}$ を推定する方法について説明する．ここでは，Gaussian prior を用いて最大事後確率 (maximum a posteriori; MAP) 推定を行う．次のような目的関数 $\mathcal{M}$ を定義し，$\mathcal{M}$ の値を最大化する最適解を見つけることにする:

$$\begin{aligned}\mathcal{M} &= \log \prod_{n=1}^{N} Q_{\mathrm{M}}(\mathbf{h}^n|\mathbf{w}^n) - \frac{1}{2\sigma^2}\sum_{l=1}^{L}\mu_l^2, \\ &= \sum_{n=1}^{N}\sum_{t=1}^{|\mathbf{w}^n|}\left[-\log Y_{\mathrm{M}}(\mathbf{w}^n, t) + \sum_{l=1}^{L}\mu_l g_l(\mathbf{w}^n, t, h_t^n)\right] \\ &\quad -\frac{1}{2\sigma^2}\sum_{l=1}^{L}\mu_l^2. \end{aligned} \tag{8}$$

ここで，σ は Gaussian prior に関するハイパーパラメータである．目的関数の偏微分は次のようになる:

$$\begin{aligned}&\frac{\partial \mathcal{M}}{\partial \mu_l} \\ &= \sum_{n=1}^{N}\sum_{t=1}^{|\mathbf{w}^n|}\left[-\frac{\partial}{\partial \mu_l}\log Y_{\mathrm{M}}(\mathbf{w}^n, t) + g_l(\mathbf{w}^n, t, h_t^n)\right] - \frac{1}{\sigma^2}\mu_l, \\ &= \sum_{n=1}^{N}\sum_{t=1}^{|\mathbf{w}^n|}\left[-\sum_{h'=0}^{|\mathbf{w}^n|} q_{\mathrm{M}}(h'|\mathbf{w}^n, t) g_l(\mathbf{w}^n, t, h') + g_l(\mathbf{w}^n, t, h_t^n)\right] \\ &\quad -\frac{1}{\sigma^2}\mu_l. \end{aligned} \tag{9}$$

最適なパラメータ M は，上述の $\mathcal{M}$ と $\partial\mathcal{M}/\partial\mu_l$ を用いることにより，L-BFGS 法などの準ニュートン法により求められる．

次に，既にトークン単位のモデルのパラメータ M が得られているという条件のもとで，文単位のモデルのパラメータ $\Lambda = \{\lambda_1, \cdots, \lambda_K\}$ を推定する方法を説明する．MAP 推定を用いることにし，目的関数 $\mathcal{L}$ とその偏微分 $\partial\mathcal{L}/\partial\lambda_k$ を次のように定義する:

$$\begin{aligned}\mathcal{L} &= \log \prod_{n=1}^{N} P_{\Lambda,\mathrm{M}}(\mathbf{h}^n|\mathbf{w}^n) - \frac{1}{2\sigma'^2}\sum_{k=1}^{K}\lambda_k^2, \\ &= \sum_{n=1}^{N}\left[-\log Z_{\Lambda,\mathrm{M}}(\mathbf{w}^n) + \sum_{k=1}^{K}\lambda_k f_k(\mathbf{w}^n, \mathbf{h}^n)\right] \\ &\quad -\frac{1}{2\sigma'^2}\sum_{k=1}^{K}\lambda_k^2 + \sum_{n=1}^{N}\log Q_{\mathrm{M}}(\mathbf{h}^n|\mathbf{w}^n), \end{aligned} \tag{10}$$

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \lambda_k} &= \sum_{n=1}^{N}\left[-\frac{\partial}{\partial\lambda_k}\log Z_{\Lambda,\mathrm{M}}(\mathbf{w}^n) + f_k(\mathbf{w}^n, \mathbf{h}^n)\right] - \frac{1}{\sigma'^2}\lambda_k, \\ &= \sum_{n=1}^{N}\left[-\sum_{\mathbf{h}'\in\mathcal{H}(\mathbf{w}^n)} P_{\Lambda,\mathrm{M}}(\mathbf{h}'|\mathbf{w}^n) f_k(\mathbf{w}^n, \mathbf{h}') + f_k(\mathbf{w}^n, \mathbf{h}^n)\right] \\ &\quad -\frac{1}{\sigma'^2}\lambda_k. \end{aligned} \tag{11}$$

ここで，σ' は Gaussian prior に関するハイパーパラメータである．最適なパラメータ Λ は準ニュートン法により求めることができるが，$\mathcal{L}$ と $\partial\mathcal{L}/\partial\lambda_k$ の計算には，全ての可能な係り先の組み合わせについての和が含まれるため，厳密な計算を行うのは困難である．そこで，無作為に生成されたサンプルを用いて近似的にこれらの値を計算することにする [1, 11, 13]．訓練データ中の n 番目の文 $\mathbf{w}^n$ に対して，S 個のサンプル $\{\mathbf{h}^{n(1)}, \cdots, \mathbf{h}^{n(S)}\}$ が確率分布 $Q_{\mathrm{M}}(\mathbf{h}|\mathbf{w})$ から生成されるとする．これらのサンプルを用いると，係り先の組み合わせについての和を含む項は次のように近似的に計算できる:

$$\begin{aligned}&\log Z_{\Lambda,\mathrm{M}}(\mathbf{w}^n) \\ &= \log \sum_{\mathbf{h}'\in\mathcal{H}(\mathbf{w}^n)} Q_{\mathrm{M}}(\mathbf{h}'|\mathbf{w}^n)\exp\left\{\sum_{k=1}^{K}\lambda_k f_k(\mathbf{w}^n, \mathbf{h}')\right\}, \\ &\simeq \log \frac{1}{S}\sum_{s=1}^{S}\exp\left\{\sum_{k=1}^{K}\lambda_k f_k(\mathbf{w}^n, \mathbf{h}^{n(s)})\right\}, \end{aligned} \tag{12}$$

$$\begin{aligned}&\sum_{\mathbf{h}'\in\mathcal{H}(\mathbf{w}^n)} P_{\Lambda,\mathrm{M}}(\mathbf{h}'|\mathbf{w}^n) f_k(\mathbf{w}^n, \mathbf{h}') \\ &= \sum_{\mathbf{h}'\in\mathcal{H}(\mathbf{w}^n)} Q_{\mathrm{M}}(\mathbf{h}'|\mathbf{w}^n)\frac{f_k(\mathbf{w}^n, \mathbf{h}')}{Z_{\Lambda,\mathrm{M}}(\mathbf{w}^n)}\exp\left\{\sum_{k'=1}^{K}\lambda_{k'} f_{k'}(\mathbf{w}^n, \mathbf{h}')\right\}, \\ &\simeq \frac{1}{S}\sum_{s=1}^{S}\frac{f_k(\mathbf{w}^n, \mathbf{h}^{n(s)})}{Z_{\Lambda,\mathrm{M}}(\mathbf{w}^n)}\exp\left\{\sum_{k'=1}^{K}\lambda_{k'} f_{k'}(\mathbf{w}^n, \mathbf{h}^{n(s)})\right\}. \end{aligned} \tag{13}$$

確率分布 $Q_{\mathrm{M}}(\mathbf{h}|\mathbf{w})$ の変数間には依存関係が存在しないため，$Q_{\mathrm{M}}(\mathbf{h}|\mathbf{w})$ からサンプルを生成するのは容易である[*2]．また準ニュートン法の反復計算では，同じサンプルを繰り返し使用した．

2.4 局所的素性

局所的素性としては，MSTParser version 0.4.2[*3] で使用されているものと同一の素性を用いた．この素性には，係り元のトークン，係り先のトークン，それらの周辺および間に存在するトークンの，語形や品詞タグが含まれる．係り先のトークンから係り元のトークンへの方向と距離も含まれる．また，各トークンに対する詳細な形態素情報等が利用できる場合には，それも含まれる．

実験では，訓練データ中に 5 回未満しか出現しなかった素性は削除した．

[*2] $Q_{\mathrm{M}}(\mathbf{h}|\mathbf{w})$ からのサンプルの生成は，単純に文中の各トークンの係り先を独立に選ぶことにより行った．そのためサイクルを含むような，依存構造木としては不適切なサンプルが生成される可能性がある．解析時に $P_{\Lambda,\mathrm{M}}(\mathbf{h}|\mathbf{w})$ からギブスサンプリングにより生成されるサンプルも，やはりそのような不適切な構造を含む可能性がある．このように提案手法では，不適切な解候補を含んだ必要以上に大きな解空間を探索しており，それにより係り受け解析の精度が低下する可能性があるが，本研究では計算上の効率の点からこのような手法を用いた．

[*3] http://sourceforge.net/projects/mstparser/

2.5 大域的素性

大域的素性として，9 種類の素性を利用した．これらの素性は Collins ら [4] が使用した素性と似ているが，彼らの素性は句構造解析のために利用されたものであり，係り受け解析のためにここで用いる素性とはやや異なる．以下では，図 1 の依存構造木を例にして説明する．

2.5.1 Child Unigram+Parent+Grandparent

この素性テンプレートは，以下の要素から構成される 4 つ組である: (1) 子ノード，(2) その親ノード，(3) 親ノードから子ノードへの方向 (左 (l) または右 (r))，(4) 祖父母ノード (もし親ノードがルートノードの場合は特別な記号 ϕ とする)．

実際の素性は，素性テンプレート中の各ノード (トークン) をその語形と品詞に展開して作成する．実験では，5 つ未満の文にしか出現しなかった素性は削除した．これらの手続きは，以下の他の素性についても同様に適用した．

例 (図 1): $\langle A, B, l, C\rangle$, $\langle B, C, l, \phi\rangle$, $\langle D, G, l, C\rangle$, $\langle E, G, l, C\rangle$, $\langle F, G, l, C\rangle$, $\langle G, C, r, \phi\rangle$, $\langle H, C, r, \phi\rangle$.

2.5.2 Child Bigram+Parent

この素性テンプレートは，以下の要素から構成される 4 つ組である: (1) 子ノード，(2) その親ノード，(3) 親ノードから子ノードへの方向，(4) 最も近い外側 (親ノードとは反対側) の兄弟ノード (該当するノードが無い場合は特別な記号 ϕ とする)．この素性テンプレートは，McDonald ら [9] により二次の素性として用いられたものとほぼ同じである．

例 (図 1): $\langle A, B, l, \phi\rangle$, $\langle B, C, l, \phi\rangle$, $\langle D, G, l, \phi\rangle$, $\langle E, G, l, D\rangle$, $\langle F, G, l, E\rangle$, $\langle G, C, r, H\rangle$, $\langle H, C, r, \phi\rangle$.

2.5.3 Child Bigram+Parent+Grandparent

この素性テンプレートは 5 つ組である．最初の 4 つの要素 (1)–(4) は Child Bigram+Parent と同じである．残りの 1 つの要素 (5) は祖父母ノードである．

例 (図 1): $\langle A, B, l, \phi, C\rangle$, $\langle B, C, l, \phi, \phi\rangle$, $\langle D, G, l, \phi, C\rangle$, $\langle E, G, l, D, C\rangle$, $\langle F, G, l, E, C\rangle$, $\langle G, C, r, H, \phi\rangle$, $\langle H, C, r, \phi, \phi\rangle$.

2.5.4 Child Trigram+Parent

この素性テンプレートは 5 つ組である．最初の 4 つの要素 (1)–(4) は Child Bigram+Parent と同じである．残りの 1 つの要素 (5) は 2 番目に近い外側の兄弟ノードである．

例 (図 1): $\langle A, B, l, \phi, \phi\rangle$, $\langle B, C, l, \phi, \phi\rangle$, $\langle D, G, l, \phi, \phi\rangle$, $\langle E, G, l, D, \phi\rangle$, $\langle F, G, l, E, D\rangle$, $\langle G, C, r, H, \phi\rangle$, $\langle H, C, r, \phi, \phi\rangle$.

2.5.5 Parent+All Children

この素性テンプレートは 1 つ以上の要素から構成される組である．最初の要素 (1) は親ノードであり，残りの要素はその全ての子ノードである．親ノードの左側にある子ノードと右側にある子ノードは区別する必要があるため，下の例では後者にプライムを付けている．

例 (図 1): $\langle A\rangle$, $\langle B, A\rangle$, $\langle C, B, G', H'\rangle$, $\langle D\rangle$, $\langle E\rangle$, $\langle F\rangle$, $\langle G, D, E, F\rangle$, $\langle H\rangle$.

2.5.6 Parent+All Children+Grandparent

この素性テンプレートは 2 つ以上の要素から構成される組である．最後の要素以外は，Parent+All Children と同じである．最後の要素は祖父母ノードである．

例 (図 1): $\langle A, B\rangle$, $\langle B, A, C\rangle$, $\langle C, B, G', H', \phi\rangle$, $\langle D, G\rangle$, $\langle E, G\rangle$, $\langle F, G\rangle$, $\langle G, D, E, F, C\rangle$, $\langle H, C\rangle$.

2.5.7 Child+Ancestor

この素性テンプレートは次の要素から構成される 2 つ組である: (1) 子ノード，(2) その祖先ノードの 1 つ．田村らは，このような素性を用いることにより日本語係り受け解析の精度を改善できたと報告している [17].

手法	DA (%)	CM (%)
MSTParser proj. (一次の素性)	91.0	37.5
MSTParser proj. (二次の素性)	**91.7**	**42.6**
MSTParser non-proj. (一次の素性)	90.4	34.1
MSTParser non-proj. (二次の素性)	91.5	40.6
Gibbs proj. (大域的素性無し)	90.6	34.9
Gibbs proj. (大域的素性有り)	**91.7**	42.0
Gibbs non-proj. (大域的素性無し)	90.3	32.9
Gibbs non-proj. (大域的素性有り)	91.5	40.8

表 1　WSJ コーパスでの係り受け解析結果

例 (図 1): $\langle A, B\rangle$, $\langle A, C\rangle$, $\langle B, C\rangle$, $\langle D, G\rangle$, $\langle D, C\rangle$, $\langle E, G\rangle$, $\langle E, C\rangle$, $\langle F, G\rangle$, $\langle F, C\rangle$, $\langle G, C\rangle$, $\langle H, C\rangle$.

2.5.8 Acyclic

この素性は 2 つの値のうちのどちらかを持つ．もし依存構造木がサイクルを含まなければ *true* であり，サイクルを含めば *false* である．

例 (図 1): *true*

2.5.9 Projective

この素性は 2 つの値のうちのどちらかを持つ．もし依存構造木が交差を含まなければ *true* であり，交差を含めば *false* である．

例 (図 1): *true*

3 実験

実験には，Penn Treebank WSJ corpus(英語) と，CoNLL-X shared task のデータセット (デンマーク語，オランダ語，ポルトガル語，スウェーデン語) を使用した．係り受け解析の精度を評価するために，下記の評価指標を用いた:

$$\textbf{係り先精度 (DA)} = \frac{\langle \text{係り先が正しく同定されたトークン数} \rangle}{\langle \text{全トークン数} \rangle},$$

$$\textbf{文正解率 (CM)} = \frac{\langle \text{正しく解析された文数} \rangle}{\langle \text{文の総数} \rangle}.$$

なお，句読点は除いて精度を計算した．

3.1 WSJ コーパスでの実験結果

Penn Treebank WSJ コーパスを使用して実験を行った．WSJ コーパスは句構造文法のアノテーションを持つが，山田らの主辞規則 [15] を用いて依存構造木に変換した．section 2–21 を訓練用，section 22 をパラメータ調整用，section 23 を評価用に使用した．パラメータ調整用と評価用データに対する品詞タグは，訓練用データにて学習した品詞タグ付けシステム [20] を用いて付与した (評価用データに対するタグ付けの精度は 97.1% であった)．ハイパーパラメータ等の値は次のように設定した: $\sigma = 0.25$, $\sigma' = 0.25$, $R = 100$, $S = 100$. 局所的素性の数は 14,910,831 であり，大域的素性の数は 4,946,085 であった．実験は，Opteron 250 プロセッサと 8GB のメモリを持つ計算機で行った．計算時にメモリが足りなかったため，学習データの素性ベクトルはディスク上に格納して処理した．局所的モデルと大域的モデルの両方のパラメータを学習するのに要した時間は約 28 時間であり，評価用データを CLE アルゴリズムを用いて解析するのに要した時間は約 71 分であった．

実験結果を表 1 に示す．この表で，**Gibbs proj.** は提案手法で Eisner アルゴリズムを用いた場合の結果を，**Gibbs non-proj.** は提案手法で CLE アルゴリズムを用いた場合の結果を，**(大域的素性無し)** は局所的素性のみを用いて大域的素性は用いなかった場合の結果を，**(大域的素性有り)** は局所的素性と大域的素性の両方を用いた場合の結果をそれぞれ表す．解析精度の比較を行うために，MSTParser version 0.2 を用いて係り受け解析を行った結果を，**MSTParser** として示してある．この係り受け解析器は交差の無い係り受

素性	DA (%) / CM (%)		素性の数
	追加した場合	削除した場合	
None	90.3 / 32.9	91.5 / 40.8	0
All	91.5 / 40.8	90.3 / 32.9	4,946,085
Child Unigram+Parent+Grandparent	90.7 / 35.5	91.4 / 40.4	587,836
Child Bigram+Parent	90.9 / 37.4	91.4 / 40.6	499,079
Child Bigram+Parent+Grandparent	91.2 / 38.5	91.3 / 40.2	1,835,514
Child Trigram+Parent	91.1 / 38.4	91.4 / 40.4	1,322,725
Parent+All Children	90.6 / 35.5	91.5 / 40.4	104,676
Parent+All Children+Grandparent	90.7 / 35.0	91.5 / 40.8	163,310
Child+Ancestor	90.4 / 33.4	91.4 / 40.4	432,941
Acyclic	90.2 / 32.5	91.3 / 40.7	2
Projective	90.3 / 32.9	91.5 / 40.6	2

表 2　各素性を利用した場合/利用しなかった場合の係り受け解析結果: 各素性に対して，その素性のみを使用した場合 (追加した場合) と，その素性のみを使用しなかった場合 (削除した場合) の結果.

け解析 [7] も，交差の有る係り受け解析 [8] も扱うことができ，また**一次の素性** (局所的素性) だけではなく**二次の素性** (大域的素性の一種)[9] も利用することができる．提案手法は，大域的素性を利用した場合に高い精度を得た．全ての大域的素性を用いた場合の Gibbs proj. の係り先精度 (DA) は，二次の素性を用いた場合の MSTParser proj. の精度と等しかった．しかしながら，提案手法では MSTParser よりも多くの大域的素性を利用しているにも関わらず，文正解率 (CM) はやや低かった．考えられる原因の一つとしては，学習アルゴリズムの違いが挙げられる．MSTParser ではマージン最大化に基づくオンライン学習アルゴリズム (MIRA) を利用しているが，提案手法では exponential model を Gaussian prior と共に用いている．また別の原因として，提案手法ではサンプリングを用いた近似計算を行っており，これにより十分な精度が得られなかった可能性も考えられる．

個々の大域的素性の効果を調べるために行った実験の結果を，表 2 に示す．各大域的素性に対して，2 通りの場合について CLE アルゴリズムを使用して実験を行った．1 つはその大域的素性のみを使用した場合 (**追加した場合**) であり，もう 1 つはその大域的素性以外の他の全ての素性を使用した場合 (**削除した場合**) である．各大域的素性を追加した場合，Acyclic と Projective 以外の全ての場合において精度が向上した．各大域的素性を削除した場合，精度はあまり低下しなかった．このことから，これらの素性は比較的大きな冗長性を持っていると思われる．

次に，大域的モデルのパラメータを学習した結果，各素性に対してどのような重みが与えられたかを調べた．一般的に，大きな重みが与えられた素性ほど，そのモデルを用いた分類に大きな影響を与えると考えられる．大域的素性の中で，1 つも子ノードを持たない，または 2 つ以上の子ノードを持つ前置詞を表す素性は大きな負の値を持っており，ちょうど 1 つの子ノードを持つ前置詞を表す素性は大きな正の値を持っていた．前置詞は多くの場合 1 つの子ノードしか持たないが，そのような性質などが大域的素性により学習されたと考えられる．

3.2 CoNLL-X Shared Task データでの実験結果

CoNLL-X shared task(多言語係り受け解析)[2] で用いられたデータを使用して実験を行った．デンマーク語，オランダ語，ポルトガル語，スウェーデン語のデータ[*4] を利用した．提案手法で使われるハイパーパラメータ等の値は，WSJ コーパスでの実験で使用されたのと同じ値を用いた．この shared task では依存関係ラベルの正解率も含めてシステムの性能評価を行ったが，提案手法では依存関係ラベルの推定は行わないため，WSJ コーパスの場合と同様に係り先精度を用いて評価を行った．

実験結果を表 3 に示す．解析精度の比較を行うために，この shared task に参加した上位 3 つのシステムの結果も示してある (**CoNLL 1st, 2nd, 3rd**)．提案手法は，4 つの全ての言語において，最も高い係り先精度を得た．2 番目に精度の高いシステムとの有意差検定を行ったところ，オランダ語のデータで有意差 ($p < 0.05$) があった (デンマーク語，オランダ語，ポルトガル語，スウェーデン語のデータでの p 値は，それぞれ 0.10，0.0047，0.46，0.29 であった)．

4 関連研究

大域的な素性を用いて係り受け解析を行う方法は，これまでにもいくつか研究されている．

工藤ら [19] は，決定的 (deterministic) な日本語の係り受け解析手法を提案した．彼らの方法では解析は決定的に行われるため，既に係り先が決まったトークンに関する情報をその後の処理で利用することができる．動的素性と名付けられたそのような素性を利用することで，解析精度が大きく向上したと報告している．また英語の決定的な係り受け解析においても，同様の素性が有効であることが山田ら [15] により報告されている．決定的な解析手法では，計算量を増加させることなくこのような素性を利用できる．しかしながら利用可能な素性は，処理を行う時点で既に決定している構造に関する情報に限られる．

McDonald ら [9] は二次の素性 (依存構造木中で，2 つの隣り合う兄弟ノードとその親ノード間を結ぶ，2 つのエッジに関する素性) を利用することができる係り受け解析手法を提案した．彼らの手法では二次の Eisner アルゴリズムを用いて，交差の無い係り受け解析を行う．交差の有る係り受け解析を行う場合には，そこで得られた交差の無い依存構造木を greedy アルゴリズムを用いて変形することにより，より最適な交差の有る依存構造木を得る．この方法は効率的で高い精度を得ることができる．しかしながら，m 次の Eisner アルゴリズムの計算量は $O(n^{m+1})$ であるため [9]，この手法でさらに高次の素性を扱うのは難しい．

Riedel ら [12] は，整数線形計画法 [14] を用いた係り受け解析のための手法を提案した．彼らは，「等位接続詞が結ぶトークンは同じ品詞でなければならない」「動詞は 2 つ以上の主語を持ってはいけない」というような言語学的な制約を取り入れて解析を行うために，整数線形計画法を利用した．この手法は文単位の様々な情報を利用することができる．しかしながら大域的な情報を，学習可能な重みが付与されて柔軟に制約の度合いが変えられる素性として利用するのではなく，必ず満たさなければならない強い制約として利用するため，ノイズの多いデータなどに対しては問題が生じることが指摘されている [12]．

係り受け解析以外の言語処理においても，大域的な情報の利用は試みられている．Collins らは，リランキングに基

[*4] http://nextens.uvt.nl/~conll/

手法	DA (%)			
	デンマーク語	オランダ語	ポルトガル語	スウェーデン語
CoNLL 1st	90.58	83.57	91.36	89.54
CoNLL 2nd	89.80	82.91	91.22	89.50
CoNLL 3rd	89.66	81.73	90.30	89.05
Gibbs proj. (大域的素性無し)	89.94	80.73	90.84	88.83
Gibbs proj. (大域的素性有り)	90.82	81.55	91.38	**89.80**
Gibbs non-proj. (大域的素性無し)	90.04	83.71	90.66	88.73
Gibbs non-proj. (大域的素性有り)	**90.96**	**84.85**	**91.40**	89.62

表 3 CoNLL-X Shared Task データでの係り受け解析結果

づく句構造解析手法を提案した [4]. この方法では，まず基本的な構文解析器 (base parser) を用いて入力された文を解析し，n-best 解を求める．次にその解候補に対して，候補の構文木中の素性を利用して，別の統計的モデルにより再度順位付けを行う．この方法は任意の文単位の素性を用いることができる．しかしながら，n-best 解を得るために使用される基本的な解析器は一般的にあまり複雑な素性を扱うことはできず，もし n-best 解の中に正解が含まれていなければ正しい解を得ることはできない．そのため，より良い n-best 解を得るための研究も行われている [3]. Johnson らは，単一化文法のための exponential model を提案した [6]. このモデルでは大域的な素性を利用することが可能であり，pseudo-likelihood estimator を用いてパラメータの推定を行っている．Rosenfeld らは whole-sentence exponential language model を提案した [13]. 彼らは文中の任意の素性を利用して言語モデルを構築するために，exponential model を用いて確率モデルを定義し，ギブスサンプリング等のサンプリング手法を用いて計算を行っている．

5 結論

本稿では，文中の任意の素性を利用することができる，ギブスサンプリングを用いた係り受け解析手法について検討した．この手法は最大全域木に基づく係り受け解析手法を応用したものであり，エッジのスコアには構文木全体の確率分布からギブスサンプリングを用いて計算される周辺確率を使用する．実験の結果，大域的な素性を利用することにより高い精度が得られることを確認した．

本稿ではトークン間の依存関係のみに注目し，係り受け関係のラベルについては考慮しなかった．各トークンの係り先だけではなく係り受け関係のラベルも扱う場合，それらの組み合わせを考慮しなければならないため，係り先のみを扱う場合に比べて探索空間が非常に大きくなる．しかしながら，そのような高次元の探索空間はギブスサンプリングで扱うことが可能であると思われる．依存関係のラベルも考慮した係り受け解析を行うことは今後の課題である．

謝辞

英語の主辞規則を提供していただいた (株) ジャストシステムの山田寛康氏に深く感謝いたします．

参考文献

[1] Abney, S. P.: Stochastic Attribute-Value Grammars, *Computational Linguistics*, Vol. 23, No. 4, pp. 597–618 (1997).

[2] Buchholz, S. and Marsi, E.: CoNLL-X Shared Task on Multilingual Dependency Parsing, *Proceedings of CoNLL 2006*, pp. 149–164 (2006).

[3] Charniak, E. and Johnson, M.: Coarse-to-fine n-best parsing and MaxEnt discriminative reranking, *Proceedings of ACL 2005*, pp. 173–180 (2005).

[4] Collins, M. and Koo, T.: Discriminative Reranking for Natural Language Parsing, *Computational Linguistics*, Vol. 31, No. 1, pp. 25–69 (2005).

[5] Covington, M. A.: A Fundamental Algorithm for Dependency Parsing, *Proceedings of ACM Southeast Conference 2001*, pp. 95–102 (2001).

[6] Johnson, M., Geman, S., Canon, S., Chi, Z. and Riezler, S.: Estimators for Stochastic "Unification-Based" Grammars, *Proceedings of ACL '99*, pp. 535–541 (1999).

[7] McDonald, R., Crammer, K. and Pereira, F.: Online Large-Margin Training of Dependency Parsers, *Proceedings of ACL 2005*, pp. 91–98 (2005).

[8] McDonald, R., Pereira, F., Ribarow, K. and Hajic, J.: Non-projective Dependency Parsing using Spanning Tree Algorithms, *Proceedings of HLT/EMNLP 2005*, pp. 523–530 (2005).

[9] McDonald, R. and Pereira, F.: Online Learning of Approximate Dependency Parsing Algorithms, *Proceedings of EACL 2006*, pp. 81–88 (2006).

[10] Nivre, J.: An Efficient Algorithm for Projective Dependency Parsing, *Proceedings of IWPT 2003*, pp. 149–160 (2003).

[11] Pietra, S. D., Pietra, V. D. and Lafferty, J.: Inducing Features of Random Fields, *IEEE Transactions Pattern Analysis and Machine Intelligence*, Vol. 19, No. 4, pp. 380–393 (1997).

[12] Riedel, S. and Clarke, J.: Incremental Integer Linear Programming for Non-projective Dependency Parsing, *Proceedings of EMNLP 2006*, pp. 129–137 (2006).

[13] Rosenfeld, R., Chen, S. F. and Zhu, X.: Whole-Sentence Exponential Language Models: A Vehicle For Linguistic-Statistical Integration, *Computers Speech and Language*, Vol. 15, No. 1, pp. 55–73 (2001).

[14] Roth, D. and Yih, W.: A Linear Programming Formulation for Global Inference in Natural Language Tasks, *Proceedings of CoNLL 2004*, pp. 1–8 (2004).

[15] 山田寛康, 松本裕治: Support Vector Machine を用いた決定性上昇型依存構造解析, 情報処理学会論文誌, Vol. 45, No. 10, pp. 2416–2427 (2004).

[16] 伊庭幸人, 種村正美, 大森裕浩, 和合肇, 佐藤整尚, 高橋明彦: 統計科学のフロンティア 12 計算統計 II マルコフ連鎖モンテカルロ法とその周辺, 岩波書店 (2005).

[17] 田村晃裕, 高村大也, 奥村学: 符号化問題として解く日本語係り受け解析, 情報処理学会研究報告 2006-NL-176, pp. 17–24 (2006).

[18] 内元清貴, 関根聡, 井佐原均: 最大エントロピー法に基づくモデルを用いた日本語係り受け解析, 情報処理学会論文誌, Vol. 40, No. 9, pp. 3397–3407 (1999).

[19] 工藤拓, 松本裕治: チャンキングの段階適用による係り受け解析, 情報処理学会論文誌, Vol. 43, No. 6, pp. 1834–1842 (2002).

[20] 中川哲治, 工藤拓, 松本裕治: Support Vector Machine を用いた形態素解析と修正学習法の提案, 情報処理学会論文誌, Vol. 44, No. 5, pp. 1354–1367 (2003).