自然言語処理 149－9
（2002．5．23）

Support Vector Machine を用いた決定性上昇型構文解析

山田 寛康†，松本 裕治††

†北陸先端科学技術大学院大学 情報科学研究科

††奈良先端科学技術大学院大学 情報科学研究科

h-yamada@jaist.ac.jp, matsu@is.aist-nara.ac.jp

本稿では，機械学習アルゴリズム Support Vector Machine を用いた英語構文解析法を提案する．高精度な構文解析を行うには，句のラベルだけでなく，句の主辞がもつ語彙情報をも考慮する必要がある．しかし従来の統計的構文解析モデルは，データスパースネスの問題から，主辞の語彙情報を素性として大量に使用することは，逆に精度低下の要因となっていた．機械学習アルゴリズム Support Vector Machine は 素性空間の次元数に依存しない高い汎化性能と，Kernel 関数によって素性の組合せまでも考慮した学習が可能である．そのため主辞の語彙情報を含めた多くの素性とその組合わせを考慮した学習が行える．しかし，SVM は 確率を推定するのではなく，2 つのクラスを識別する分類器であり，従来多くの統計的構文解析モデルが採用している確率モデルへの直接的な適用が困難である．
本稿では，上昇型解析アルゴリズムを用い，構文解析の各段階を，文脈に適切な解析木構築手続きへの分類問題とみなすことで SVM を適用し，解析木構築規則の学習を行う．解析木は，SVM が分類器であることから，決定的に構築される．本手法を Penn Treebank コーパスを用いて評価した結果，labeled recall/precision で 88.2/89.0%という高い精度を得ることができた．

キーワード：統計的構文解析，決定性構文解析，サポートベクター学習

Deterministic Bottom-up Parsing with Support Vector Machines

YAMADA Hiroyasu †，MATSUMOTO Yuji ††

†Japan Advanced Institute of Science and Technology

††Graduate School of Information Science, Nara Institute Science and Technology

h-yamada@jaist.ac.jp, matsu@is.aist-nara.ac.jp

In this paper, we propose a parsing method for English sentences with machine learning algorithm called Support Vector Machines (SVMs). The performance of statistical parsing strongly depends on how to deal with lexical information and incorporate them into the statistics for parsing. Data sparseness problem arises when using large number of features like head words. As a result, we cannot estimate correct statistics for construction of parse trees. SVMs not only have high generalization performance in sparse data using a large number of features like head words, but also can take into account the combinations of features by virtue of polynomial kernel functions. However, SVMs are classifiers, not probabilistic estimator. Thus, it is difficult to apply SVMs to the probabilistic parsing model directly.
Our parser constructs a parse tree for an input sentence with a deterministic bottom-up algorithm. Each parsing process is regarded as a classification task which classifies the context into a procedure for constructing parsed trees. We evaluated our parser using the Penn Treebank corpus, and the result attained over the 88.2/89.0% labeled recall/precision.

Keywords : Statistical Parsing, Deterministic Parsing, Support Vector Learning

1 はじめに

構文解析は，自然言語処理における基礎技術の一つであり，また機械翻訳や情報検索など幅広い応用に使用されるため，汎用的で高精度な解析を行う必要がある．統計的構文解析は，解析済コーパスから，解析に必要な規則を自動学習することで，高い汎用性と解析精度を実現する．近年，統計的構文解析において，主辞の語彙情報を考慮することが重要であると報告されている．しかし主辞の語彙情報すべてを考慮する場合，データスパースネスの問題から，解析に必要な統計量や確率値を適切に推定できないという問題がある．

機械学習アルゴリズム Support Vector Machine [9] は，汎化性能が，使用する素性の数に依存しない理論的な裏付けがある．このため統計的係り受け解析 [8] をはじめ，単語や品詞情報など多数な素性を使用する必要がある自然言語解析規則の学習において，非常に高い精度が報告されている [6, 4]．また SVM は多項式 Kernel 関数を適用することで，素性の組合せを網羅的に考慮した学習が実現できる．

しかし SVM は確率値を推定するのではなく，2 つのクラスを識別する分類器である．そのため，従来の多くの統計的構文解析が採用している確率モデルを直接適用することが困難である．

本稿では，SVM を用いて英語構文解析を行うための手法を提案する*．解析木構築の過程を，文脈情報から，適切な解析木を構築する手続きへの分類問題とみなすことで，SVM を適用し解析木構築規則の学習を行う．ま

*この研究は筆者が奈良先端科学技術大学院大学博士後期課程在学中に行った研究である．

自然言語処理 149－9
（2002．5．23）

Support Vector Machine を用いた決定性上昇型構文解析

山田 寛康†，松本 裕治††

†北陸先端科学技術大学院大学 情報科学研究科

††奈良先端科学技術大学院大学 情報科学研究科

h-yamada@jaist.ac.jp, matsu@is.aist-nara.ac.jp

本稿では，機械学習アルゴリズム Support Vector Machine を用いた英語構文解析法を提案する．高精度な構文解析を行うには，句のラベルだけでなく，句の主辞がもつ語彙情報をも考慮する必要がある．しかし従来の統計的構文解析モデルは，データスパースネスの問題から，主辞の語彙情報を素性として大量に使用することは，逆に精度低下の要因となっていた．機械学習アルゴリズム Support Vector Machine は 素性空間の次元数に依存しない高い汎化性能と，Kernel 関数によって素性の組合せまでも考慮した学習が可能である．そのため主辞の語彙情報を含めた多くの素性とその組合わせを考慮した学習が行える．しかし，SVM は 確率を推定するのではなく，2 つのクラスを識別する分類器であり，従来多くの統計的構文解析モデルが採用している確率モデルへの直接的な適用が困難である．
本稿では，上昇型解析アルゴリズムを用い，構文解析の各段階を，文脈に適切な解析木構築手続きへの分類問題とみなすことで SVM を適用し，解析木構築規則の学習を行う．解析木は，SVM が分類器であることから，決定的に構築される．本手法を Penn Treebank コーパスを用いて評価した結果，labeled recall/precision で 88.2/89.0%という高い精度を得ることができた．

キーワード：統計的構文解析，決定性構文解析，サポートベクター学習

Deterministic Bottom-up Parsing with Support Vector Machines

YAMADA Hiroyasu †，MATSUMOTO Yuji ††

†Japan Advanced Institute of Science and Technology

††Graduate School of Information Science, Nara Institute Science and Technology

h-yamada@jaist.ac.jp, matsu@is.aist-nara.ac.jp

In this paper, we propose a parsing method for English sentences with machine learning algorithm called Support Vector Machines (SVMs). The performance of statistical parsing strongly depends on how to deal with lexical information and incorporate them into the statistics for parsing. Data sparseness problem arises when using large number of features like head words. As a result, we cannot estimate correct statistics for construction of parse trees. SVMs not only have high generalization performance in sparse data using a large number of features like head words, but also can take into account the combinations of features by virtue of polynomial kernel functions. However, SVMs are classifiers, not probabilistic estimator. Thus, it is difficult to apply SVMs to the probabilistic parsing model directly.
Our parser constructs a parse tree for an input sentence with a deterministic bottom-up algorithm. Each parsing process is regarded as a classification task which classifies the context into a procedure for constructing parsed trees. We evaluated our parser using the Penn Treebank corpus, and the result attained over the 88.2/89.0% labeled recall/precision.

Keywords : Statistical Parsing, Deterministic Parsing, Support Vector Learning

1 はじめに

構文解析は，自然言語処理における基礎技術の一つであり，また機械翻訳や情報検索など幅広い応用に使用されるため，汎用的で高精度な解析を行う必要がある．統計的構文解析は，解析済コーパスから，解析に必要な規則を自動学習することで，高い汎用性と解析精度を実現する．近年，統計的構文解析において，主辞の語彙情報を考慮することが重要であると報告されている．しかし主辞の語彙情報すべてを考慮する場合，データスパースネスの問題から，解析に必要な統計量や確率値を適切に推定できないという問題がある．

機械学習アルゴリズム Support Vector Machine [9] は，汎化性能が，使用する素性の数に依存しない理論的な裏付けがある．このため統計的係り受け解析 [8] をはじめ，単語や品詞情報など多数な素性を使用する必要がある自然言語解析規則の学習において，非常に高い精度が報告されている [6, 4]．また SVM は多項式 Kernel 関数を適用することで，素性の組合せを網羅的に考慮した学習が実現できる．

しかし SVM は確率値を推定するのではなく，2 つのクラスを識別する分類器である．そのため，従来の多くの統計的構文解析が採用している確率モデルを直接適用することが困難である．

本稿では，SVM を用いて英語構文解析を行うための手法を提案する*．解析木構築の過程を，文脈情報から，適切な解析木を構築する手続きへの分類問題とみなすことで，SVM を適用し解析木構築規則の学習を行う．ま

*この研究は筆者が奈良先端科学技術大学院大学博士後期課程在学中に行った研究である．

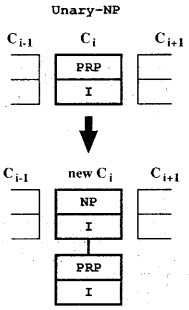

図 2: 手続き 'Unary-NP'

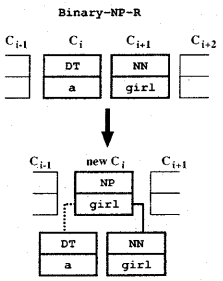

図 3: 手続き 'Binary-NP-R'

成したノードの主辞は H (右 (R) または左 (L)) で示される方向の子ノードとする．図 3 に **Binary-NP-R** の例を示す．文頭から i 番目の位置で，ラベル NP のノード new C_i を作成し C_i を左の子，C_{i+1} を右の子とし二分木を構築する．そして新たに生成したノードの主辞は R で示された方向に位置する子ノード，即ち "girl" となる．

- **Shift**: 解析位置を一つ右に shift する．図 4 に **Shift** の例を示す．

次に上に示した 3 種類の手続きを用いてどのように解析木を構築するかを説明する．図 5 に解析アルゴリズムの疑似コードを示す．

図 5 で，s は解析された部分木のルートノードを保持し，$|s|$ は構築された部分木の数を表す．また s_i は文頭から i 番目の部分木を表す．関数 $get\_contextual\_feature(s,i)$ は，現在の解析位置 i において s_i の周りの文脈から素性を抽出し，その素性ベクトル $\mathbf{x}$ を返す関数である．具体的な素性については 4.2 節で述べる．*model* は学習された SVM を表し，関数 *classify* は，*model* と関数 $get\_contextual\_information$ で抽出した素性ベクトル $\mathbf{x}$ から，現在の解析位置における適切な解析木構築手続き y を返す．*construction* は解析木構築手続き y と s を引数に，手続きを実行し解析木を構築する．

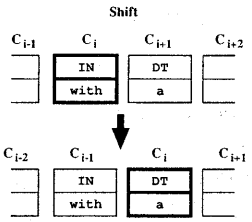

図 4: 手続き 'Shift'

本手法では，入力文に対し品詞タグ付けを行った結果を入力とする．解析の初期状態では，s には入力文の n 個の単語と品詞のペア $c_i(1 \leq i \leq n)$ が終端ノードとして格納される．解析位置を表す i は 1 とする．

解析は文頭から文末の方向へ順に進み，各解析位置 i のおいて，周りの文脈から抽出した素性ベクトル $\mathbf{x}$ を基に，関数 *classify* により解析木構築手続き y を決定する．そして *construction* で手続き y を実行し，解析木を決定的に構築する．これらの処理を繰返し行い，解析位置が文末に達し場合 $(i == |s|+1)$，変数 *no_construction* の真偽値を調べる．変数 *no_construction* が *true* であった場合，即ち解析木を一つも構築することができないことを意味し．ここで解析を終了する．変数 *no_construction* が *false* であれば，文頭にもどり同様の処理を繰り返す．解析を終了した時点で解析木が一つの場合 $(|s| == 1)$，解析が成功したことになり，それ以外は部分的に解析ができたことを意味する．図 6 は 'I saw a girl with a small telescope' という文に対して↓で示す解析位置での解析過程を示す．図 6 で破線で囲まれた横に列挙されているノードの列が，s を表している．(a) は，関数 *classify* により **Binary-NP-R** という解析木構築手続きを推定した場合を表している．(b) は *construction* により手続き **Binary-NP-R** を実行し，二分木を構築した結果を表す．

3.1 N 進木から二分木への変換

前節で説明した解析アルゴリズムでは，二分木の解析木しか構築できない．しかし一般の構文木は N 進木の形式であるため，N 進木を扱えるように拡張する必要がある．本稿では主辞規則を利用し，N 進木と二分木を相互に変換を行う方法を用いる．主辞規則は，各句の中でどのノードが主辞であるを判別する規則で，今回 Collins[2] らが用いた規則に対し，いくつか修正したものを用いた．

主辞規則は，主辞規則を適用する句のラベル，主辞探索の開始位置，及び主辞決定条件の 3 つから構成される．主辞探索の開始位置は，right であれば，適用する句の最

Input Sentence: $s = c_1, c_2, \cdots, c_i, \cdots c_n$
Initialize:
$i = 1$;
no_construction = *true* ;
Start:
while *true* do begin
if $i == |s| + 1$ then
if no_construction == *true* then break;
no_construction = *true*
$i = 1$;
else
$\mathbf{x} = get\_contextual\_features(s, i)$;
$y = classify(model, \mathbf{x})$;
if $y ==$ **Unary-X** or **Binary-X-H** then
$s = construction(s, i, y)$;
no_construction = *false* ;
else if $y ==$ **Shift** then
$i = i + 1$
end;
end;
end;

図 5: 構文解析アルゴリズム疑似コード

右の子ノードから探索し, left であれば最左の子ノードから開始する. 主辞決定条件に列挙してあるラベルは, 探索してるノードのラベルと比較し, 一致した場合主辞と判定する (“,” で区切られた要素の一つ一つが条件に対応する).

表 1 は名詞句における主辞規則を示す. 条件の要素が () で括られた場合, それを 1 つの条件とみなし, 括弧内のラベルのいずれかと一致すれば主辞に決定されることを意味する. 表 1 の場合, 句の最右のノードから順にしらべていき, POS, NN, NNP, NNPS, または NNS のいずれかのラベルを持ったノードが存在すれば, そのノードが主辞となる. 複数存在する場合は開始位置が right であるため, より右の子が主辞となる. もし該当ラベルを持つノードが存在しなければ, 条件の次の要素である NX について最右ノードから順に調べていく. 条件に該当するラベルをもったノードが一つも見つからなければ, 開始位置である最右ノードを主辞とする.

次に, 主辞規則を利用した二分木への変換方法について説明する. 図 7 は, ラベル X で, 4 つの子ノードを持つ解析木で, その主辞が主辞規則から H と判定されたものとする. 二分木への変換は, まず主辞を中心に左隣のノードと二分木を作る (図 7(2)). これを主辞より左のノードに対し再帰的に繰り返す (図 7(3)). すべての

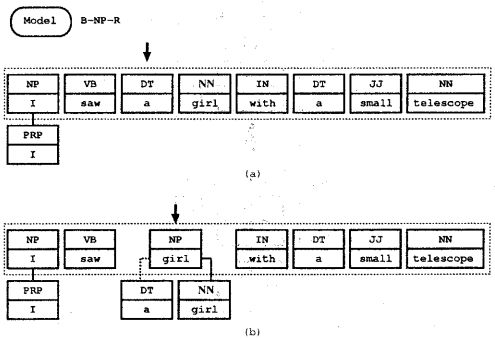

図 6: 構文解析の例

左ノードに対し変換が終了したら, 同様のことを主辞より右のノードにも適用し, 二分木に変換する (図 7(4)). ここで構築した各二分木のラベルを X' とし, これを中間ノードと呼び X と区別する. このようにして構築した二分木は, 主辞規則及び中間ノードを使用することで, 元の N 進木に変換可能である. この変換手法を学習及び解析の前・後処理に適用することで, 解析では常に二分木を扱うだけの簡潔な解析アルゴリズムを実現し, N 進木も扱うことが可能となる.

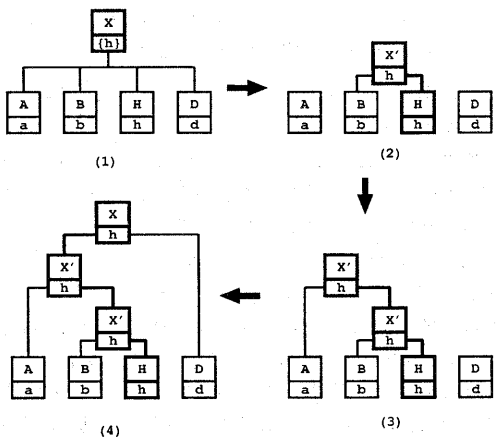

図 7: 二分木への変換

表 1: 名詞句の主辞規則

label	開始位置	決定条件
NP	right	(POS, NN, NNP, NNPS, NNS), NX, JJR, CD, JJ, JJS, RB, QP, NP

4 SVM を用いた解析木構築規則の学習

4.1 素性として考慮する文脈長

解析木構築手続きの学習に, 素性として考慮する文脈の長さについて説明する. 学習及び, 解析は i 番目の解析位置の文脈を手掛かりに行う. 考慮する文脈の長さは解析位置 i に対し左文脈の長さを l と 右文脈の r の組 (l, r) で表すことができる.

本稿では文脈の長さを固定するモデルを固定文脈長モデル, 動的に長さを変えるモデルを可変文脈長モデルと呼ぶことにする. 固定文脈長モデルは, 接続詞及び, punctuation なども一文脈長と数えるため, 分類に必要な文脈を考慮できない場合がある. 接続詞や punctuation などは, 文や節など境界に使用されるため, 後続する要素が適切な解析木を構築する重要な手掛かりとなる. そこで本稿では, これらのノードが右文脈に出現した数分, 素性として考慮する右文脈を伸長する可変文脈長モデルを提案する.

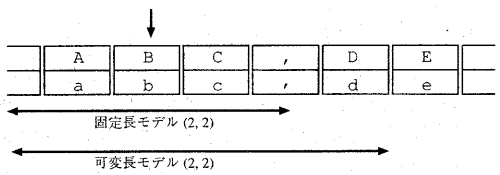

図 8: 固定文脈長と可変文脈長モデル

図 8 に左右の文脈長さを (2,2) とした場合, 固定文脈長モデルと可変文脈長モデルの相違を示す. 図 8 において, ↓の位置おいて, 固定文脈長モデルが "," までしか素性として考慮できないに対し, 可変文脈長モデルでは, その後ろのノードまで考慮することが可能となる.

4.2 素性

学習及び解析時に使用する素性について説明する. 素性は, 解析位置からの距離 p, 素性の種類 t, 素性の値 v の 3 つ組で表す. p は i より左であれば負の値, 右であれば正の値, 0 であれば 自身の位置を表す. 素性の種類について下記に示す.

- 主辞に関する素性
  label : ノードのラベル
  word : 主辞の単語
  pos : 主辞の品詞
- 前置詞句に関する素性: *PP-non-head*
  前置詞句の主辞は前置詞自身となる. このため主辞の情報だけを素性に用いると, 前置詞句の係り先に関する曖昧性を解消するために必要な情報がない. そこで前置詞句に関しては非主辞の情報を用いた. 具体的には前置詞句の非主辞の品詞, 単語, 及びラベルを用いた.
- 接続詞に関する素性: *CC*
  'and' や 'or' などの接続詞は, 複数の語や文からなる構造を表すよい手掛かりとなる. 例えば "A, B, C and D" というフレーズの解析で, 'and' は A, B, C 及び D が一つの句の構成要素であることを表す手掛かりととなる. しかし主辞の情報のみを使用すると and は主辞とならないため, 解析が進むにつれ, 接続詞が伴っていたことを素性として使用できない. そこで, 接続詞 CC に関しては, 接続詞を伴う句の解析が完成するまで, CC が伴っている句であることを素性として使用する.

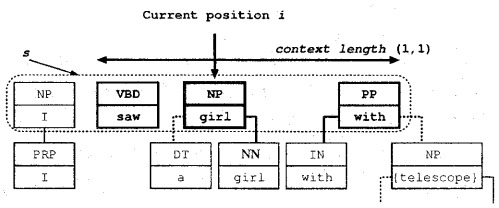

図 9: 使用する素性の例

図 9 に考慮する左右の文脈をそれぞれ 1 とした場合に, 使用する素性の例を示す. 図 9 における ↓の位置で使用する素性を以下に示す.
(-1, *label*, VBD), (-1, *pos*, VBD), (-1, *word*, saw), (0, *label*, NP), (0, *pos*, NN), (0, *word*, girl), (+1, *label*, PP), (+1, *pos*, IN), (+1, *word*, with), (+1, *PP-non-head label*, NP), (+1, *PP-non-head pos*, NN), (+1, *PP-non-head word*, telescope).

4.3 訓練事例のグループ化

学習は, 文脈から解析木を構築する手続きへの分類問題とみなし, SVM を適用する. 訓練事例は解析済の構文木をモデルとみなし, 訓練データ文を構文解析することで得られる. この場合一つの事例は解析の各段階で行った手続きに対応するため, 訓練データの文数が増加すると, 学習事例が膨大となり, 現実的な時間での学習が困難となる. 実際 Penn Treebank の section2 から section21 の訓練セット約 40,000 文から得られた訓練事例数は 3,000,000 以上となった.

この問題を解決するために, 訓練データをラベル素性毎にいくつかのグループに分割し, 分割された各事例集合で学習を行うことで, 学習に要するコストを削減する. 具体的には解析位置とその右隣のラベル素性 $(0, label, A)$ 及び $(1, label, B)$ を素性として含む事例の集め, その事例数が m 以上のものを $L_{A,B}$ として一つのグループにする. 残りの m より少ないものは L_A, とし $(0, label, A)$ を含くむ事例を集めグループ化する. 今回この閾値 m を 暫定的に 30,000 とし, 解析位置のラベルが NP のような数の多い事例を細分割した. そして個々の事例集合ごとに, SVM を用いて学習を行う.

解析は, 事例集合 $L_{A,B}$ で学習したモデルを $model_{A,B}$ とした場合, 解析に使用するモデルは解析位置及びその右隣のラベル a, b により $model_{a,b}$ を使用する. $model_{a,b}$ が存在しない場合は $model_{a,}$ を使用して分類を行う.

5 実験

5.1 データ

実験に使用したデータは Penn Treebank Wall Street Journal コーパス [5] を用いた. このコーパスは 24 分割されており, 従来, 標準データセットとして section2 から section21 の文を訓練データ, section23 をテストデータとして用いているため, 本稿もこれに従った. また, 今回テストデータの品詞は, Penn treebank に付与されているタグをそのまま使用し, 構文解析自体の解析精度を調査した.

評価は, Penn Treebank を用いた従来研究 [3, 7, 2, 1] と同様に次に示す 5 つの指標を用いた.

Labeled Recall(LR) = $\frac{\text{ラベルが一致したノード数}}{\text{treebank 全体のノード数}}$

Labeled Precision(LP) = $\frac{\text{ラベルが一致したノード数}}{\text{解析器の出力したノード数}}$

Cross Brackets(CB) = 一文あたりの平均クロス数

No crossing (0CBs) = クロスがない文の割合

2 or less crossing (2CBs) = 2 クロス以下の文の割合

また比較的短い文に対する性能をみるために, 40 単語以下からなる文のみでの評価を length ≤ 40 とし, すべての文に対する評価を all sentences とする.

5.2 結果

先ず素性として考慮する文脈長を固定にした場合と, 本稿で提案した可変長にする場合とを比較した. 結果を表 2 に示す. 表の (2,2) は左, 右文脈の長さ表している.

表 2: 固定文脈長モデルと可変文脈長モデルの比較

	(2,2)	
all sentences	固定長	可変長
LR(%)	84.2	**87.0**
LP(%)	85.1	**87.9**
CB	1.51	1.18
0CB(%)	54.6	**61.3**
2CB(%)	77.5	**82.7**
length ≤ 40		
LR(%)	85.0	**87.8**
LP(%)	85.8	**88.6**
CB	1.30	1.00
0CB(%)	58.0	**64.5**
2CB(%)	80.3	**85.3**

表 2 より, 右文脈をより長く考慮する可変文脈長モデルが, 固定文脈長モデルと比べて 5 つの評価指標いずれにおいても良い結果が得られた. 英語は右から左へ修飾する場合が多く, 本手法では文頭から文末の順に決定的に解析を行うため, 右文脈を長めに考慮することで, より正確な解析が実現される.

次に可変文脈長モデルで, 文脈長の違いによる精度の変化について調査した. 結果を表 3 に示す. 表の () 内の数字は, それぞれ左文脈の長さと右文脈の長さを表す.

表 3 より, (2, 3) の場合が最も高い精度で labeled recall/precision で 88.1/88.9%を得た. また右文脈を長く考慮する (1, 3), (2, 3) 及び (2, 4) モデルが (1, 2) と (2, 2) の精度を上回っている. この結果からも右文脈を長めに考慮することが重要であることがわかる. また最も右文脈を長く考慮する (2, 4) モデル が (2, 3) モデルより劣っている. これは常に右文脈を長く考慮するのではなく, 状況により適切に変化させる必要があることを示唆している.

非主辞情報の考慮

工藤らは SVM を用いて決定的に日本語係り受け解析を行う手法を提案し, その中で動的素性と呼ばれる, 解

表 3: 文脈長の違いによる精度

all sentences	(1,2)	(1,3)	(2,2)	(2,3)	(2,4)
LR(%)	86.7	87.3	87.0	**88.1**	87.6
LP(%)	87.6	88.3	87.9	**88.9**	88.5
CB	1.16	1.05	1.18	1.02	1.05
0CB(%)	60.9	62.5	61.3	**63.5**	62.9
2CB(%)	83.1	84.4	82.7	**85.1**	84.6
length ≤ 40					
LR(%)	87.5	88.2	87.8	**88.8**	88.4
LP(%)	88.3	88.9	88.6	**89.5**	89.0
CB	0.98	0.90	1.00	0.87	0.90
0CB(%)	64.2	65.5	64.5	**66.3**	65.8
2CB(%)	85.7	86.6	85.3	**87.3**	86.7

表 4: 非主辞情報の効果

all sentences	head-based	non-head
LR(%)	88.1	**88.2**
LP(%)	88.9	**89.0**
CB	1.02	1.03
0CB(%)	63.5	**64.2**
2CB(%)	**85.1**	85.0
length ≤ 40		
LR(%)	88.8	**89.0**
LP(%)	89.5	**89.6**
CB	0.87	0.87
0CB(%)	66.3	**66.8**
2CB(%)	**87.3**	87.2

析の各段階において動的に決定される素性を用いている [8]. 係り受け解析では, 解析の各段階で係り関係が決定すると, その影響が後の解析に多大な効果を与える. 彼らの実験では, 解析の各段階で動的に決定された係り関係を素性とすることで, 精度に大きく貢献することが報告されている.

そこで本稿でも, 解析の各段階で決定された情報を素性として使用する. 具体的には, これまでの主辞情報を基本とする素性に, 非主辞ノードのラベルを素性として追加した. 非主辞のノードは主辞のノードを修飾するため, 係り関係と見ることができる. 非主辞の素性は *non-head-label-L* 及び *non-head-label-R* の 2 つで, どちらから係るかという方向 (L or R) も区別した. 左右の文脈長を (2,3) とし可変文脈長モデルでの実験結果を表 4 に示す. 表 4 で head-based はこれまでの主辞を基本とする素性を使用した場合で, non-head は新たに非主辞のラベルを素性として追加した場合の解析精度を表す.

非主辞情報を考慮した場合, 若干の精度向上が見られた. 今後より効果が得られるように, これら非主辞情報をどのように使用するかについて調査する必要がある.

5.3 関連研究との比較

本手法の有効性を示すために関連研究との比較を行った. 比較の対象としたのは Ratnaphakhi[7], Collins[2], Charniak[1] の 3 つの研究で, これらはいずれも Penn Treebank を使用して評価実験を行っている.

Ratnaphakhi は最大エントロピー法と上昇型アルゴリズムで解析木を構築する手法を提案している [7]. 解析木を構築する *shift-reduce* 法の手続きを, 最大エントロピー法によりスコア付けを行っている. 最大エントロピー法の学習に用いる素性は, 文脈からの情報を利用し様々な素性選択を行っている.

Collins は, 主辞の語彙情報を考慮した 3 つの確率モデルを提案している [2]. 最も高い解析精度のモデルは, 文法規則の適用確率を推定する場合, Penn Treebank に付与されている意味情報を利用し補語か否かを識別することで, 動詞の下位範疇化に関する情報を考慮している.

Charniak は最大エントロピー法を用いて, 上昇型及び下降型両方の解析アルゴリズムを使用し 2 段階で行う解析器を提案している [1]. 一段階目は, 上昇型チャートパーザを使用し, 解析木の候補を生成する. 二段階目ではそれら生成した候補に対し, 最大エントロピー法を使用し確率値を再推定し, 最も高い確率をもった解析木を出力する.

これら 3 つの研究は, 品詞タグ付けも含めた解析を行っている. 比較のため本稿でも, 中川らの提案する修正学習法よる手法 [6] を使用し, テストデータの品詞タグづけを行った. 結果を表 5 に示す. 比較には, 本手法で最も精度の高い, 非主辞情報も考慮した可変文脈長モデル (2,3) を用いた. 表 5 で (i) はテストデータに対する品詞を, Penn treebank の正解タグを使用した場合で, (ii) は中川らの手法を使用し, 品詞タグ付けを行った場合の解析精度である.

表 5 より, 構文解析の精度は品詞タグ付けの精度に大きく影響されることがわかる*. これは本手法は決定的に解析を行うため, 入力となる, 単語と品詞のペアに誤りがあった場合正しい解析ができない. 今後品詞の誤りに大きく影響を受けないように, 品詞タグに関する部分に曖昧性許すなどの対処が必要である.

我々の手法は, テストデータに対し品詞タグ付けを行った場合でも, Ratnaphakhi の結果と同等である. 彼

*中川らの手法による品詞タグ付け精度は 96.7%であった.

表 5: 関連研究との比較

	本手法		関連研究		
all sentences	(i)	(ii)	Rat99	Col00	Cha00
LR(%)	88.2	86.8	86.3	88.1	**89.6**
LP(%)	89.0	87.5	87.5	88.3	**89.5**
CB	1.03	1.15	–	1.06	0.88
0CB(%)	64.2	61.2	–	64.0	**67.6**
2CB(%)	85.0	83.1	–	85.1	**87.7**
length ≤ 40					
LR(%)	89.0	87.5	–	88.5	**90.1**
LP(%)	89.6	88.0	–	88.7	**90.1**
CB	0.87	0.98	–	0.92	0.74
0CB(%)	66.8	63.8	–	66.7	**70.1**
2CB(%)	87.2	85.4	–	87.1	**89.6**

の方法が, 様々な素性選択をしているのに対し, 我々は主辞や非主辞といった素性を多数使用しても, 精度は劣化しない.

Collins の手法と比較すると, 品詞タグ付けの精度が向上すれば, 彼の解析器と同等の精度が期待できる. また彼の方法で, 補語の判定に, Penn treebank の意味情報を利用しているため, このコーパスに強く依存する問題がある. 我々の手法は, できるだけ表層的な情報を素性として使用しているため, 他のコーパスでの学習, 解析も柔軟に対応できると考えている.

Charniak の手法は, 品詞タグ付けの精度に関わらず, 我々の手法を大きく上回っている. 彼の方法は上昇及び下降型の双方を効果的に使用しているためで, 結果的に本手法を含め 4 つ解析器で最も良い解析精度を得ている. 今後本手法にも, 部分的にでも 2 段階で行う解析手法の適用を検討する必要がある.

6 まとめ

本稿では, SVM を用いて英語構文解析を行うための手法を提案し, Penn treebank コーパスを使用した実験を行いその有効性について報告した. 解析木構築の過程を, 文脈情報から, 適切な解析木を構築する手続きへの分類問題とみなすことで, SVM による学習を行った. また学習したモデルを使用し, *shift-reduce* 法に類似した 3 つの手続きを用いて bottom-up に解析木を構築するアルゴリズムに適用し, 決定的に解析を行った.

Penn treebank コーパスを用いた実験では, 英語では, 右から左の要素を修飾する場合が多く, これを考慮するために右文脈を長めに考慮する可変文脈長モデルが有効であることがわかった. 関連研究との比較では, 品詞タグ付けの精度が高ければ, 表層的な情報のみを使用し, 特別な素性選択を必要することなしに, Ratnaparkhi, 及び Collins の方法と同等以上の精度が期待できることがわかった.

今後の課題

本手法は決定的に解析を行うために, その入力となる, 単語に対する品詞タグ付け精度に強く依存する. 品詞タグ付け精度に大きな影響を受けないように, 品詞に関しては曖昧性を考慮できるように拡張を行う必要がある.

今回可変文脈長モデルを簡単な制約により実現したが, よりよい文脈長について詳細に調査する必要がある. 更に非主辞情報を追加することで若干の精度向上が確認できたため, 非主辞情報を有効的に使える手法を検討する必要がある. また現在, 学習及び解析に多くの時間を必要とするため, 今後効率的な学習についても検討していきたい.

謝辞

修正学習法による品詞タグ付けの学習に関し, 様々な助言, 及びツールを提供して下さった中川哲治氏に心から感謝致します.

参考文献

[1] Eugene Charniak. A Maximum-Entropy-Inspired Parser. In *Proceedings of the Second Meeting of North American Chapter of Association for Computational Linguistics (NAACL2000)*, pages 132–139, 2000.

[2] Michael Collins. Three Generative, Lexicalised Models for Statistical Parsing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (jointly with the 8th Conference of the EACL)*, pages 16–23, 1997.

[3] E. Black et al. A Procedure for Quantitatively Comparing the Syntactic Coverage of English Grammars. In *Proceedings of the February 1001 DARPA Speech and Natural Language Workshop*, 1991.

[4] Hiroya Takamura et. al. Ensemble based on Feature Space Restructuring with Application to WSD. In *Proceedings of the Sixth Natural Language Processing Pacific Rim Symposium (NLPRS2001)*, pages 41–48, November 2001.

[5] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993.

[6] Tetsuji Nakagawa, Taku Kudoh, and Yuji Matsumoto. Revision Learning and its Application to Part-of-Speech Tagging. In *Proceedings of Association for Computational Linguistics (to appear)*, 2002.

[7] Adwait Ratnaparkhi. Learning to Parse Natural Language with Maximum Entropy Models. *Machine Learning*, 34(1-3):151–175, 1999.

[8] Kudo Taku and Yuji Matsumoto. Japanese Dependency Analysis Based on Support Vector Machines. In *Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 18–25, 2000.

[9] Vladimir N. Vapnik. *Statistical Learning Theory*. A Wiley-Interscience Publication, 1998.

[10] 山田 寛康, 工藤 拓, and 松本 裕治. Support Vector Machine を用いた日本語固有表現抽出. **情報処理学会論文誌**, 43(1):44–53, 2002.