

Support Vector Machine による英語係り受け解析

宮田 高志 山本 薫 松本 裕治

奈良先端科学技術大学院大学 情報科学研究科

{takashi,kaoru-ya,matsu}@is.aist-nara.ac.jp

日本語の「文節」に相当する単位を英語に導入することで句構造を文節間依存構造に変換し、これを Support Vector Machine で学習することにより係り受け解析器を作成した。「文節」の定義によって異なるが、解析精度として文単位で 10~13%、文節単位で 70~73% を得た。文節内の係り受けが全て正しいと仮定すると、これは単語単位で 80~85% の精度に相当する。また、精度に対する各素性の影響を調べ、文節の前後の単語などの周囲の特徴が精度に寄与していることを明らかにした。

[キーワード] 文節間依存構造解析, Support Vector Machine, 素性選択

English Dependency Parsing Based on Support Vector Machine

MIYATA Takashi, YAMAMOTO Kaoru, and MATSUMOTO Yuji

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

We have developed a chunk-based English dependency parser using Support Vector Machine. First, we use corpus of dependency structure among chunks which is translated from phrase structure provided in Penn treebank. Support Vector Machine is used to train to estimate if a given pair of chunks are in dependency relation or not. The parser segments into chunks, and calls the model trained by Support Vector Machine to determine if two chunks depend on each other. Our preliminary experiment shows that the parser achieves 10-13% accuracy on sentence-level, 70-73% on chunk-level. Assuming that all dependencies in chunks be correct, these results can be considered to correspond 80-85% in word-dependency accuracy. We also investigate impacts against parsing accuracy and find out that words before and after the chunk contributes our parsing performance.

[Keyword] Chunk-Based Dependency Structure Analysis, Support Vector Machine, Feature Selection

1 はじめに

近年、統計的学習アルゴリズムの一つである Support Vector Machine (SVM) [9] が様々な分野に応用され、その高い性能が注目されている。自然言語処理においても文書分類 [8] やチャンキング [6]・係り受け解析 [5] などにおいて、従来の統計的手法よりも少ない学習データで高い精度を達成できることや大量の素性を導入しても過学習を起しにくいことなどが報告されている。Hidden Markov Model や Maximum Entropy Model などの確率的モデルに基づく学習アルゴリズムとは異なり、SVM は次のよう

な性質をもつ。

1. 学習対象に対して事前に確率分布を仮定しない、すなわちノンパラメトリックな推定を行なう。
2. カーネル関数を使うことで、学習に必要な資源をさほど増やすことなく、大量の素性を扱うことができる。
3. カーネル関数を工夫することで素性間の依存関係を網羅的に扱うことができる。

1 は、(訓練事例に確率的なノイズの混入を許すようにモデルを拡張することは可能であるが) 原則として学習対象に対して決定的なモデルを仮定し、訓練

事例の頻度情報は使わないことを意味する。このため、訓練事例には「量より質」が求められる。2, 3はSVMの中心的な特徴であるカーネル法とよばれる計算上の工夫によって可能となっている。

一方、日本語において高い精度を達成している多くの統計的構文解析器[10, 11, 5]は句構造ではなく、文節単位の係り受けに基づいている。句ではなく文節という単位を使うことの利点は、二つの単位が関係するかどうかを決定する際に(文節が付属語を含んでいることから)より多くの情報が利用可能となること、文を大きな単位に分割することで組合せの数を減らせることなどがあげられる。注意深く文節を定義すれば、英語の構文解析においてもこのような効果を得られると期待できる。そこで我々は英語に対しても文節という単位を導入した。

論文の構成は以下の通りである: 2節では作成した文節係り受け解析システムの全体的な構成を説明し、学習時および解析時の処理の流れを概観する。3節ではSVMについて簡単に説明し、4節では文節の定義と実験で使用した素性について述べる。5節では二つの実験の結果について報告し、6節では関連研究との比較を行なう。最後に7節においてまとめと今後の課題について述べる。

2 全体の構成

2.1 学習

図1に学習時の処理の流れを示す。英語においては利用可能な係り受け解析済みコーパスがなかったため、学習データとしては句構造のタグがふられたPenn treebank (Wall Street Journal)を依存構造に変換して用いることとした。図1の各段階は次のように進む。

1. Collins [3]の主辞決定規則を用いて各句ごとに主辞を決め、主辞以外の語は全て主辞に係るとして句構造を語の依存木に変換する。
2. 正規表現によって記述した文節の定義に従って、文を文節に分割し(4.1節を参照)、文節中の主辞の係り先(を含む文節)をその文節の係り先とする。一つの文節が複数の主辞を含む場合は、もとの句構造においてより高い位置にあったもの(兄弟であれば左にあったもの)の係り先を優先する。これによって語の依存木を文節の依存木に変換する。

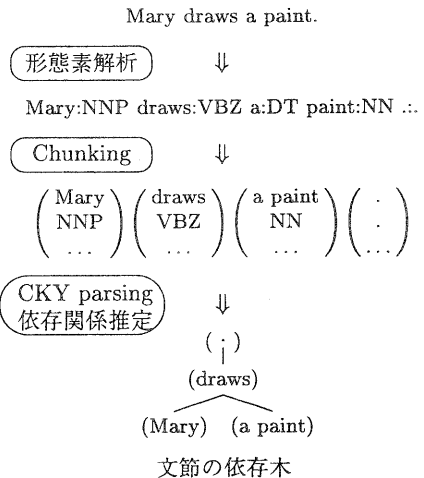


図2: 文節係り受け解析システムの解析時における処理の流れ

3. 一文中の全ての二文節について文節内および文節間の各種情報を素性ベクトルに変換し(4.2節を参照)、それらが係り受けの関係にあれば+1、そうでなければ-1というラベルをふって学習器へ渡す。学習器としてはSVM^{light} [4]を用いた。これは、与えられた素性ベクトルに対して、符号がクラスを、絶対値が確信度を表すような実数を返す分類器とみなせる。

文節の決定および素性ベクトルの生成にはPenn treebankに付与された品詞を使用した。

2.2 解析

図2に解析時の処理の流れを示す。解析時には、まず与えられた文を形態素解析して語の区切りと品詞および原形を求め¹、これを学習時と同じ定義に従って文節に区切る。次に文中の全ての二文節に対して、それらを素性ベクトルに変換したものを2.1節で作成した分類器に渡すことで、各文節対が係り受けの関係にあるかどうかを推定する。この推定をもとに次のようにして文全体の係り受けを決定する。

- 二つの文節 i, j の係りにくさを、文節自身の情報 $f_1(i), f_1(j)$ および両方から決まる情報(文節間の距離や間にある句読点の数など) $f_2(i, j)$ を

¹コーパス中の文を評価に使っているため、実際には予め付与されたタグを使う。

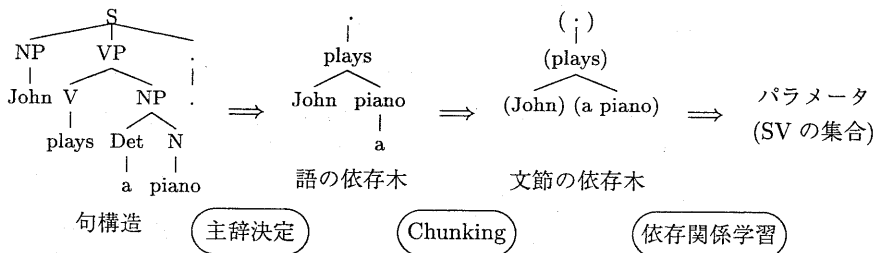


図 1: 文節係り受け解析システムの学習時における処理の流れ

使って計算されるコスト $C(i, j)$ で評価する。

$$C(i, j) = -\log N_i s(x(i, j)) \quad (1)$$

$$x(i, j) = c(f_1(i), f_1(j), f_2(i, j)) \quad (2)$$

$$s(x) = \frac{1}{1 + e^{-x}} \quad (3)$$

$c(\dots)$ は分類器が返す実数値であり、符号が分類のクラス (i.e. 文節 i が文節 j に係るかどうかな) を、絶対値が確信度を表す。本論文では SVM が返す分離平面からの距離をそのまま使った。 N_i は $\sum_j N_i s(i, j) = 1$ とするための正規化係数である。

- 文全体の係り受け D_{opt} は、(非交差・非循環的であるものの中で) 次のコストの和を最小にする係り受けであるとする。

$$D_{opt} = \arg \min_D \sum_{(i, j) \in D} C(i, j) \quad (4)$$

3 Support Vector Machine

SVM は n 次元ユークリッド空間上の事例を正例と負例に線形分離する、二値分類器である。すなわち、与えられた事例 $\vec{x} = (x_1, x_2, \dots, x_n)$ に対して式 (5) の符号によってそのクラスを判別する。

$$c(\vec{x}) = \vec{w} \cdot \vec{x} + b \quad (5)$$

ここで \vec{w} と b は学習によって決めるパラメータである。今、訓練事例が次のように与えられたとする。

$$(\vec{x}_1, y_1), \dots, (\vec{x}_l, y_l), \quad \vec{x}_i \in R^n, \quad y_i \in \{-1, +1\}$$

クラスの判別は式 (5) の符号だけで行なうので \vec{w} と b には定数倍の不定性があるが、式 (6) の制限を課すことで同じ判別を行なうパラメータが一意に決まるようになる。

$$\forall i : y_i(\vec{w} \cdot \vec{x}_i + b) - 1 \geq 0 \quad (6)$$

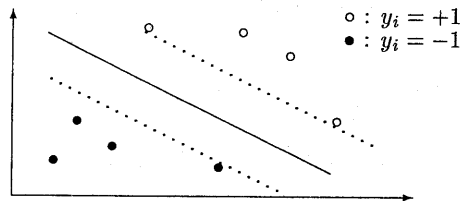


図 3: 分離平面とマージン

訓練事例と式 (6) の関係を示したのが図 3 である。図 3 において、実線が式 (5) を 0 とおいた分離平面 $\vec{w} \cdot \vec{x} + b = 0$ を表し、上側の点線が $\vec{w} \cdot \vec{x} + b = +1$ 、下側の点線が $\vec{w} \cdot \vec{x} + b = -1$ を表す。二つの点線上にある事例をサポートベクトルとよび、二つの点線の距離をマージンとよぶ。一般に原点から平面 $\vec{a} \cdot \vec{x} + b = 0$ までの距離は $|b|/|\vec{a}|$ で計算できるので、マージンは $2/|\vec{w}|$ となる。

SVM の学習アルゴリズムは式 (6) の条件を満たすパラメータの中で、マージンを最大にするものを求める。これは制約 (6) の下で $|\vec{w}|$ あるいは $|\vec{w}|^2$ を最小化することに帰着される。ラグランジュの未定乗数法を使うと式 (7) を Lagrangean として、制約 (8) と式 (9) および式 (10) の方程式が得られる。

$$L_P = \frac{1}{2} |\vec{w}|^2 - \sum_{i=1}^l \alpha_i \{y_i(\vec{w} \cdot \vec{x}_i + b) - 1\} \quad (7)$$

$$\forall i : \alpha_i \geq 0 \quad (8)$$

$$\frac{\partial L_P}{\partial \vec{w}} = \vec{w} - \sum_{i=1}^l \alpha_i y_i \vec{x}_i = 0 \quad (9)$$

$$\frac{\partial L_P}{\partial b} = \sum_{i=1}^l \alpha_i y_i = 0 \quad (10)$$

式 (9) から求めた \vec{w} および式 (10) を式 (7) に代入した式 (11) を新たな Lagrangean とみなすと、最終的

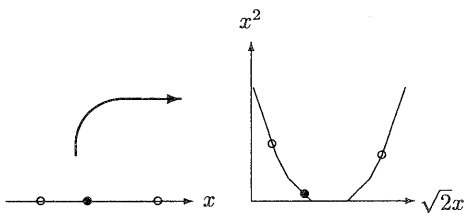


図 4: 高次元への写像の例

にマージンを最大化することは、制約 (8) の下で式 (11) を最大化することに帰着される。

$$L_D = -\frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j \vec{x}_i \cdot \vec{x}_j + \sum_{i=1}^l \alpha_i \quad (11)$$

同様に、クラスの判別に使う関数 (5) も式 (12) に書き直すことができる。

$$c(\vec{x}) = \sum_{i=1}^l \alpha_i y_i \vec{x}_i \cdot \vec{x} + b \quad (12)$$

以上の議論の結果として、学習のための目的関数 (11) および識別関数 (12) がともに事例を表すベクトルの内積 $\vec{x} \cdot \vec{x}'$ だけで計算できることに注意されたい。このことは次に説明する、事例を高次元へ写像する場合において重要な役割をはたす。

非線形の場合への拡張 線形分離できない事例に対しても、それらをより高い次元に写像することで線形分離できる場合がある。例えば 1 次元の場合、図 4(左) のような三つの訓練事例は式 (5) の識別関数では正例と負例を区別できない。しかし、写像

$$\psi: x \mapsto (x^2, \sqrt{2}x, 1) \quad (13)$$

によって 1 次元の事例を 3 次元²⁾ の事例に対応させる (図 4(右)) と、写像した先では線形な識別関数で正例と負例を区別できるようになる。ここで、目的関数 (11)・識別関数 (12) とともにベクトルの内積だけで計算できることに注意すると、高次元に写像した時の目的関数・識別関数はこれらに含まれている内積 $\vec{x} \cdot \vec{x}'$ を $\psi(\vec{x}) \cdot \psi(\vec{x}')$ で置き換えるだけで得られることがわかる。すなわち、写像した先での内積 $\psi(\vec{x}) \cdot \psi(\vec{x}')$ がもとの事例 \vec{x}, \vec{x}' から直接計算できれば、高次元の事例 $\psi(\vec{x}), \psi(\vec{x}')$ そのものを記憶したり操作したりする必要はない。逆にいえば、もとの事例 \vec{x}, \vec{x}' か

²⁾ 3 つ目の次元は常に 1 なので、実質的には 2 次元である。

```
punc := {".", ",", ":", ";"}; # マクロ定義
[pos=IN][pos=VBG][pos in punc]?; # 前置詞句
[lex/(hasn't|haven't|hadn't)/]
[pos=RB]*[lex="been"][pos=VBN][pos in punc]?;
# 完了形 受動態
```

図 5: 文節定義の例

ら何らかの実数を計算する関数 K に対して、式 (14) を満たす写像 ψ が存在すれば、式 (15) および式 (16) は ψ で写像した事例に対する線形分離問題の目的関数および識別関数になっているということになる。

$$\psi(\vec{x}) \cdot \psi(\vec{x}') = K(\vec{x}, \vec{x}') \quad (14)$$

$$L'_D = -\frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j K(\vec{x}_i, \vec{x}_j) + \sum_{i=1}^l \alpha_i \quad (15)$$

$$c'(\vec{x}) = \sum_{i=1}^l \alpha_i y_i K(\vec{x}_i, \vec{x}) + b \quad (16)$$

一般に式 (14) を満たす写像 ψ が存在するような関数 K をカーネル関数とよぶ³⁾。図 4 の場合は、 $\psi(\vec{x}) \cdot \psi(\vec{x}') = (x \cdot x' + 1)^2$ がカーネル関数となる。さらにこのカーネル関数は一般に $(\vec{x} \cdot \vec{x}' + 1)^d$ と書け、 d 次の多項式カーネル関数とよばれる。 d 次の多項式カーネル関数の特徴として、事例中の各次元 (素性) に対して d 個までの依存関係を考慮した判別ができることがあげられる。

4 実験

4.1 Chunking

学習時および解析時の文節区切りは、図 5 のような正規表現に従って行なわれる。例えば最後の規則は「表層形が正規表現/(hasn't|haven't|hadn't)/に合致する語の後に品詞が RB である語が 0 回以上繰り返し、その後に表層形が been である語と品詞が VBN である語がきて、最後に句読点が 0 個または 1 個ならんだものを一まとめにせよ」ということを意味する。文節区切りでは与えられた形態素列の先頭から各正規表現に合致するかどうかを確かめてゆき、最も長い形態素列に合致した正規表現を使って、形態素列の先頭から正規表現に合致した部分を文節として切り出す。残った部分について同様に文節を切り出してゆくが、どの正規表現にも合致しなかった

³⁾ 与えられた関数 K に対して式 (14) を満たすような写像 ψ が存在するための必要十分条件は Mercer 条件とよばれる。

時は先頭の一語だけで文節を作る。実験ではこのような規則を約 50 個用意することで、次のような単位を文節としてまとめた。

- 名詞連続:
'Black Monday', 'three months ago'
- 動詞群:
'will have to go', 's been'
- 数表現:
'22.6%', '1.3 million', '2:37p.m.'
- 前置詞句:
'on the results', 'after the 1987 crash,'

ただし、前置詞句の定義は係り受けを決定する上で次のような問題をひき起こす。例えば、“John played the piano after Mary sang.” のような文の場合、前置詞句の定義があると（正規表現の照合は最長一致で行なうので）‘after Mary’ で文節を構成してしまう。すると句構造では動詞 sang よりも接続詞 after の方が高い位置にあるために、「動詞が（接続詞を含んだ）主語に係る」という事例が学習されてしまう。この問題は Penn treebank では接続詞と前置詞が区別していないことに起因する。

また、句読点に関しても単純に最後の文節に含めるべきかどうかは自明でない。日本語の場合は文末の文節は係り先をもたないという特別な性質をもち、読点を文節に含めることでその文節が文末かどうかで区別できることや、句点がつくつかないかはその文節の係り先の決定に影響を及ぼすことなどから、文節に句読点を含めることは妥当である。しかし英語の場合、これらの句読点の「効果」は必ずしも句読点の直前の文節に帰すべきであるとは限らない。

文節の単位はできるだけ大きい方が解析には有利なので、以上の二つの影響を調べるために本論文では次の三種類の定義を使って実験を行なった。

文節 A 前置詞および句読点を文節に含める。

文節 B 前置詞と名詞句は分離する。

文節 C 前置詞と名詞句を分離し、句読点も独立した文節とする。

文節の範囲を決めた後、システムはその文節の「主辞」と「関係語」を同じように正規表現で記述した約 50 個の定義に従ってそれぞれ一単語ずつ選ぶ。ここで「主辞」とはその文節中で主要な語を意味する⁴。

⁴文節の単位が句ではないので、ここでは Collins の主辞決定規則は使えないことに注意されたい。

「関係語」とは日本語の付属語に相当するもので、最も左の前置詞である。なお、どの定義にも合致しなかった場合はその文節の最後の語が選択される⁵。

4.2 学習に利用する素性

SVM の特徴として「大量の素性を導入しても学習時間にはほとんど影響がなく、過学習を起こしにくい」ということが報告されている [8, 5]。ここでは日本語における先行研究 [10, 11, 5] を参考に次のような素性を使った。

- 文節内の特徴:
主辞、関係語、先頭の単語、末尾の単語、語数、句読点の数、引用符の数、冠詞の数、大文字のみの単語の数、小文字のみの単語の数、大文字で始まる単語の数、接続詞の数
- 周囲の特徴:
前後の文節の主辞・関係語、前後の単語、二つ前の単語、二つ後の単語
- 文節間の特徴:
距離（文節・単語単位）、句読点の数、引用符の数、冠詞の数、接続詞の数、wh 語の数

「語数」および「大文字のみの単語の数」「小文字のみの単語の数」「大文字で始まる単語の数」は ‘the New York Stock Exchange Composite’ のような比較的長い固有名詞を特徴付けるために導入した。これらの固有名詞は Wall Street Journal に特有のものが多く、特定の動詞と共起することが予想される。「接続詞の数」に関しては after や for など前置詞にも接続詞にも使われる 66 語のリストを用意し、このリストに載っている語がいくつ出現したかを用いる。この情報を使うことで 4.1 節で述べたような係り先の問題が緩和されることが期待される。「前後の文節の主辞・関係語」は [5] における動的素性と同様の効果を期待して導入した。英語においては係り先の候補が二方向になるために動的素性をそのまま実装することは困難であるが、多くの係り受けは隣接する文節同士であるため、単純に前後の文節の情報を使うことでもある程度の効果が期待できる。

これらの素性を分類器に渡す時は、「主辞」や「関係語」・「単語」とあるものについてはその表層形・原形・品詞の三つの素性を作る。さらに、全ての素性の値が 0 か 1 になるように、表層形と原形について

⁵実装の都合上このようにしたが、関係語に関しては合致しなかった場合は「関係語なし」とすべきであると思われる。

は Wall Street Journal の 23 番に含まれる全ての表層形 (約 8400) と原形 (約 6900) に番号をふり、その番号の素性を 1 にセットすることで素性に変換した。約 50 の品詞についても同様である。また「語数」については 0, 1, ..., 9, 10 以上の 11 段階、文節間の距離については -10 以下, -9, ..., 9, 10 以上の 21 段階、「数」とあるものについては 0, 1, 2 以上の 3 段階に分けてそれぞれ素性に変換した。これによって、一つの文節は $(8400 + 6900 + 50) \times 4 + 11 + 3 \times 7 + (8400 + 6900 + 50) \times 8 \approx 184000$ 次元の 0-1 ベクトルに変換される。分類器に渡される素性は文節二つ分の素性の他に文節間の素性 $21 \times 2 + 3 \times 5 = 57$ が加わり、約 368000 次元 (このうち 1 であるのは高々 95ヶ所) の 0-1 ベクトルになる。

5 結果と考察

5.1 文および文節単位の精度

実験は、Wall Street Journal の 23 番に含まれる文のうち最初の 600 文を 100 文ずつ 6 つに分割し、500 文を訓練事例として使い、残りの 100 文で評価するということを 6 回行なった (6-fold cross validation)。SVM の設定は、カーネル関数として 3 次の多項式 (3 節参照) を使い、エラーの重要度は 1 とした⁶。表 1 に 6 回の評価の平均を示す。表 1(上) および表 1(中) における文節単位の精度は、Charniak や Collins らの先行研究 [1, 2] と直接比較することはできないが、かりに文節内の単語係り受けが全て正しいと仮定した時の精度が表 1(下) である。文節の定義は十分単純なので、この仮定は妥当であると思われる。

三つの文節定義による精度を比較すると、文節 B と文節 C にはあまり大きな違いはみられない。これは「文節直後の単語」という素性を使っているからだと思われる。一方、訓練事例の数は文節 A・文節 B・文節 C でそれぞれ約 6 万, 10 万, 14 万 (500 文 6 セットの平均) であり、学習に必要とされる計算時間は訓練事例数の線形以上で増加するので、5.2 節の実験では文節 B を使った。

⁶式 (6) を式 (17) のように変更して、 $|w|^2$ ではなく、 $|w|^2 + C \sum_i \xi_i$ を最小化することになると、各事例についてある程度マージンの中へ入り込むことを許容しつつマージンを最大化するように問題を拡張することができる。

$$v_i : y_i(\bar{w} \cdot \bar{x}_i + b) - 1 + \xi_i \geq 0 \quad (17)$$

この時の C がエラーの重要度を表し、小さな値に設定することでエラーよりもマージンの方を重視して学習するようになる。

表 1: 三種類の文節定義に対する精度

訓練事例:					
文節 A		文節 B		文節 C	
文	文節	文	文節	文	文節
90.78	98.71	88.95	98.79	88.35	98.92

評価事例:					
文節 A		文節 B		文節 C	
文	文節	文	文節	文	文節
12.79	69.51	10.36	73.54	10.78	73.51

単語単位の精度 (評価事例):

文節 A	文節 B	文節 C
84.76	83.31	80.14

(文節内の係り受けが全て正しいと仮定)

5.2 精度に対する各素性の影響

各素性が精度に及ぼす影響を調べるために、分類器に素性ベクトルを与える際に特定の素性を (係り側・受け側とも) 省略して精度をはかるという実験を行なった。厳密には学習もその素性を取り除いて行なうべきであるが、およその目安にはなると思われる。精度が顕著に下がった素性は「主辞」「関係語」「先頭の単語」「末尾の単語」「直前の文節の主辞」「直前の単語」「直後の文節の主辞」「直後の単語」「二つ後の単語」および「二文節間の距離」である。その他の素性についてはむしろ精度が上がっていることから、これら以外の素性は不要だったことが疑われる。興味深いのは「二つ前の単語」は精度に悪影響を及ぼしているのに対し「二つ後の単語」は精度の向上に貢献している点である。これは英語では多くの文節が後ろから前へ係るということと関係していると推測される。また、日本語では有効であるとされている文節中および二文節間の句読点の有無が、この実験では精度に対してあまり影響がなかった。さらに二種類の二文節間の距離のうち、単語単位ではかかる方が精度に及ぼす影響が大きい。実際、この実験で用いた素性の中でもっとも影響が大きいといえるが、これがどういう意味をもつのかは今後の検討課題である。

6 関連研究との比較

英語の統計的構文解析においては、Collins [3, 2], Ratnaparkhi [7], Charniak [1] らが labeled recall/precision で 87~90% を達成している。Collins

表 2: 解析時に素性を省略した場合の精度

省略した素性	文	文節
省略なし	10.36	73.54
主辞	9.09	69.88
関係語	9.08	70.22
先頭の単語	9.23	72.06
末尾の単語	9.09	70.50
語数	10.89	73.96
句読点の数	12.52	74.64
引用符の数	11.81	74.68
冠詞の数	12.54	74.84
大文字のみの単語の数	12.33	74.89
小文字のみの単語の数	11.81	73.96
大文字で始まる単語の数	12.00	74.42
接続詞の数	12.52	74.21
直前の文節の主辞	6.87	70.69
直前の文節の関係語	10.36	73.54
直前の単語	6.87	69.91
二つ前の単語	11.59	73.61
直後の文節の主辞	8.54	70.51
直後の文節の関係語	10.36	73.54
直後の単語	8.34	70.17
二つ後の単語	9.27	72.17
二文節間の距離 (文節単位)	11.24	70.89
二文節間の距離 (単語単位)	5.62	63.63
二文節間にある句読点の数	11.40	72.99
二文節間にある引用符の数	12.31	74.38
二文節間にある冠詞の数	12.66	74.14
二文節間にある接続詞の数	10.32	73.48
二文節間にある wh 語の数	12.49	74.21

は句の間の依存確率モデルを提案しており、我々の係り受けモデルに近いが、句構造における統語範疇(木構造における中間ノード)を用いる点が異なる。我々の係り受けモデルでは二つの文節に係るか係らないかを区別するだけで、統語範疇に相当する情報は係り側の文節に含まれる(もしくは係り側の文節に隣接する)関係語によって補うという方針をとっている。Ratnaparkhi は多階層のチャンキングを行なうことで木構造を構築している。各階層のチャンキングは Maximum Entropy Model に基づいている。

確率的モデルに基づく方法ではデータの過疎性に対処するために何らかの smoothing を施す必要がある。Charniak [1] が指摘しているように、この点で

Maximum Entropy Model は有利であるが、どの確率を smoothing するかは人手で決めなければならない、網羅性という点では [5] や我々のように SVM を使う方が優れている。ただし、今回の実験からわかるように、SVM でも不要な素性の棄却までは行なえないので、素性自身の吟味は必要である。

7 まとめと今後の課題

本論文では、SVM に基づいた英語の文節係り受け解析器を作成し、文単位で 10~13%、文節単位で 70~73% の精度を得た。文節内の係り受けが全て正しいと仮定すると、これは単語単位で 80~85% の精度に相当する。また、各素性の影響を調べ、文節の前後の単語などの周囲の特徴が有効であることを明らかにした。

今後は二つ以上の素性の相互作用やそれらがどのような情報を捉えていることで精度を向上させているのかについて、詳細に検討したい。また [5] における動的素性に関してもある程度制限した形でならば解析に組み入れることは可能であると思われる。

謝辞

アルゴリズムや実装において有益な助言をいただいた、奈良先端大学院大学情報科学研究科の工藤拓氏に感謝します。

参考文献

- [1] Eugene Charniak. "A Maximum-Entropy-Inspired Parser". In *Proceedings of the 6th Applied Natural Language Processing Conference (ANLP) and the 1st Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pp. NAACL 132-139, USA, April-May 2000.
- [2] Michael Collins. "Three Generative, Lexicalised Models for Statistical Parsing". In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and the 8th Conference of the European Chapter of the Association for Computational Linguistics (ACL-EACL)*, pp. 16-23, Spain, July 1997.
- [3] Michael John Collins. "A New Statistical Parser Based on Bigram Lexical Dependence". In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and the 8th Conference of the European Chapter of the Association for Computational Linguistics (ACL-EACL)*, pp. 24-31, Spain, July 1997.

- cies". In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 184–191, Santa Cruz, California, USA, June 1996.
- [4] Thorsten Joachims. "Making Large-Scale Support Vector Machine Learning Practical". In Bernhard Schölkopf, Christopher J. C. Burges, and Alexander J. Smola, editors, *Advances in Kernel Methods — Support Vector Learning*, chapter 11, pp. 169–184. MIT Press, 1999.
- [5] Taku Kudo and Yuji Matsumoto. "Japanese Dependency Structure 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*, pp. 18–25, Hong Kong, October 2000.
- [6] Taku Kudo and Yuji Matsumoto. "Use of Support Vector Learning for Chunk Identification". In *Proceedings of the 4th Conference on Computational Natural Language Learning (CoNLL-2000) and the 2nd Learning Language in Logic Workshop (LLL-2000)*, pp. 142–144, Lisbon, September 2000. (Shared Task Session).
- [7] Adwait Ratnaparkhi. "A Linear Observed Time Statistical Parser Based on Maximum Entropy Models". In *Proceedings of the 2nd Conference on Empirical Methods in Natural Language Processing*, pp. 1–10, USA, August 1997.
- [8] Hirotoshi Taira and Masahiko Haruno. "Feature Selection in SVM Text Categorization". In *Proceedings of the 16th National Conference on Artificial Intelligence (AAAI'99) and the 11th Conference on Innovative Applications of Artificial Intelligence (IAAI'99)*, pp. 480–486, Florida, July 1999.
- [9] Vladimir Naumovich Vapnik. *The Nature of Statistical Learning Theory*. Statistics for Engineering and Information Science. Springer-Verlag, 2nd edition, December 1999.
- [10] 藤尾正和, 松本裕治. "語の共起確率に基づく係り受け解析とその評価". 情報処理学会論文誌, Vol. 40, No. 12, pp. 4201–4212, December 1999.
- [11] 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". 第五回年次大会 併設ワークショップ「構文解析 — 現状の分析と今後の展望 —」論文集, pp. 41–48, 電気通信大学, March 1999. 言語処理学会.