論文

チャンキングの段階適用による日本語係り受け解析

Japanese Dependency Analysis using Cascaded Chunking

工藤 拓 (奈良先端科学技術大学院大学 情報科学研究科)
[学生会員#200100751]

松本 裕治 (奈良先端科学技術大学院大学 情報科学研究科)
[正会員#7922681]

連絡先
〒630-0101 奈良県生駒市高山町8916-5 奈良先端科学技術大学院大学 情報科学研究科 自然言語処理学講座
TEL: 0745-72-5246 FAX: 0743-72-5249
email:taku-ku@is.aist-nara.ac.jp

概要

本稿では, チャンキングの段階適用による日本語係り受け解析手法を提案し, その評価を行う. 従来の係り受け解析は, 任意の二文節間の係りやすさを数値化した行列を作成し, そこから動的計画法を用いて文全体を最適にする係り受け関係を求めるというモデルに基づくものが多かった. しかし, 解析時に候補となるすべての係り関係の尤度を計算する必要があるため効率が良いとは言えない. 本提案手法は, 直後の文節に係るか係らないかという観点のみで決定的に解析を行うため, 従来方法に比べ, モデル自身が単純で, 実装も容易であり, 高効率である. さらに, 従来法では, 個々の係り関係の独立性を前提としているが, 本提案手法はその独立性を一部排除することが可能である. 本提案手法を用い, 京大コーパスを用いて実験を行った結果, 従来法と比較して効率面で大幅に改善されるとともに, より高い精度を示した.

Abstract

In this paper, we propose a cascaded chunking method for Japanese dependency structure analysis. Conventional approaches mainly consist of two steps: First, the dependency matrix is constructed, in which each element represents the probability of a dependency. Second, an optimal combination of dependencies are determined from the matrix. However, such a method is not always efficient since it needs to calculate all the probabilities of candidates. Our proposed model is more simple and efficient, since it parses a sentence deterministically only deciding whether the current segment modifies segment on its immediate right hand side. In addition, proposed model does not assume the independence constraint in dependency relation. Experiments using the Kyoto University Corpus show that the method outperforms previous systems as well as improves the parsing and training efficiency.

1. はじめに

日本語の構文解析において係り受け解析は, 自然言語処理の基本技術の一つとして認識されており, 従来から多くの研究が行われている. 初期の研究では, 二文節間の係りやすさを決定するルールを人手で作成していたが, 網羅性, 一貫性という面で問題が多い. 近年では, 構文情報が付与された大規模コーパスが利用可能になったことで, 機械学習アルゴリズムを用いた統計的な構文解析技術が提案されるようになってきた.

従来の統計的係り受け解析は, 文中の任意の二文節間の係りやすさを数値化した行列を作成し, その中から動的計画法を用いて文全体を最適にする係り受け関係を求めるというモデルに基づくものが多かった. しかしながら, 解析時や学習時にすべての係り関係の候補を対象としなければならないために, 効率が良いとは言えない. さらに, 各二文節の係り関係は他と独立と仮定しているが, ある係り受け関係が他の係り受けに影響を及ぼすこともあり, この仮定は必ずしも適切ではない.

本稿では, チャンキングの段階適用による係り受け解析モデルを提案する. 提案するモデルは, 文節がその直後の文節に係るか係らないかという観点のみで決定的に解析を行うため, 従来法に比べ高効率である. また, チャンキングを採用することで, 解析精度を損うことなく, 一文あたりに用いる学習データの量を減らすことに成功した. SVM といった学習データ数に対し非線型に計算量が増加するような学習モデルではこれまで学習が事実上困難であったが, これによりある程度大量のデータからの学習が可能なった. さらに, チャンキングの導入により, 係り関係そのものを素性として導入することができ, 他の係り関係は他とは独立としていた従来手法に比べ精度向上につながった.

本稿で提案する係り受け解析も, 実際の係り関係の同定に機械学習アルゴリズムを用いる. 提案手法は, 機械学習アルゴリズムに依存しない手法であるが, 本稿では, Support Vector Machine （SVM）[1] を用いる. その理由として, SVM は, 他の学習モデルと比較して極めて汎化能力が高く , 高次元の素性集合を用いても過学習しにくいとされていこと. さらに, Kernel 関数を変更することで, 非線形のモデル空間を仮定したり, 複数の素性の組み合せを考慮した学習が可能な点が挙げられる. 自然言語処理においては, 文書分類や, 係り受け解析, 英語の単名詞句同定などに応用され高い精度を示している[2)~6)].

本稿の構成は以下の通りである. 2 章で従来法と提案手法の違いを述べ, 3 章で SVM の説明を行う. さらに 4 章で京大コーパスを用いた評価実験を提示し, 最後に 5 章で本稿をまとめる.

2. 係り受け解析モデル

2.1 係り受け解析モデル（従来法）

はじめに, これまでに日本語係り受け解析によく用いられているモデルについて説明する. まず, あらかじめ文節にまとめられ属性付けされた文節列 $\{b_1, b_2, \ldots, b_m\}$ を B, 係り受けパターン列 $\{Dep(1), Dep(2), \ldots, Dep(m–1)\}$ を D と定義する. ただし,$Dep(i)$ は, 文節 b_i の係り先文節番号を示す.

これ以降, D は以下の制約を満たすものと仮定する.

(1) 文末を除き, 各文節はその文節の後方側に必ず一つの係り先を持つ.

(2) 係り受け関係は交差しない.

統計的係り受け解析とは, 上記の二つの制約のもとで, 入力文節列 B に対する条件付き確率 $P(D|B)$ を最大にする係り受けパターン列 D を求めることと定義できる.

$$D_{best} = \arg\max_{D} P(D|B)s$$

ここで, それぞれの係り関係は独立であると仮定すると, $P(D|B)$ は,

$$P(D|B) = \prod_{i=1}^{m-1} P(Dep(i)=j \,|\, \mathbf{f}_{ij})$$
$$\mathbf{f}_{ij} = \{f_1, \ldots, f_n\} \in \mathbf{R}^n$$

のように変形できる. ここで, 確率 $P(Dep(i)=j \,|\, \mathbf{f}_{ij})$ は文節 b_i と文節 b_j が言語的素性集合 $\mathbf{f}_{ij}$ を持つ時に, 文節 b_i が文節 b_j に係る確率を示す. $\mathbf{f}_{ij}$ は文節 b_i と文節 b_j に関する種々の言語的特徴を表わす n 次元の特徴ベクトルである. 最終的に, これらの確率値をもとに D_{best} を決定する. この時, 確率 $P(Dep(i)=j \,|\, \mathbf{f}_{ij})$ を格納する行列は**係り受け行列**と呼ばれている. 従来法は, 与えられた入力文に対し, 係り受け行列を作成する処理と, 係り受け行列から最尤の係り受け候補選択し出力する処理の, 基本となる二つの処理から構成される. 確率 $P(Dep(i)=j \,|\, \mathbf{f}_{ij})$ を推定する手法としては, 語の共起確率を用いる手法[7], 決定木を用いる手法[8], 最大エントロピー法を用いる手法[9], Support Vector Machine を用いる手法[6] などがこれまでに提案されている.

従来手法は, 係り受け確率の作成に $O(n^2)$ (n は文節の個数) の計算量が必要となる. さらに, 係り受け行列から最尤の係り受け候補を全探索した場合, 全体として $O(n^3)$ の計算量となる. ビームサーチを行なった場合でも最低 $O(n^2)$ の計算量となる.

2.2 チャンキングの段階適用による解析（提案法）

チャンキングの段階適用による構文解析は, 英語の統計的構文解析においては古くから適用されてきた[10),11)]. このアルゴリズムのおおまかな流れは以下のようになっている.

(1) 入力として基本句列を与える.
(2) 文を先頭から眺めて行き, 任意の基本句の連続を, 新しい非終端ノードをまとめ上げる（チャンキング）.
(3) まとめ上げられた句の連続から, 主辞のみを残し, それ以外は削除する.
(4) 非終端ノードが一つになれば終了, それ以外は 2 に戻る

チャンキングはチャンクの状態を示すタグ付与することと解釈できるため, このモデルは基本句に対するタグ付与を段階的に適用した形になっている.

ここで, チャンキングの段階適用手法を日本語係り受けに適用することを考える. 日本語の係り受け構造は, 後方参照であること, 三つ以上の文節で一つの非終端ノードを形成しないことを考慮すれば, アルゴリズム以下のようになる.

(1) 入力文節すべてに対し, 係り受けが未定という意味の O タグを付与する.
(2) 文末の文節を除く O タグが付与された文節に対し, 直後の文節に係るか推定する. 係る場合は その文節に D タグに置き換える. 後ろから 2 番目の文節は無条件に D タグ に置き換える.
(3) O タグが付与された文節の直後にある文節のうち, D タグが付与されている文節をすべて削除する. 削除できる理由としては, 非交差条件から削除される文節は他から修飾される事は無く, それ自身の係り先も既に同定されているために, 係り受け候補として考慮する必要が無くなるためである.
(4) 残った文節が一つ（文末の文節）の場合は終了, それ以外は 2. に戻る.

このアルゴリズムによる解析例を**図 1** に示す.

提案法手法の計算量は, 最悪の場合に $O(n^2)$ となる. しかし, 多くの文節が直後の文節に係ることを考えると, 提案手法の計算量は, 実際にはこれより小さくなる.

図 1　係り受け解析例

Fig. 1　Example of dependency structure analysis

2.3　学習事例の抽出方法

従来手法では, 学習データ中のすべての二文節の候補を学習事例として抽出していた. このような抽出方法では, 学習データを不必要に多くしてしまい, 学習の効率が悪い.

本稿で提案するチャンキングの段階適用による係り受け解析では, **図 2** に示すように, テストと学習が, 同じ「解析」という処理から継承されて実装される. 唯一の相違点は, 学習時には学習コーパスを参照するのに対し, テスト時には機械学習の結果から推定することにある. 学習事例（係る事例, 係らない事例のペア）の抽出は, あたかも解析を行ないながら実際には学習コーパスを参照する形として実現される. 多くの文節が直後の文節に係る事を考慮すると, 学習事例の数を必要最小限に抑える事ができ, 学習時間が事例数に非線形に増加するような学習モデルでもある程度大量の学習データを扱う事が可能である.

2.4　静的素性と動的素性

統計的日本語係り受け解析に有効とされてきた素性には, 着目している 2 文節の主辞の語彙や品詞, 語形の活用形, 二文節間の距離, 句読点, 引用符の有無などがある. これらの素性は文節の作成時に決定される素性であり, このような素性集合のことをまとめて**静的素性**と呼ぶこととする. 日本語の係り受け関係

図 2　学習とテストの処理

Fig. 2　Process of training and testing

は, 文節に含まれる機能表現が大きな制約となり, 静的素性だけで係り先の大部分を限定することができる. しかし, 複数の係り先の候補がある場合や, 並列構造, 複文構造の場合, 個々の係り受け関係の従属性を考慮しないと係り関係を決定しにくいことがある.

内元らは, 係り関係を「係る」「越える」「手前」と 3 つに分けることで係り関係の従属性を部分的に考慮するモデルを構築している[12]. しかし, 係り受け確率は,「係る」「超える」「手前」といった係り元と係り先の位置関係で決定されるため, 3 つに分割するだけでは, 係り先が係る文節や, 係り先/係り元に係る文節といった関係は考慮できない. また, 一つの係り関係を決定するのに, 候補となるすべての係り受け確率を計算する必要があるため, 効率が悪い.

金山らは, 文法を用いて係り先の候補を絞り, それらの候補を確率の条件部に入れるモデルを提案している[13]. このモデルは, 係り関係の曖昧性を解消するのに効果的な手法である. しかし, 金山らも, 係り先が係る文節や, 係り先/係り元に係る文節といった関係を考慮するに及んでいない.

我々は, 解析途中に得られた係り関係そのものをフィードバックし, 素性として追加する **動的素性** の概念を取りいれ, 精度向上に成功している[6]. しかし, 文献 6) では, 動的素性を従来法の係り受けモデルの拡張として取り入れ, 実際の解析には, 関根の提案する文末から解析するアルゴリズム[14] を用いている. そのため, 動的素性として着目している二文節より後方にある文節しか対象にすることができないという問題点がある.

ここで, 我々は, 動的素性の概念をチャンキングの段階適用による係り受け解析に適用することを考える. 提案手法は, 文頭から解析し, 係り関係の推定と解析を同時に行っていくため, 自然な形で動的素性を投入する事ができる. ただし, 基本的にボトムアップに解析していくために, 動的素性として, 着目してい

図 3　三つの動的素性

Fig. 3　Three types of dynamic features

る二文節よりスコープの大きい係り関係は投入できない.

具体的に, 我々は動的素性として以下の三つの素性を考慮する（**図 3**）.

(1)　着目している係り先に係る文節 （A）

(2)　着目している係り元に係る文節 （B）

(3)　着目している係り先が係る文節 （C）

内元や金山らの手法と異なる点は, 上記の (A),(B),(C) といった着目している係り関係以外の係り関係を考慮できる点にある. これらの素性は, 解析中, 二つの係り関係が決まった時点で互いに素性情報を提供しあうことで容易に実現することが可能である.

2.5　従来法との比較

我々は, 提案するチャンキングの段階適用による係り受け解析は, 従来法と比較して以下のような利点を持つと考える.

- モデルの効率性

  従来法の計算量は, CYK 等で全探索を行なうと最悪 $O(n^3)$, また, そこからビームサーチを行なうと $O(n^2)$ となる. 一方, 提案法は, 最悪の場合に $O(n^2)$ である. しかし, 多くの文節が直後の文節に係ることを考えると, 計算量は実際にはこれより小さくなり, 線形時間に近づいていく. また, 学習においても, チャンキングを適用することで, 一文あたりの学習データ数を減らすことができ, SVM のように計算量が事例数に対し非線形に増加するような学習アルゴリズムでも比較的大量の学習データを利用できるようになる.

- 動的素性の考慮

  従来研究にも, 係り関係の従属性を部分的に考慮するモデルは存在したが, 着目する 2 文節以外の係り関係を考慮するには及んでいなかった. 提案手法では, 段階的に係り受け解析を行いながら, すでに同定が終わった係り受け関係を新たな素性として動的に追加することにより, 着目する 2 文節以

外の係り関係を考慮することができる.

- 非交差条件の自動考慮
  従来法にはモデル自身に非交差条件を考慮する仕組みが備わっていない. 実際の解析時に, 交差する条件を逐次考慮する必要があり, それらの処理は各パーザに依存する. 一方, 提案手法は, それ自身簡潔であるにもかかわらず, モデル自身が非交差条件を考慮する機能を備えている.
- 機械学習アルゴリズムの非依存性
  提案手法は, 直後の文節に係るか係らないかという観点で決定的に解析を行うため, 二値分類が行える学習器であれば, あらゆるものが適用可能である. 係りやすさの尤度や確率値は本提案手法には必ずしも必要ではない.
- 係りタイプの考慮
  従来法は, 係り受け解析に特化した手法であり, そのままの形では係り受けのタイプ （通常の係り受け, 並列, 同格 等） を出力しにくい. 本提案手法は, 係りタイプに応じて使用するタグを使い分けるだけでよく, アルゴリズムそのもに手を入れる必要はない.

3. Support Vector Machines

本章では, 係り受けの学習に用いた機械学習アルゴリズム Support Vector Machines (SVM)[1] について簡単な説明を行う.

二値分類問題において, 正例, 負例 の 2 つのクラスに属す学習データのベクトル集合を,

$$(\mathbf{x}_i, y_i), \ldots, (\mathbf{x}_l, y_l) \qquad \mathbf{x}_i \in \mathbf{R}^n,\ y_i \in \{+1, -1\}$$

とする. ここで $\mathbf{x}_i$ はデータ i の特徴ベクトルで, 一般的に n 次元の素性ベクトル ($\mathbf{x}_i = (f_1, f_2, \ldots, f_n) \in \mathbf{R}^n$) で表現される. y_i はデータ i が, 正例（1）, 負例（-1）かを表わすスカラーである.

SVM は, 以下のような n 次元 Euclid 空間上の平面で正例, 負例の分離を行う.

$$\mathbf{w} \cdot \mathbf{x} + b = 0 \qquad \mathbf{w} \in \mathbf{R}^n, b \in \mathbf{R}$$

$\mathbf{w}, b$ が, 求めるべきパラメータとなるが, SVM では, 近接する正例と負例の間の間隔（**マージン**）を最大化する戦略 (マージン最大化) によって, これらを導出する. 具体的な証明等は文献 1) に譲るが, マージン最大化は, 以下の制約付き最

小化問題で実現される[☆].

$$
\begin{aligned}
&Minimize: \quad L(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 \\
&Subject\ to: \quad y_i((\mathbf{w}\cdot\mathbf{x}_i)+b) \geq 1 \quad (i = 1 \ldots l)
\end{aligned}
$$

さらに, 一般的な分類問題においては, 学習データを線形分離することが困難な場合ある. このような場合, 各素性の組み合わせを考慮し, より高次元な空間に学習データを写像すれば線形分離が容易になる. 実際の証明は省略するが SVM の学習, 分類アルゴリズムは事例間の内積しか使用しない. この点を生かし, 各事例間の内積を適当な Kernel 関数におきかえることで, SVM は低次元中の非線形分類問題を高次元中の線形分離問題としてみなし分類を行う事が可能となっている. 多くの Kernel 関数が提案されているが, 我々は以下の式で与えられる d 次の多項式 Kernel 関数を用いた.

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^d$$

d 次の 多項式 Kernel は d 個までの素性の組み合わせ (共起) を考慮した学習モデルに帰着できる.

4. 実験および考察

4.1 実験環境と設定

実験に用いたコーパスは京大コーパス（Version 2.0）[15] の一部で, 学習には 1 月 1 日から 8 日分（7958 文）, テストには 1 月 9 日分の （1246 文）を用いた. さらに, 大規模な学習データを用いる実験として, 京大コーパス (Version 3.0) を使い, 二分割交差検定を行なった. 実験には SVM 学習ツール *Tiny*SVM を用いた[☆☆]. Kernel 関数には多項式 Kernel を用い, 次元数は文献 6) と共通にするため, 3 に固定した.

次に, 学習に用いた静的素性を, **表 5** に示す. これらは 若干な差異はあるものの[6),8),9),12)] 等で用いられた素性であり, 日本語係り受け解析に用いられる素性として一般的なものである. ただし, **主辞**とは文節内で品詞が特殊, 助詞, 接尾辞となるものを除き, 文末に一番近い形態素, **語形**とは文節内で品詞が特殊となるものを除き, 文末に一番近い形態素のことを指す. また, 見出し語の選択に関し

☆ 実際の実験では多少の解析誤りを認める Soft Margin の項を追加したモデルを用いている

☆☆ http://cl.aist-nara.ac.jp/~ taku-ku/software/tinysvm

ては適当な頻度による閾値を設けず, 学習データ中のすべての語彙を用いた.

次に, 動的素性の取り扱いについて説明する. タイプ A, B の動的素性には, 文節内の機能語の部分を一つの表現に縮約した表現を用いた. 具体的には, 助詞, 副詞, 連体詞, 接続詞 については見出し語そのものを, 活用形のあるものはその活用形を, その他の品詞については, 品詞と品詞細分類を与えた. また, タイプ C の動的素性には, 主辞の品詞と品詞細分類を与えた.

4.2 実 験 結 果

我々の提案手法と, 従来方法[6] の結果を**表 1** にまとめた. ただし, 係り受け正解率とは, 文末の一文節を除くすべての文節に対して, 正しく係り先が同定できたものの割合, 文正解率とは, 文全体の解析が正しいものの割合を示す. 大規模な学習データによる実験結果を**表 2** にまとめた.

結果, 係り受け正解率, 文正解率ともに, 従来方法に比べ精度が向上している. 効率面でも, 学習に要する時間が, 2 週間程度という非現実的な時間から, 10 時間程度と実用に耐えうる時間に短縮され, 一文あたりの平均解析時間は, 2.1 秒程度から, 0.5 秒程度に改善された☆. 実際の解析精度の低下がないことから, 提案手法は, 従来法に比べ高効率であるといえる.

また, 京大コーパス 3.0 を用いた大規模な実験においても, 現実的な時間で学習が終了した. 従来法では, このような大規模な学習データを扱うことは困難で, 表 1 で示した結果以上の精度を達成することは事実上困難であった. しかし, 本提案手法により, 大規模なデータを扱うことが可能となり, 結果として 90.46% という高い精度を示す結果となった.

4.3 動的素性の効果

表 3 に, 動的素性のうちそのいくつかを削除した場合の解析精度を示した. 結果から, どのタイプの動的素性も係り受け解析に有効であることが分かる.

さらに, **図 4** に, 動的素性を用いた場合と用いなかった場合の精度と学習データのサイズの関係を示した. 学習データが少量の時は, 動的素性は有効に機能していない. これは, 学習データが少量の時は, 動的素性が過学習の要因として働き, 十分に汎化が行われていないことを意味している. しかし, 学習データを増加させるとともに, 動的素性の効果は次第と大きくなっている. 学習曲線から考察するに, 学習データをさらに増やすと, 精度はさらに向上し, 動的素性の有無

☆ 学習は AlphaServer 8400 上で, 解析は PentiumIII（1GHz） の Linux 上で行った.

表 1　実験結果

Table 1　Results of our experiments

提案法	係り受け正解率	89.29% (10057/11263)
	文正解率	47.53% (589/1239)
	学習事例数	110355
	学習時間	約 10 時間
	解析時間	0.5 秒/文
従来法	係り受け正解率	89.09% (10034/11263)
	文正解率	46.17% (572/1239)
	学習事例数	459105
	学習時間	約 2 週間
	解析時間	2.1 秒/文 (beam 幅 1)

表 2　大規模データによる実験結果

Table 2　Results using large amount of training data

係り受け正解率	90.46 %
文正解率	53.16 %
学習事例数	261254
学習時間	約 2 日
解析時間	0.7 秒/文

による精度差がさらに大きくなると予想される.

4.4　素性の頻度と精度

我々は, 低頻度の素性を故意に削除することは行わず, 頻度 1 のものも含めすべての素性集合を使用した. 一方, 最大エントロピー法を学習アルゴリズムとして採用している過去の研究[12] では, 低頻度の素性を削除することが一般的に行われている. しかし, どの程度の頻度の素性を削除するといった明確な選択基準は提案されておらず, その多くは発見的に設定されてる.

また, 春野らは, 学習アルゴリズムとして決定木を用いた係り受け解析において, 語彙情報を利用すれば必ず解析精度が向上するという期待は成立しないと述べている[8]. しかし, 品詞や活用形が同一にもかかわらず係り関係が異なる例

表 3　動的素性の効果

Table 3　Effect of dynamic features

削除した動的素性	解析精度 (精度の増減)	
	係り受け正解率	文正解率
A	89.01% (−0.28%)	46.64% (−0.89%)
B	89.19% (−0.10%)	46.64% (−0.89%)
C	89.01% (−0.28%)	46.97% (−0.56%)
AB	88.96% (−0.33%)	46.32% (−1.21%)
AC	88.74% (−0.55%)	46.56% (−0.97%)
BC	88.75% (−0.54%)	45.92% (−1.61%)
ABC	88.71% (−0.58%)	45.19% (−2.34%)

図 4　学習データと解析精度

Fig. 4　Relationship between accuracy and size of training data

が実際に存在する☆. このような場合, 語彙情報を用いない限り正しい係り受け関係を同定するのは困難であり, 語彙情報を意図的に削除する事は我々の直観と反している.

我々は, SVM を学習アルゴリズムとして使用した場合, 語彙情報といった低頻度の素性を削除することがどれくらい精度に影響を与えるのか, 実験, 調査を行った. **表 4** にその結果を示す.

結果, 頻度閾値が 1 の場合, つまり低頻度の素性を削除せず, すべて採用した

☆ 「名詞の + 形容詞 + 名詞」の例などがある.
「水面の美しい川」と「近所の美しい川」は係り関係が異なる

表 4　素性の頻度閾値と解析精度

Table 4　Relationship between accuracy and threshold of frequencies

頻度閾値	解析精度 (精度の増減)	
	係り受け正解率	文正解率
1	89.29% (±0.00%)	47.83% (±0.00%)
2	88.68% (−0.61%)	46.31% (−1.52%)
4	87.78% (−1.51%)	44.54% (−3.29%)
6	87.63% (−1.66%)	44.78% (−3.05%)
8	87.36% (−1.93%)	42.46% (−5.37%)
10	87.28% (−2.01%)	42.22% (−5.61%)

表 5　使用した静的素性

Table 5　List of static features

前/後 文節	主辞見出し, 主辞品詞, 主辞品詞細分類, 主辞活用, 主辞活用形, 語形見出し, 語形品詞, 語形品詞細分類, 語形活用, 語形活用形, 括弧の有無, 句読点の有無, 文節の位置 (文頭, 文末)
文節間	距離 (1,2-5,6 以上), すべての助詞 (は, が, を, に ...), 括弧, 句読点の有無

場合が最も精度が高くなることが分かった. また, 表から, 頻度閾値を大きくするにつれて精度が単調に低下していることが分かる. このことは, SVM が語彙といった低頻度の素性も含め, 与えられた素性集合から柔軟に素性を選択していることを示唆している. この結果から, 語彙情報は, その頻度によらず統計的係り解析において極めて有効に機能することが改めて確認できた.

4.5　従来法, 関連研究との比較

4.5.1　解析モデルの観点から

我々は, 従来法に基づき, 機械学習アルゴリズムとして SVM を用いた係り受け解析を提案した[6]. 従来手法では, 係り受けの尤度（確率値）を要求するため, 我々は, 学習データ中から実際に係った関係を正例, 係らなかった事例を負例として学習を行い, 分離平面からの距離を尤度とすることで, 従来モデルの枠組で

図 5　再帰的な動的素性

Fig. 5　Recursive dynamic features

図 6　例外的事例の追加

Fig. 6　Risks of learning with exceptional examples

解析を行った. 直感的には, すべての係り受け候補を学習データとし, 学習データの量が多いぶん従来法のほうが高い精度が得られると期待できる. しかしながら, 実際には提案手法の方が高い精度を示している. この要因はいったい何なのであろうか.

従来法ではすべての係り受け候補を学習データとするため, 係った文節と似た語形情報を持つ後方の文節は例外的な事例となってしまう. さらに, 非交差条件により手前に係らなかった事例で, 同じように係った文節と似た語形情報を持つものも例外的な事例となる. 例えば, **図 6** の場合, 従来法では 「この本を — 探している」という二つの文節は, この表現だけを見れば, 係り受け関係があっても不自然ではないが, この場合, 負例として学習されてしまう. また, 多くの文節は直後の文節に係る事を考えると, 上記のような事例は無視できない量となり, 結果として不必要に多くの例外的事例までも学習する可能性がある. 従来法では, このような例外的な事例は,「文節間の距離」が有効に機能することで, 係り先として選択されにくいとされてきた. しかし, 実験の結果, 上記のような例外事例を学習データとして加えることは, 係り受け解析には必ずしも有効ではないとう事が分かった.

この結果は, 同じ係らない現象でも, 越えて係ったのか, 手前に係ったのか, さらに非交差条件から係らなかったのか, これら 3 つの関係には, 学習事例として効力に差あることを示唆している. 特に,「文節はできるだけ近い文節に係る」というヒューリスティックスを考えると, 実際に係った事例と越えて係った事例

の事例差を優先して学習することが重要であろう. 宇津呂ら[16] は, 係り関係を,「係る」「越える」のみに限定したモデルを, 内元ら[12] は, 係り関係を,「係る」「越える」「手前」という三つに分けるモデルを提案し, 従来の「係る」「係らない」のみを考慮するモデルよりも精度が向上したと報告している. これらの手法は, 上記の考察による例外的事例を切り分けて考えることができるため, 有効な戦略だと言える. 我々の手法も, チャンキングを段階的に適用することで,「係る」と「超える」の 2 つの関係のみに限定しているといえる. ただし, 我々の手法は, チャンキングを適用しながら学習事例を抽出するために, 非交差条件により係らない関係は学習事例に追加されない点が彼等のモデルと異なる. 実験結果により, 非交差条件により係らない関係を学習事例として追加しない事も, 積極的に精度を低下させる要因にはならないことが分かった.

提案手法は, 精度向上につながりにくい関係を学習事例から排除し, 解析時にも, それらの関係を考慮しないことにより, 解析精度を低下させることなく, 学習, 解析効率を向上させたモデルとなっている.

4.5.2 学習の観点から

係り受け解析は, 係り元と係り先の素性の組を学習する必要があるため, 学習器は素性の非線形性を考慮できなければならない. 内元[9),12)], 金山[13)] らは学習アルゴリズムに, 最大エントロピー（ME）法を選択しているが, ME は素性の独立性を前提としているため, 係り元と係り先の素性の組を新たな素性として追加しなければならず, 係り受け解析の学習には不向きであると考える. 実際に, 同じ学習, テストコーパスを用いて実験を行っている内元らの手法は, 我々の静的素性のみの手法と比較しても 1% 程度劣っている（87.93%）. 使用した素性がほとんど変わらないことを考えると, 上記の ME の持つ弱点が性能差の要因の一つとなっていると考える.

決定木（決定リスト）は, 貪欲的に非線形性を考慮することが可能であるが, それ自身では過学習に陥いりがちなので, Boosting[17)] の弱学習器として用いるのが一般的である. Boosting と SVM は, 共に**マージン最大化** という戦略に基づくアルゴリズムであり, 過学習を起こしにくいとされている☆. しかし, 弱学習器としての決定木は, 語彙を投入しても逆に精度が下がるという報告[8)] がある事から, 慎重な素性選択（どの語彙を用いるか等）を要求する点で劣る.

☆ Boosting と SVM はマージンの捉え方, 計測方法が異なる

一方 SVM は, Kernel 関数により計算量を変える事なく素性の非線形性を考慮できる学習器であり, 係り受け解析に適している. また, SVM は与えられた素性数に依存しない高い汎化能力を持つため, 他の手法に比べ, 慎重な素性選択を要求せず, それ自身が素性選択の能力を持ちあわせている. 実際に, 頻度閾値を設定せず, 語彙を含めすべての素性を用いた場合が最も精度が良く, 過学習に陥いるとことなく, 適切な素性を選択していることがその事実を裏付けている.

4.6 今後の課題

4.6.1 後方文脈の考慮

係り受け関係の同定に, 係り先よりも後方にある文節列 (後方文脈) の有効性が明らかになっている. 我々の提案手法は, 直後の文節のみを考慮するために, 後方文脈を考慮する能力を備えていない. 例えば「(僕の母) の (ダイヤの指輪)」といった例は, 提案する決定的なモデルでは, 多くの場合正しく解析できないだろう.

しかし, 決定的な解析においても, 後方文脈を考慮しながら解析することで上記のような問題が解決できると考えられる. 後方文脈を考慮するモデルとしては, 金山らの研究が興味深い. 彼らは, 文法を用いて係り先の後方を絞り, それらの候補を係り受け確率の条件部に投入することで, 後方文脈を考慮し, 精度向上に成功している. この手法の長所は, 文法を用いることで, 解析に有効な限られた後方文脈のみを用いる点にある. すべての候補ではなく, 文法で制限するために, 例外的な事例を排除するとともに, 学習効率を維持することができる.

我々は, 金山の手法に基づき, 後方文脈を考慮するモデルを構築したいと考える. その際, 金山らは既存の文法を用いているが, 既存の文法に依存しない解析手法の提案を試みたい.

4.6.2 動的素性の詳細な調査

本稿では, 基本となる三種の動的素性（A,B,C）を提案し, それらが係り受け解析に有効である事を示した. 一方で我々の提案手法は, 注目している二文節よりスコープの狭いあらゆる係り関係が動的素性として考慮可能であるため, **図 5** のように, これらの基本三素性を再帰的に繋げた素性（A',B',C'）も動的素性の候補となりうる. しかし, 不必要に素性を増やしていくと, 過学習に陥いったり, 効率性の面で問題がある. また, 実際問題として, これらの再帰的な動的素性をどのように表現し個々の学習器に与えるのかも問題となる. 今後, 再帰的な動的素性を含め, 動的素性に関し, さらなる調査を行いたいと考えている.

4.7 単語の利用

本稿では, 係り受け解析において, その出現頻度によらず単語情報が有効であることを示した. しかしながら, 単語そのものを素性として用いる場合, どうしてもデータ量の過疎性の問題は避けられない.

過去の研究において, 分類語彙表のようなシソーラスの情報は統計的係り受け解析には有効ではなく, 過疎性の問題に対処できない事が報告されている[7),8)]. 我々も同様に分類語彙表を素性として用いた実験を行ったが, 解析精度に変化は見られず, 過去の研究と同様の結果となった.

さらなる精度向上のためには, 分類語彙表以外のシソーラスや, より構造化された格フレーム情報などを用いる事が考えられるが, これらは係り受け解析とは異なるの観点から編纂された「静的な」情報であるため, 精度向上につながる保証は無い.

その一方で, 近年, 少量の既知データと大量の未知データとを同時に使用しながら学習を行う種々の手法が提案されており, 文書分類等の応用においてその有効性が報告されている. 例えば, Co-Training[18)], Transductive SVM[19)], 独立成分分析 (ICA) や主成分分析 (PCA) を用いた素性空間再構成[20)] などがある. これらの手法を用いることで, 係り受けの観点から収集された既知データ及び未知データ両方を考慮しながら学習が行なわれることが期待される. つまり, 学習データに則した一種の「動的」な情報が利用でき, 「静的」に構築された情報のみを用いる場合に比べ, 精度向上の可能性が高い. また, 大量の未知データには, 既知データ中に出現しなかった語彙情報が含まれる可能性が高く, 結果的にそらの単語情報を取り入れながら学習することが期待できる.

今後は, これらの未知データを学習に取り入れる手法を用い, 単語の過疎性に対処するモデルの構築に取り組みたいと考える.

5. おわりに

本稿では, チャンキングの段階適用による係り受け解析モデルを提案した. 提案するモデルは, 文節がその直後の文節に係るか係らないかという観点のみで決定的に解析を行うため, 従来法に比べ高効率である. また, チャンキングを採用することで, 解析精度を損うことなく, 一文あたりに用いる学習データの量を減らすことに成功した. SVM といった学習データ数に対し非線型に計算量が増加するような学習モデルではこれまで学習が事実上困難であったが, 本提案手

法により, ある程度大量のデータからの学習が可能となり, 結果としてこれまでに達成できなかった高い精度を実現した. さらに, チャンキングの導入により, 係り関係そのものを素性として導入することができ, 他の係り関係は他とは独立としていた従来手法に比べ精度向上につながった.

なお, 我々は, 提案手法に基づく係り受け解析器 *CaboCha* (南瓜) をフリーフトウェアとして公開している☆.

参 考 文 献

1) Vapnik, V. N.: *Statistical Learning Theory*, Wiley-Interscience (1998).
2) Joachims, T.: Text Categorization with Support Vector Machines: Learning with Many Relevant Features, *Proceedings of ECML-1998*, pp. 137–142 (1998).
3) 平博順, 春野雅彦: Support Vector Machine によるテキスト分類における属性選択, 情報処理学会論文誌, Vol. 41, No. 4, p. 1113 (2000).
4) Kudo, T. and Matsumoto, Y.: Use of Support Vector Learning for Chunk Identification, *Proceedings of CoNLL/LLL-2000*, pp. 142–144 (2000).
5) Kudo, T. and Matsumoto, Y.: Chunking with Support Vector Machines, *Proceedings of NAACL-2001*, pp. 192–199 (2001).
6) Kudo, T. and Matsumoto, Y.: Japanese Dependency Structure Analysis Based on Support Vector Machines, *Proceedings of EMNLP/VLC-2000*, pp. 18–25 (2000).
7) 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, p. 4201 (1999).
8) 春野雅彦, 白井諭, 大山芳史: 決定木を用いた日本語係り受け解析, 情報処理学会論文誌, Vol. 39, No. 12, p. 3117 (1998).
9) 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).
10) Abney, S.: Parsing By Chunking, *Principle-Based Parsing*, Kluwer Academic Publishers, pp. 257–300 (1991).

☆ http://cl.aist-nara.ac.jp/ taku-ku/software/cabocha/

11) Ratnaparkhi, A.: A Liner Observed Time Statistical Parser Based on Maximum Entropy Models, *Proceedings of EMNLP-1997*, pp. 1–10 (1997).
12) 内元清貴, 村田真樹, 関根聡, 井佐原均: 後方文脈を考慮した係り受けモデル, 自然言語処理, Vol. 7, No. 5, pp. 3–17 (2000).
13) 金山博, 鳥澤健太郎, 光石豊, 辻井潤一: 3 つ以上の候補から係り先を選択する係り受けモデル, 自然言語処理, Vol. 7, No. 5, pp. 71–91 (2000).
14) 関根聡, 内元清貴, 井佐原均: 文末から解析する統計的係り受け解析アルゴリズム, 自然言語処理, Vol. 6, No. 3, pp. 59–73 (1999).
15) 黒橋禎夫, 長尾眞: 京都大学テキストコーパス・プロジェクト, 言語処理学会第 3 回年次大会, pp. 115–118 (1997).
16) 宇津呂武仁, 西岡山滋之, 藤尾正和, 松本裕治: コーパスからの日本語従属節係り受け選好情報の抽出およびその評価, 自然言語処理, Vol. 6, No. 7, pp. 29–60 (1999).
17) Freund, Y. and Schapire, R.E.: Experiments with a new Boosting algoritm, *Processing of ICML-1996* (1996).
18) Blum, A. and Mitchell, T.: Combining Labeled and Unlabeled Data with Co-training, *In Proceedings of COLT-1998*, pp. 92–100 (1998).
19) Joachims, T.: Transductive Inference for Text Classification using Support Vector Machines, *Proceedings of ICML-1999*, pp. 200–209 (1999).
20) Takamura, H. and Matsumoto, Y.: Feature Space Restructuring for SVMs with Application to Text Categorization, *Proceedings of EMNLP-2001*, pp. 51–57 (2001).