決定木の混合を利用した日本語係り受け解析

春野 雅彦* 白井 諭† 大山 芳史†
*ATR 人間情報通信研究所
†NTT コミュニケーション科学研究所

1 はじめに

本稿ではコーパスから決定木を構成し、日本語係り受け解析における係り受け確率の計算と必要な属性の選択に利用する手法を提案する。続いて、自然言語データに多く含まれる例外的事例を抽出し係り受け確率の精度を更に向上させるため複数の決定木を作成し係り受け確率の計算する手法(ブースティング)について述べる。

これまでに行われた統計的依存解析の研究 [1] では、文節の種類によらず予め決められた属性すべてを使った条件付き確率で係り受け確率を評価するため有限のデータから計算する条件付き確率が正確であるためには、属性数は少数に成らざるを得なかった(データスパースネスの問題)。我々の手法では決定木を利用することで、係り受け関係にある文節とそうでない文節を弁別する属性が、2文節の種類や周囲の環境に応じて重要な順に必要な数だけ選択される。したがって大量の属性をシステムに与えても必要がなければ利用されず、データスパースネスの問題を避けられる。このため我々のシステムでは従来研究で解析精度向上に有効であることが知られている属性を自由に取り込むことが可能となった。また、ブースティングの適用によって自然言語システムの性能を落とす主要な原因である例外的な表現を取り扱う規則も学習することを可能とした。

本稿の構成は以下の通りである。2章で決定木を用いて係り受け解析の統計モデルを構成する手法について述べ、ブースティングや学習に用いた属性についても説明する。3章ではEDRコーパスを用いて行った様々な実験結果について報告する。最後に4章で本稿をまとめる。

2 統計的係受け解析と決定木の利用

2.1 統計的係り受け解析モデル

入力文を S とし、S が m 個の文節集合 $B(\{< b_1, f_1 >, \cdots, < b_m, f_m >\})$ に分けられるとする。ただし、b_i は各々文節であり、f_i は i 番目の文節が持つ様々な属性である。ここで文全体の係り受け集合 D を $D = \{mod(1), \cdots, mod(m-1)\}$ とする。$mod(i)$ は i 番目の文節が係る文節の番号を示している。これ以降 D は以下の条件を満たすものと仮定する。

- 各文節は自分より後ろに必ず係り先を持つ
- 各係り受けが交差することはない(非交差条件)

統計的係り受け解析とは、1文に訓練データの観点から見て最も確率が高い係り受け集合 D_{best} を割り当てる過程である。係り受け集合は文節集合から決定されるので、統計的係り受け解析は以下の様に書くことが出来る。

$$D_{best} = argmax_D P(D|S) = argmax_D P(D|B)$$

各係り受けが先ほどの条件を満し、他の係り受けとは独立であると仮定すると $P(D|B)$ は

$$P(D|B) = \prod_{i=1}^{m} P(mod(i) = j|f_1, \cdots, f_m) \qquad (1)$$

と変形出来る。注意すべきことは各係り受けは独立であるが、その係り受け確率自体は1文全体の属性集合 $\{f_1, \cdots, f_m\}$ から決定されていることである。我々が提案する手法は学習データから決定木 DT を構成して、属性集合 $\{f_1, \cdots, f_m\}$ 中から必要な属性を自動的に選択する。すなわち(1)式中の係り受け確率 $P(mod(i) = j|f_1, \cdots, f_m)$ を(2)式の様に近似しようとするものである。

$$P(mod(i) = j|f_1, \cdots, f_m) \simeq P(mod(i) = j|DT) \qquad (2)$$

2.2 係り受け確率の計算と利用する属性

学習に使用する属性は2文節(前文節と後文節)とその周辺の構文的特徴を中心とした言語情報で、その一覧を表1に示す。今回は決定木を利用した係り受け解析の基本性能を評価する目的で、比較的単純な構文的属性のみを用いた。例外的なのは属性番号2,9の主辞見出しであり、頻出単語の表層、分類語彙表 [5] のカテゴリを利用した。これらの属性の構文解析精度への影響については次章で述べる。学習するクラスはコーパス中での係り受け関係の有無を示す0,1の2値である。決定木構成アルゴリズムは、係り受け関係の判定に関連の深い属性を順に選び決定木 DT を構成する。

(2)式の $P(mod(i) = j|DT)$ を計算する準備として、決定木DTから文節 b_i と文節 b_j が1文内で係り受け関

係にあった確率 $P(mod(i,j)|DT)$ (クラスが1である確率) をラプラス推定量として計算する。$P(mod(i,j)|DT)$ は文節 b_i と文節 b_j の係り易さを示す確率であるが、あらゆる距離で現れた係り受け関係を含んでいる。従って $P(mod(i,j)|DT)$ から $P(mod(i) = j|DT)$ を計算するには、b_i の他の係り先候補も考慮し $P(mod(i,j)|DT)$ を b_i の全ての係り先で正規化を行った (3) 式を用いる。もちろん (3) 式の $P(mod(i,j)|DT)$ の代わりに決定木中の頻度分布を用いて $P(mod(i) = j|DT)$ を計算することも可能である。その場合と比較して (3) 式は遠い係り受けを重視する傾向がある。

$$P(mod(i) = j|DT) = \frac{P(mod(i,j)|DT)}{\sum_{k \geq i}^{m} P(mod(i,k)|DT)} \quad (3)$$

属性番号	内容	属性番号	内容
1	前文節番号	10	後文節主辞品詞
2	前文節主辞見出し	11	後文節語形
3	前文節主辞品詞	12	後文節句読点の有無
4	前文節語形	13	後文節括弧開の有無
5	前文節句読点の有無	14	後文節括弧閉の有無
6	前文節括弧開の有無	15	文節間距離
7	前文節括弧閉の有無	16	文節間読点の有無
8	後文節番号	17	文節間"は"の有無
9	後文節主辞見出し		

表 1: 学習に利用する属性一覧

入力: N 個の事例 $< e_1, w_1 >, \ldots, < e_N, w_N >$
ここで e_i,w_i は各々、事例とその重みを表す
初期化 $i = 1, \ldots, N$ の全ての事例に対して w_i =1
Do for $t = 1, 2, \ldots, T$
  w_i を与えて *C4.5*(事例の個数を重みでカウント) を呼び
  決定木 T_t を作る
  $Error$ を T_t を用いて正しく解析出来なかった事例とする
  T_t の誤り率: $\epsilon_t = \sum_{i \subset Error} w_i / \sum_{i=1}^{N} w_i$
  $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$
  T_t で正しく解析出来た事例は, 重みを以下の様に修正
    $w_i = w_i \beta_t$
出力 最終的な確率予測
  $T_f = \sum_{t=1}^{T} (log \frac{1}{\beta_t}) T_t / \sum_{t=1}^{T} (log \frac{1}{\beta_t})$

表 2: Adaboost アルゴリズムで決定木を混合する方法

2.3 ブースティングアルゴリズムの適用

表 2 に Adaboost [2] アルゴリズムによって決定木を混合する方法を示す。まず全事例の重みを 1 に初期化し、順次決定木の作成と結果の評価による重みの更新を続ける。最終的な予測確率は各決定木の解析精度に応じてそれらを重み付け平均することで得られる。なお途中で誤り率 T_t の誤り率 ϵ_t が 0.5 以上になった時にはループを抜け学習を終了する。このアルゴリズムで構成される決定木は一般的な木から次第に例外的事例に対応した特殊なもの木へと変化していく。

3 実験と考察

提案手法の定量的評価を行うため、EDR コーパス [7] を用いて以下の 3 項目の実験を行った。以下の各節でそれぞれの結果について述べる。

- データ量と解析精度の関係
- 種々の属性の影響
- 文節主辞の単語、分類語彙表カテゴリを属性に加えた場合の精度

本研究で構文解析の精度とは構文解析システムが付けた係り受中で、EDR コーパスでも係り受け関係が付与されたものの割合を示す。また、訓練データ、テストデータは以下の方法で作成した。

1. EDR コーパスから文を抽出し形態素解析を行った後、文節に分解した。
2. 1 の出力から 2 文節ずつの組み合わせを作成し、これを EDR コーパスの係り受け可否の情報 (ブラケット情報のみ) と比較する。この時、EDR の係り受けとの対応を完全に取ることが出来ない文節組を含む文のデータは採用しない (その文から作られる 2 文節の組み合わせのうち、1 つでも不正なものがある時は文全体のデータを採用しない)。
3. 2 で残ったコーパス (総文数 207802, 総文節数 1790920) を 20 個のファイルに分ける (1 個のファイルが約 1 万文強)。訓練データは文数に応じて、各ファイルの先頭から同じ文数ずつ取り出し作成した。テストデータ (1 万文) は、訓練データとの重なりが無いように、20 に分けた各ファイルの 2501 文目から 500 文ずつ取り出して作成した。

3.1 訓練データ数と解析精度

表 3 に様々な数の訓練データから決定木を作成し、同じ 1 万文のテストデータで評価した解析精度を示す。図 1 は訓練データ数と構文解析精度の関係を分かり易くするため、同じデータを学習曲線の形に書き改めたものでる。実線がブースティング、点線が単一の決定木の解析精度を表す。ブースティングの繰り返し回数は 5 回である。

図 1 から両者とも訓練データ数が 3 万文程度までは急激に解析精度が向上し、その後学習曲線の立ち上がりは鈍り始め 4 万文から 5 万文で収束にかなり近付くことが分かる。2 つの学習曲線を比較するとあらゆるデータ数に対してブースティングの精度が高く、特に少ない事

訓練データ数	3000 文	6000 文	10000 文	20000 文	30000 文	50000 文
単一決定木	82.07%	82.70%	83.52%	84.07%	84.27%	84.33%
ブースティング	83.10%	84.03%	84.44%	84.74%	84.91%	85.03%

表 3: 訓練データ数と解析精度

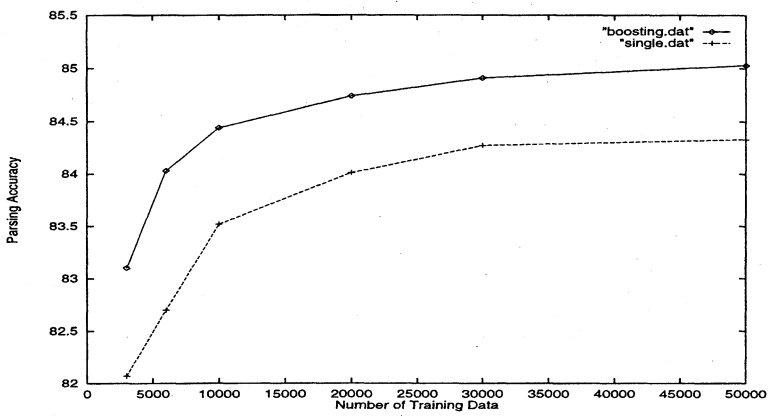

図 1: 学習曲線

例数に対して曲線の立ち上がりが早いことが分かる。また、表 4 は訓練事例数 5 万文の場合の様々な繰り返し回数に対するブースティングの正解率を示す。この表から繰り返し回数が増えてもほとんど訓練事例への過適応が起こらないことが分かる。

次に解析精度の最大値 85.03% という数字について検討する。最近の英語の統計的構文解析システム [1, 3] では我々と同種の情報を用いて 86 ～ 87% の解析精度が得られていること、日本語文節の係り先は自分より後ろであり, 英語よりも予測が行い易いと考えられることの 2 点を考慮すると現状の精度は高いとは言えない。[1, 3] で利用される Penn Treebank [4] コーパスは細かい形態素情報を含んでいるため part-of-speech tagger の構成にも同時に利用されている。また、これらのパーザーでは様々な構文的カテゴリ名を利用した学習を行っている。一方、我々は EDR コーパスの括弧付けのみを利用し、形態素解析には Chasen [6] を用いた。このことから解析精度差が生じるものと考えられる。

他方、学習システムの評価はデータの質に大きく依存する。1 万文のテストデータを訓練データとしても使い解析精度を評価したところ、解析精度は 88.85% に留まった。このことから低い解析精度の原因として、データの揺れやノイズによる影響も考えられる[1]。今後様々な機関で精度の高い日本語コーパスの構築が進むことでこれらの問題は解決されるであろう。

[1]これらの考察は同じく EDR コーパスに Collins [1] と同様の手法を適用した藤尾らのシステムの解析精度が 80.48% に留まっていることからも支持される。

3.2 種々の属性の影響

表 5 に 1 万文の訓練データに対して、単一決定木の場合の各属性の解析精度への影響を示した。具体的には個々の属性を利用しない場合にどの程度解析精度が低下するかを表している。

表 5 の結果から係り受け解析に特に有効な属性は前文節の語形と文節間距離であることが分かる。この 2 属性の組合せは直感的に理解すると'可能な範囲で出来るだけ近い係り先を優先する'という頻繁に用いらてきたヒューリスティックスを表すと考えて良い。'可能な範囲'や'優先のさせかた'を統計を用いて柔軟に設定出来るのが学習に基づく手法の利点であるとも言える。この結果から今後より高い解析精度を達成するためには、文節語形の詳細な検討が必要となる。

他の属性の多くは僅かずつ解析精度の向上に寄与している。文字種などを含むこの種の属性数を増やすことも今後の重要な課題である。括弧に関する情報が解析精度に寄与しなかった理由は EDR コーパスには、括弧を含む表現が少ないことがあげられる。この属性の有効性については他のコーパスを利用した検証が必要である。また、前文節主辞品詞が唯一解析精度を低下させているが詳細な理由は判明していない。恐らくこの属性は語形から推定出来るうえに、サ変名詞に関する形態素解析誤りが頻繁に起こるためではないかと考えられる。

繰り返し回数	1	2	3	4	5	6
解析精度	84.32%	84.93%	84.89%	84.86%	85.03%	85.01%

表 4: 繰り返し回数と解析精度 (事例数 5 万文)

属性内容	解析精度の低下	属性内容	解析精度の低下
前文節主辞品詞	-0.07%	後文節句読点の有無	+1.62%
前文節語形	+9.34%	後文節括弧開の有無	±0.00%
前文節句読点の有無	+1.15%	後文節括弧閉の有無	±0.00%
前文節括弧開の有無	±0.00%	文節間距離	+5.21%
前文節括弧閉の有無	±0.00%	文節間読点の有無	+0.01%
後文節主辞品詞	+2.13%	文節間"は"の有無	+1.79%
前文節語形	+0.52%		

表 5: 個々の属性の削除による解析精度の低下

3.3 頻出単語の表層、分類語彙表カテゴリの利用

本節では文節の主辞の語彙情報を属性として利用した場合の単一決定木の解析精度について述べる。訓練データは 1 万文で、利用した属性は以下の 4 種類である。

- 出現回数上位 100 語
- 出現回数上位 200 語
- 分類語彙表 [5] 小数点以下 1 桁
- 分類語彙表小数点以下 2 桁

表 6 は各々の属性に対する解析精度を示す。全ての属性について解析精度は主辞の語彙情報を利用しない場合 (83.52%) より劣っている。特に頻出語と分類語彙表の両者で情報を多く使うほど精度が悪くなることは注目に値する。詳細な原因を特定することは難しいが、決定木の上位の段階でこれらの属性による事例分割が行われ、その属性を利用出来ない (頻出語でない、あるいは分類語彙表に掲載されていない主辞) 事例の解析精度が下がる傾向が見られる。この実験の結果から語彙情報を利用すれば解析精度が上がるという安易な期待は成り立たないことが分かる。

語彙	100 語	200 語	分類語彙表 1 桁	分類語彙表 2 桁
精度	83.34%	82.68%	82.51%	81.67%

表 6: 主辞の語彙と解析精度

4 結論

本稿ではコーパスから構成した決定木を利用する日本語係り受け解析法について述べた。係り受け解析に決定木を利用することで多くの属性を利用した場合にも動的な属性選択が可能となった。その結果、多くの先行研究の知見を統計的学習の枠組に取り込むことが可能となった。またブースティングの適用により更に解析精度が向上することを示した。今後は提案手法でのより高い解析精度の達成と一般性検証のため以下の項目について研究を進める予定である。

- 係り側、受け側文節に関する詳細な情報を含む様々な属性の解析精度への影響の調査
- 日本語だけでなく英語等の言語への本手法の適用

参考文献

[1] Michael Collins. A New Statistical Parser based on bigram lexical dependencies. In *Proc. 34th Annual Meeting of Association for Computational Linguistics*, pages 184–191, 1996.

[2] Yoav Freund and Robert Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. 1996.

[3] David M. Magerman. Statistical Decision-Tree Models for Parsing. In *Proc.33rd Annual Meeting of Association for Computational Linguistics*, pages 276–283, 1995.

[4] Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, June 1993.

[5] 国立国語研究所. 分類語彙表. 秀英出版, 1964.

[6] 松本 裕治 他. 形態素解析システム Chasen2.0 使用説明書, 1996.

[7] 日本電子化辞書研究所. *EDR* 電子化辞書仕様説明書, 1995.