Vol.2010-NL-198 No.7
2010/9/17

決定的な解析と相対的な比較による解析の二側面を持つ日本語係り受け解析

山本悠二[†1]　増山繁[†1]

日本語係り受け解析の手法は大きく分けて，1. 決定的な解析方法と，2. 係り先候補の確信度に基づく解析方法がある．前者は係り先候補間の比較が行えないことから，特に長距離依存の係り先を同定するときに誤りを生じやすいという傾向がある．また，後者は係り先候補集合のすべての要素を探索するため，計算時間の点で問題がある．提案手法では，係り先候補の確信度に基づく解析方法での解析時間を減らすために，決定的な解析が容易な文節について先に係り先を定めた後に，相対的な比較による係り先の同定する方法を示す．京都テキストコーパス 4.0 を用いて提案手法を評価したところ，係り先候補の確信度に基づく解析方法の 1 つである相対モデルと比較してほぼ同等の解析性能を持ち，かつ，実行時間が 2.4 倍程度高速であることが確認された．

Bilateral Japanese Denpendency Parsing - Deterministic and Comparative by Relative Preferences of Dependency

Yuji Yamamoto[†1] and Shigeru Masuyama[†1]

Japanese dependency parsers fall into two main methods, 1) deterministic parsing and 2) parsing based on dependency certainties among modifee candidates. The former methods tend to make errors especially for identifying long-distance dependencies because these methods do not opt the candidate by comparing candidates. On the other hand, the latter methods have difficulty with their parsing speed due to searching the most preferable candidate from all modifee candidates. The proposed method identifies easily-analyzable dependencies by deterministic parsing and identifies the rest dependencies by parsing based on dependency certainties among modifee candidates later. Experiments using the Kyoto Text Corpus show that the proposed method runs 2.4 times faster than the relative-model parser while the dependency accuracy of the proposed method is nearly comparable with the relative-model's.

1. はじめに

係り受け解析は自然言語処理における基本技術として認識されている．特に，その精度向上は，解析後の応用タスクにおける結果に直接影響することが多いため，研究課題として重要な位置を占めている．また，計算効率についての研究も実用的な自然言語処理のアプリケーションにとって重要である．

従来提案されてきた統計的日本語係り受け解析は大きく分けて，1) 決定的な解析方法[1),2)]と，2) 係り先候補の確信度に基づく解析方法[3),4)]がある．ここで，前者の方法は，Shift-Reduce 法による決定的な解析[⋆1]を指す．この解析方法では，ある係り元に注目したときに，その係り先を決定する過程を観察すると，多くの場合，係り先候補集合のいくつかの候補を取り出して係り先となりうるか否かを調べるだけで係り先を決定できていることが確認できる．そのため，後者の係り先候補集合のすべての要素を取り出して係りやすさの確信度を求める方法に比べて高速に係り受け解析を行うことができる．特に論文 1) では，文節数に対して線形時間で解析が行えるアルゴリズムが示されている．

しかしながら，決定的な解析では，特に長距離依存の係り先を同定するときに誤りを生じやすいという傾向がある．これは，係り先候補 A も係り先候補 B も係る可能性がある場合に近い方の候補で Reduce 操作が行われやすいためである．特に精度よく長距離依存を解析することは，文短縮要約[5)]や，節単位を基本とした表現の獲得[6)]といった文全体の構造を把握する必要があるタスクにおいて重要である．このような場合，統計的日本語係り受け解析のもう 1 つの方法である，係り先候補の確信度に基づく解析方法を用いることが好ましい．この解析方法では，ある係り元の係り先を求める場合，係り元とその候補の文節対の係りやすさの確信度をすべての候補について求める．そして，確信度が最も高い候補を係り先として選択する．論文 3) では，確信度の推定方法として相対モデル[⋆2]を用いた係り受け解析モデルと，決定的な解析方法の 1 つであるチャンキングモデル[2)]の解析結果を比較し，前者は長距離依存に，後者は短距離依存に強いことを指摘した．しかし，係り先候補の確信度に基づく解析方法は，係り先候補集合のすべての要素[⋆3]について確信度を求める必要があ

†1 豊橋技術科学大学
Toyohashi University of Technology

⋆1 決定的な解析とは，常に唯一の文法規則とその適用箇所を選択する構文解析をいう[8)]．
⋆2 優先度学習[7)]により推定された識別モデルのことである．
⋆3 正確には非交差条件を満たす係り先候補に限定している．

B[]	私は	興味本位で	この	本を	読んだ。
添字番号	0	1	2	3	4
anslink[]	4	4	3	4	-1

図 1 例文における B[]，anslink[]
Fig. 1 B[] and anslink[] on an example sentence.

るため，決定的な解析方法に比べて解析時間が掛かる．例えば，論文 4) による解析アルゴリズムを使用すると，解析時間は文節数の二乗に比例する．

そこで本稿では，係り先候補の確信度に基づく解析方法での解析時間を減らすために，決定的な解析が容易な文節について先に係り先を定めた後に，残りの文節について係り先の相対的な比較による係り先の同定を行う手法を提案する．ここで，基本的な考え方について述べる．先に述べたように決定的な解析では係り先候補集合のいくつかが係り先になりうる可能性がある場合，誤りが生じやすい．一方，文献 8) で示されているように，このような曖昧性がある係り受けは，係り元，係り先候補に特徴がある．提案手法は，このような係り受けに関して，決定的な解析では係り先の同定を保留しておき，後の係り先の相対的な比較によって定める．

2. 統計的日本語係り受け解析

依存文法に基づく係り受け解析でよく用いられているモデルについて説明する．前提として日本語文における係り受けは以下の制約を満たすものとする．

(1) 係り受けは前方から後方に向いている (後方修飾).
(2) 係り受けは交差しない (非交差条件).
(3) 係り受けは係り先を 1 つだけ持つ.
(4) 文末は係り先を持たない.

以下では記号の定義を行う (例は図 1 に示す)．まず，N 個の文節列で構成される日本語文について，文節列を保持する配列を B[] とする．以降，配列は 0 から始まるものとする．つまり，B[] は B[0] から B[N-1] までアクセスすることができる．また，係り受け解析後の，推定された係り先文節の添字番号が格納されている配列を estlink[] とする．また，訓練データの文節列については，正しい係り先文節の添字番号が格納されている配列 anslink[] が与えられる．

統計的係り受け解析では，始めに訓練データとして与えられた複数個の文節列を用いて，係り先を求めるための識別モデルを作る．ここで，B[] の i 番目と j 番目の文節対を表す素性ベクトルを $F(\langle \mathtt{i},\mathtt{j}\rangle,\mathtt{B})$ と表記する．また，このとき使用する機械学習に非線形カーネルを導入することが多い[⋆1]ため，特徴空間への非線形写像についてのいくつかの定義を導入する．まず，入力として与えられた素性ベクトルを特徴空間へ非線形写像する関数を $\phi(\cdot)$ と置く．つまり，先に示した文節対の素性ベクトルを非線形写像したものは $\phi(F(\langle \mathtt{i},\mathtt{j}\rangle,\mathtt{B}))$ である．以降，簡略化のため，$\phi(F(\langle \mathtt{i},\mathtt{j}\rangle,\mathtt{B}))$ を $\psi(\langle \mathtt{i},\mathtt{j}\rangle,\mathtt{B})$ と表記する．また，4 節は $\psi(\langle \mathtt{i},\mathtt{j}\rangle,\mathtt{B})$ を単位ベクトルにしたものを使用する．これも簡略化のために $\omega(\langle \mathtt{i},\mathtt{j}\rangle,\mathtt{B}) = \psi(\langle \mathtt{i},\mathtt{j}\rangle,\mathtt{B})/||\psi(\langle \mathtt{i},\mathtt{j}\rangle,\mathtt{B})||$ と表記する．なお，非線形カーネルは特徴空間内での内積を，入力された素性ベクトル上の空間 (入力空間) 内での内積の非線形写像で計算することができる (詳細は例えば文献 9) を参照)．例えば，統計的日本語係り受け解析でよく用いられる多項式カーネル (次元数を d とする) の場合，

$$\psi(\langle \mathtt{i},\mathtt{j}\rangle,\mathtt{B}) \cdot \psi(\langle \mathtt{k},\mathtt{l}\rangle,\mathtt{B'}) = \{1 + F(\langle \mathtt{i},\mathtt{j}\rangle,\mathtt{B}) \cdot F(\langle \mathtt{k},\mathtt{l}\rangle,\mathtt{B'})\}^d \quad (1)$$

で計算することができる．

3. 決定的な解析，相対的な比較による解析

3.1 決定的な解析

Shift-Reduce 法による決定的な解析とは，スタックを利用し，Shift と Reduce という 2 つの操作を組み合わせることによってボトムアップに文を解析するものである．論文 1) では，着目している文節対が識別モデルにより係ると判定される場合に Reduce 操作，それ以外の場合に Shift 操作を行い，文解析を行う．特にこのアルゴリズムは，Shift 操作と Reduce 操作に工夫を施すことで，解析時間が文節数に対して線形時間で解析が行えるところが特徴である．ここで，識別モデルの重みベクトルを $\mathbf{w}$ と定義する．このとき，文節列 B[] における i 番目 と j 番目の文節対を識別モデルで判定したときの値は $\mathbf{w} \cdot \psi(\langle \mathtt{i},\mathtt{j}\rangle,\mathtt{B})$ となる．この識別モデルは，文節対を判定したときに，値が正の値を取れば「文節対は係る」と対応付けられ，負の値を取れば「文節対は係らない」と対応付けられるように学習されているものとする．解析アルゴリズムの擬似コードは論文 1) に記載がある．

3.2 相対的な比較による解析

先に示した決定的な解析では，識別モデルを用いて，ある係り元について最短の係り先となりうる候補を求めていた．しかし，係り受け解析は，依存関係の曖昧性から，複数の候補

⋆1 特に素性集合の要素の組合せを考慮するために多項式カーネルが用いられる．

のうちで，より係り先になりやすい候補を選択する必要性が生じる．相対的な比較による解析では，ある係り元の係り先を求める場合，識別モデルを用いて，係り元とその候補の文節対の係りやすさの確信度をすべての候補について求める．そして，確信度の高い候補を係り先として選択する．ここで，文節対の係りやすさの確信度を求める識別モデルの重みベクトルを $\mathbf{w}$ とする．このとき，文節列 B[] における i 番目 と j 番目の文節対の係りやすさの確信度は $\mathbf{w} \cdot \psi(\langle \mathtt{i}, \mathtt{j} \rangle, \mathtt{B})$ となる．この識別モデルは，ある文節対が他の文節対と比較したときに，より係りやすいものであれば確信度の値もより大きくなるように学習されているものとする．解析アルゴリズムとしては，例えば，文末から解析するもの[4] がある．

4. 決定的な解析と相対的な比較による解析の二側面を持つ日本語係り受け解析

4.1 基本的な考え

提案手法では，係り先候補の確信度に基づく解析方法での解析時間を減らすために，決定的な解析が容易な文節について先に係り先を定めた後に，相対的な比較による係り先の同定する方法を示す．先に述べたように，Shift-Reduce 法による決定的な解析では長距離依存の係り先を同定するときに誤りが生じやすいという傾向がある．これは，係り先候補 A も係り先候補 B も係る可能性がある場合に近い方の候補で Reduce 操作が行われやすいためである．

一方，文献 8) で示されているように，曖昧性が生じる係り受けは，例えば以下のように，ある程度は類型化が可能である (A, B, C などは名詞，V_1，V_2 などは動詞を表す．また，下線部は曖昧性がある係り元，矩形で囲まれているものは係り先となりうるものを表す)．

(1) 「A の [B の [C]]」のような連体修飾語の係り先の曖昧性
(2) 「A が [V_1 した] B を [V_2 した]」のような格要素の係り先の曖昧性
(3) 「$\cdots V_1$ したが $\cdots$ [V_2 したので] $\cdots$ [V_3 した]」のような従属節の係り先の曖昧性
(4) 「A を V_1 したので B が [V_2 し] C を [V_3 したが] D に [V_4 された]」のような並列構造の範囲の曖昧性[⋆1]

このような類型を用いれば，決定的な解析において，上記のパターンに当てはまるものを Reduce 操作しないことで係り先の同定を保留し，後の相対的な比較による解析で係り先候補を精査することができる．しかしながら，曖昧性が生じる係り受けについての類型化を人手で行うことは網羅性の点で問題があることから，機械学習の範疇で Reduce 操作を保留する機構を組み入れることが望ましい．つまり，決定的な解析において，先のようなパターンに近い文節対が出現したときに，識別モデルが負の値を返すように学習を行えるようにする．ただし，実際のタグ付けコーパスに付与されているタグは，係り先に曖昧性があるか否かという情報はないため，単純な分類問題に帰着することはできない．そこで，1 つの識別モデルを二値分類学習と優先度学習[7] を組合わせてモデルを作ることで，先の性質を持つ識別モデルを作る．

⋆1 各下線部の文節は，その後方の矩形で囲まれた文節が係り先になりうるものであることを表している．

4.2 提 案 手 法

以下では，決定的な解析と相対的な比較による解析の二側面を持つ日本語係り受け解析 “bilateral parsing” を提案する．

4.2.1 学習アルゴリズム

提案手法の学習アルゴリズムについて示す．重みベクトル $\mathbf{w}$ を用い，正解の係り先情報がわかっている文節列に対して，決定的な解析，及び，相対的な比較による解析を行う．もし，解析途中で誤った出力になる場合や，正しい出力であったとしても十分なマージンが取れていない場合は重みベクトルを更新することで正解の係り先情報がない場合でも正しく解析できるように補正する．なお，重みベクトルを更新するための学習アルゴリズムについては，Online Passive-Aggressive Algorithm[10] (以下 OPA と略記) を用いた．

擬似コードで示した学習アルゴリズムを示す前に，入力として与える引数について説明する．まず，T は，訓練データの文集合である．T の要素は,「訓練データの文節列，文節数，正しい係り先文節の添字番号の配列」の三つ組で構成される．C は，OPA で使用する引数で，マージン違反を起こす事例に対してどれだけ積極的に重みベクトルを更新するかについて決めるパラメータである．C が大きければ，与えられた事例について，OPA の定式化によって指定されたマージンを忠実に確保するようになる．I は，訓練データセット単位で何回学習を繰り返すかについて指定するパラメータである．

擬似コードを図 2 に示す．このコードでは，訓練データから文を取り出し，決定的な係り受け解析で部分的な係り受けができるように学習 (関数は bilateral_sr_train) し，その後で残りの文節において，係り先の相対的な比較による係り受けができるように学習 (関数は bilateral_comp_learn) する．なお，5 節での実験で用いた学習アルゴリズムは，論文 11) に掲載されている重みベクトルの平均化を行なった．

擬似コードで示した決定的な係り受け解析での学習アルゴリズムを図 3 に示す．このアルゴリズムは，颯々野の係り受け解析[1] の識別モデルをオンライン学習を使用して学習するのとは以下の 2 点で異なる．

```
// 出力: 更新された重みベクトル w
funciton train(T, C, I)
w ← 0;
for (iter = 1; iter <= I; iter++) {
  foreach ( (B, N, anslink)∈ T ) {
    estlink = [-1] * N;  // 文節数分 −1 が入った配列
    (w, estlink) = bilateral_sr_train( w, C, B, N, anslink, estlink );
    (w, estlink) = bilateral_comp_learn( w, C, B, N, anslink, estlink );
  }
}
```

図 2 擬似コード - Bilateral Parsing の学習アルゴリズム
Fig. 2 Pseudo code for training the bilateral parser.

- 提案手法のアルゴリズムは，末尾まで探索しても係り先が見つからない場合は，係り先がない (つまり，estlink[] の要素が -1 のまま変わらない) とする．これらの文節は，後の相対的な比較による解析で係り先を定める．
- 文節対 $\langle \mathtt{i}, \mathtt{j} \rangle$ が正しい係り受けであるとする．そして，$\langle \mathtt{i}, \mathtt{j} \rangle$ を読み込んだ時点での重みベクトルを $\mathbf{w}$ とする．このとき，$\langle \mathtt{i}, \mathtt{j} \rangle$ をより正確に識別できるように重みベクトルを更新するための条件は $0 < \mathbf{w} \cdot \omega(\langle \mathtt{i}, \mathtt{j} \rangle, \mathtt{B}) < 1$ である．つまり，分類した値が負を取る場合は重みベクトルは更新されない．これは，文節対に係り受けの曖昧性があることを考慮しているためである．仮に，文節対に係り受けの曖昧性がある場合に正例として学習すると，よく似た形式の未知の文節対に対して Reduce 操作を行い，長距離依存の係り先同定に誤りが生じる可能性があるためである．

関数 bilateral_comp_learn (図 4) は，決定的な解析で係り先が同定できなかった係り元について優先度学習を用いて重みベクトル $\mathbf{w}$ を更新するものである．優先度学習とは次のようなものである．例えば，文節列 B において，i 番目の係り先がまだ決まっていないものとする．このとき，文節対 $\langle \mathtt{i}, \mathtt{j} \rangle$ を正しい係り受けである文節対，$\langle \mathtt{i}, \mathtt{k} \rangle$ を正しくない係り受けである文節対であるとする．優先度学習では，正しい係り受けである文節対の信頼度が，正しくない係り受けである文節対の信頼度よりも大きくなるように重みベクトルを更新する．つまり，$\mathbf{w} \cdot \omega(\langle \mathtt{i}, \mathtt{j} \rangle, \mathtt{B}) > \mathbf{w} \cdot \omega(\langle \mathtt{i}, \mathtt{k} \rangle, \mathtt{B})$ のような関係になるように重みベクトルを更新する．ここで，更新前の重みベクトルと更新後の重みベクトルの違いが分かるように，前者を $\mathbf{w}_t$，後者を $\mathbf{w}_{t+1}$ と表記する．先に示した 2 つの文節対について，更新前の重みベクトルでは信頼度の順序関係が逆転していて，これを OPA で正しい関係になるように重みベクトルを更新するという設定を考える．このとき，$\mathbf{w}_{t+1} = \mathbf{w}_t + \tau\{\omega(\langle \mathtt{i}, \mathtt{j} \rangle, \mathtt{B}) - \omega(\langle \mathtt{i}, \mathtt{k} \rangle, \mathtt{B})\}$ となる (ただし $\tau > 0$)．$\mathbf{w}_{t+1}$ と $\mathbf{w}_t$ の関係式から，$\mathbf{w}_{t+1} \cdot \omega(\langle \mathtt{i}, \mathtt{j} \rangle, \mathtt{B}) = \mathbf{w}_t \cdot \omega(\langle \mathtt{i}, \mathtt{j} \rangle, \mathtt{B}) + \tau\{||\omega(\langle \mathtt{i}, \mathtt{j} \rangle, \mathtt{B})||^2 - \omega(\langle \mathtt{i}, \mathtt{k} \rangle, \mathtt{B}) \cdot \omega(\langle \mathtt{i}, \mathtt{j} \rangle, \mathtt{B})\}$ である．$\omega(\cdot)$ は単位ベクトル，$\tau > 0$ であることに注意すると，$\mathbf{w}_{t+1} \cdot \omega(\langle \mathtt{i}, \mathtt{j} \rangle, \mathtt{B}) \geq \mathbf{w}_t \cdot \omega(\langle \mathtt{i}, \mathtt{j} \rangle, \mathtt{B})$ が成立する．また，同様にして，$\mathbf{w}_{t+1} \cdot \omega(\langle \mathtt{i}, \mathtt{k} \rangle, \mathtt{B}) \leq \mathbf{w}_t \cdot \omega(\langle \mathtt{i}, \mathtt{k} \rangle, \mathtt{B})$ が成立することが確認できる．これらの性質は次のことを意味する．まず，ある文節対が一貫して正しい係り受けになる場合，ある程度学習が進むと信頼度が大きくなり，負の値から正の値を取るようになる．この場合，決定的な係り受け解析において，正例として重みベクトルを更新する対象になるため，決定的な係り受け解析のほうで係り受けの同定が行える．一方，ある文節対が係り受けに曖昧性を持つ場合，信頼度はあまり増加せず，負の値のままになりやすい．従って，曖昧性を持つ文節対は相対的な比較による解析まで係り先の同定が遅延されることが期待できる．

4.2.2 解析アルゴリズム

解析アルゴリズムについては，学習アルゴリズムの動作と類似するため動作についての細かい説明は省略する．擬似コードについては付録 A.1 に掲載する．関数 bilateral_parsing が解析アルゴリズムのメインの関数である．なお，決定的な解析側の係り受け同定である関数 bilateral_sr_parsing の文節対 $\langle \mathtt{i}, \mathtt{j} \rangle$ の分類スコア $\mathbf{w} \cdot \omega(\langle \mathtt{i}, \mathtt{j} \rangle, \mathtt{B})$ は，動的素性が一致している場合，相対的な比較による解析側の係り受け同定である bilateral_comp_parsing の確信度としても再度使用することができる．5 節での実験で用いた解析アルゴリズムは，動的素性情報込みの文節対をキーとして識別モデルのスコアを保存することで動作の効率化を図っている．

5. 実　　験

5.1 実験設定

京都テキストコーパス 4.0[⋆1]を以下の 3 つに分けて実験を行った．

- 訓練データ: 一般記事[⋆2] 1 月 1，3-11 日，社説 1-8 月，合計 24,280 文，234,639 文節
- 開発データ: 一般記事 1 月 12，13 日，社説 9 月，合計 4,833 文，47,571 文節

⋆1 http://nlp.kuee.kyoto-u.ac.jp/nl-resource/corpus.html
⋆2 ただし，以下の ID を持つ文は文節番号とその文節の係り先番号が同一であるというタグ付けの誤りがあったため，訓練データから除外した; 950101159-010，950106177-017，950106192-002.

```
// 出力: w: 更新された重みベクトル
//       estlink: 推定された係り先文節の添字番号が格納された配列
// 関数 pop: 第一引数のスタックから，先頭の要素を取り除き，その要素を返す．
// 関数 push: 第一引数のスタックの先頭に第二引数の値を追加する．
function bilateral_sr_train( w, C, B[], N, anslink[], estlink[] )
push(stack, -1);  // −1 は番兵
push(stack,  0);
for (j = 1; j < N; j++) {
  i = pop(stack);
  while (i != -1) {
    if ( (i == N - 2) && (j == (N - 1) ) )
      estlink[i] = j;
    else {
      scr = w · ω(⟨i, j⟩, B);
      y = (j == anslink[i]) ? +1 : -1;
      τ = min{C, max{ 0, (1 − y · scr) / ||ω(⟨i, j⟩, B)||^2 }};
      if (j != anslink[i]) {
        w ← w + y τ ω(⟨i, j⟩, B);
        break;
      }
      else if ( (j == anslink[i]) && (scr >= 0) ) {
        estlink[i] = j;
        w ← w + y τ ω(⟨i, j⟩, B);
      }
    }
    i = pop(stack);
  }
  push(stack, i);
  push(stack, j);
}
```

図 3 擬似コード - Bilateral Parsing の 決定的な解析側の学習アルゴリズム
Fig. 3 Pseudo code for training the deterministic parsing of the bilateral parser.

```
// 出力: w: 更新された重みベクトル
function bilateral_comp_learn( w, C, B[], N, anslink[], estlink[] )
for (i = N - 3; i >= 0; i++) {
  if (estlink[i] != -1)
    continue;  // すでに係り先が決定している場合は係り先の学習は不要
  j = estlink[i + 1];
  while (j != -1) {
    if (j != anslink[i]) {
      link_scr = w · ω(⟨i, anslink[i]⟩, B);
      nlink_scr = w · ω(⟨i, j⟩, B);
      τ = min{C, max{ 0, (1 − link_scr + nlink_scr)
                          / ||ω(⟨i, anslink[i]⟩, B) − ω(⟨i, j⟩, B)||^2 }};
      w ← w + τ{ ω(⟨i, anslink[i]⟩, B) − ω(⟨i, j⟩, B) };
    }
    j = estlink[j];
  }
}
```

図 4 擬似コード - Bilateral Parsing の 相対的な比較による解析側の学習アルゴリズム
Fig. 4 Pseudo code for training the preference-based parsing of the bilateral parser.

- 評価データ: 一般記事 1 月 14-17 日，社説 10-12 月，合計 9,284 文，89,874 文節

これらの記事の分け方は，論文 3) と同じである．

係り受け解析手法としては，提案手法である Bilateral Parsing，決定的な解析として嚆々野の線形時間係り受け解析，また，相対的な比較による解析として相対モデルを使用した．相対モデルについての実験設定については，論文 12) に詳細があるため割愛する．使用した学習器は，Bilateral Parsing が Online Passive-Aggressive Algorithm[10]，嚆々野の線形時間係り受け解析が Support Vector Machine である．各モデルの識別モデルで用いるカーネルに 3 次の多項式を使用した．なお，今回の実験で用いる識別モデルは，すべて Polynomial Kernel Inverted Representation[13] よる計算の高速化を行なっている．また，それぞれの学習器で，コストのパラメータ (C) は 1 と定めた．なお，Bilateral Parsing，相対モデルは反復回数 I を定める必要がある．I については，1 から 10 回までの反復回数のうちで開発データの係り受け正解率が最も高くなる値を使用した．

学習に用いた素性は，CaboCha 0.53 中にある素性抽出プログラム selector.pl の出力を使用した．また，他の語彙的な素性は固有のコーパスに過度に依存する可能性があるため使用していない．ただし，同素性抽出プログラムで出力される動的素性は使用している．動的素性とは，係り元もしくは係り先において解析途中で既に得られている係り受けをもとにした素性のことである．Bilateral Parsing，及び，颯々野の線形時間係り受け解析では A. 係り元にすでに係る文節，B. 係り先にすでに係る文節 についての動的素性を使用した．

なお，実験は Xeon E5504，主記憶 32GByte の Linux 上で行った．また，実装は C++ で行い，gcc 4.4.3 の O3 オプションでコンパイルしたプログラムを実行している．

5.2 実験結果

前の節で示した 3 つの手法の評価データにおける結果を表 1 に示す．ここでの係り受け正解率は，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，文正解率は，文単位で全体の文節の係り先が正しく同定できたものの割合を示す．また，提案手法以外の係り受け正解率，文正解率については，二項検定により，両側 5% で提案手法との有意差が認められるものについてダガー (†) を付記している．加えて，本稿での目的は相対モデルの高速化であるので，1 文当たりの解析時間についても記載している．

提案手法の Bilateral Parsing は，颯々野の線形時間係り受け解析と比べて係り受け正解率，文正解率ともに有意に向上している．しかし，相対モデルに関しては文正解率は若干向上しているものの，係り受け正解率については若干下がっている．ただし，相対モデルについては有意差は認められなかった．ここから，Bilateral Parsing と相対モデルはほぼ同等の性能であると考える．

次に解析時間について見ると，颯々野の線形時間係り受け解析が最も早い．Bilateral Parsing は，線形時間係り受け解析を行ったあとで，相対的な比較による係り受け解析を行なっているため，線形時間係り受け解析よりも遅くなる．一方，Bilateral Parsing と 相対モデルを比較すると 2.4 倍程度高速に動作していることが確認できた．Bilateral Parsing が相対モデルに比べて高速に動作しているのは，決定的な解析の時点で多くの係り先が決定できているためである．実際に決定的な解析で，どれくらいの文節数が，どれぐらい正確に定めることができているのかを調べた．Bilateral Parsing の決定的な解析のみを行った結果を用いて適合率・被覆率を求めた．これらは論文 14) を参考にして次のように定めた．(適合率) = (解析器が出力した文節のうち正解した数) / (解析器が係り先を出力した文節数), (被覆率) = (解析器が係り先を出力した文節数) / (末尾の文節を除いた総文節数)．ここで，`estlink[]` の要素で -1 のものは係り先を出力した文節とカウントしないことを強調する．

表 1 結果 係り受け正解率，文正解率
Table 1 Results of dependency accuracy and sentence accuracy.

解析手法	係り受け正解率 (%)	文正解率 (%)	解析時間 (秒/文)
Bilateral Parsing ($I = 8$)	90.83 (73201 / 80590)	54.17 (5029 / 9284)	0.0589
相対モデル ($I = 7$)	90.96 (73305 / 80590)	54.10 (5023 / 9284)	0.1419
颯々野の線形時間係り受け解析	89.93 † (72474 / 80590)	51.90 † (4818 / 9284)	0.0174

†: 二項検定 [有意水準 5 % 両側] で，提案手法との有意差が認めれるもの．

結果は適合率が 0.9373，被覆率が 0.9042 であった．これより，ほとんどの文節の係り先が決定的な解析の時点で精度よく決定できていることが確認できる．

6. 関連研究

決定的な解析と相対的な比較による解析の組合せについて，論文 3) で 2 つの独立した解析器を用い，それらの解析結果を人手で定めたルールによって統合する方法を示している．しかし，本手法は，これらの解析をルールを用いることなく，統一的に 1 つの識別モデルで行うことができる点が異なる．

論文 16) では，英語の係り受け解析において，部分木の Most Probable Head を求める際にトーナメントモデル[17] を用いて候補の探索を行ってから決定的にアクションをとる手法を提案している．この方法では主辞候補集合の Most Probable Head を求めるところに関してはトーナメントモデルを用いて相対的な比較を行なっている．そのため，本稿での Shift-Reduce 法での決定的な解析とは異なり，日本語係り受け解析での係り先候補集合の中から係り先候補を比較によって選択する[18] 状況に近い．

提案手法の決定的な解析の時点での結果は，係り受け解析器において信頼性の高い解析結果を部分係り受けとして出力するものに似ている．素性設定が異なるので若干公平さに欠けるが，論文 15) によると，5.2 節で示した適合率・被覆率がよりも高いスコアを達成する解析器をトーナメントモデル[18] 用いて実現できることが示されている．ただし，この方法は高精度の部分解析を求めるために手法であり，提案手法のような決定的な解析を取っていない．実際，解析アルゴリズムの transition としてみると，トーナメントモデルによる解析手順は相対モデルの解析手順と同じである．そのため，解析速度の向上を目的として，決定的な解析の代わりにトーナメントモデルを利用することはできない．

相対モデルについては，論文 12) で優先度学習で事例を作る際に相対位置素性を加えることで係り元文節からの相対的な距離を反映させるようにした係り受け解析モデルがある．

係り受け正解率，文正解率については，それぞれ 91.24 %と 55.14 % であり，相対モデルよりも解析性能はよい．このモデルも相対モデルと同様に優先度学習で定式化されている．今回提案した決定的な解析方法との併用が可能かどうかについては興味深い課題である．

7. ま と め

本稿では，係り先候補の確信度に基づく解析方法での解析時間を減らすために，決定的な解析が容易な文節について先に係り先を定めた後に，残りの文節について係り先の相対的な比較による係り先の同定を行う手法を提案した．実験結果から，係り先候補の確信度に基づく解析方法の 1 つである相対モデル比較してほぼ同等の解析性能を持ち，かつ，実行時間が 2.4 倍程度高速であることが確認された．

Shift-Reduce 法は構文解析に限らず，日本語固有表現抽出[19] や形態素解析[20] などにも応用されている．これらの分野に本手法が適用できないかについては今後の課題である．

謝辞 本研究は文部科学省グローバル COE プログラム「インテリジェント センシングのフロンティア」，日本学術振興会科研費基盤 (C) 22500129，電気通信普及財団，及び，電気通信普及財団の援助により行われた．

参 考 文 献

1) 颯々野 学，“日本語係り受け解析の線形時間アルゴリズム，” 自然言語処理，vol.14, no.1, pp.3-18，Jan. 2007.
2) 工藤 拓，松本 裕治，“チャンキングの段階適用による日本語係り受け解析，” 情報処理学会論文誌，vol.43，no.6，pp.1832-1842，Jun. 2002.
3) 工藤 拓，松本 裕治，“相対的な係りやすさを考慮した日本語係り受け解析モデル，” 情報処理学会論文誌，vol.46，no.4，pp.1082-1092，Apr. 2005.
4) 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，Apr. 1999.
5) Tadashi Nomoto. “A Generic Sentence Trimmer with CRFs，” In Proc. of ACL-08: HLT，pp.299-307，2008.
6) Hiroki Sakaji, Satoshi Sekine, Shigeru Masuyama. “Extracting Causal Knowledge Using Clue Phrases and Syntactic Patterns，” In Proc. of PAKM 2008，pp.111-122, 2008.
7) Ralf Herbrich, Thore Graepel, Peter Bollmann-Sdorra, Klaus Obermayer, “Learning Preference Relations for Information Retrieval，” ICML-98 *Workshop*: *Text Categorization and Machine Learning*，pp.80-84，1998.
8) 長尾 眞，佐藤 理史，黒橋 禎夫，角田達彦，“自然言語処理 (岩波講座 ソフトウェア科学 15)，” 岩波書店，1996.
9) John Shawe-Taylor，Nello Cristianini. “Kernel Methods for Pattern Analysis，” Cambridge University Press，2004.
10) Koby Crammer, Ofer Dekel，Joseph Keshet，Shai Shalev-Shwartz，Yoram Singer. “Online Passive-Aggressive Algorithms，” Journal of Machine Learning Research, vol.7，pp.551–585，Jan. 2006.
11) Hal Daumé III. “Practical Structured Learning Techniques for Natural Language Processing，” PhD Thesis, University of Southern California, 2006.
12) 山本 悠二，増山 繁，“係り元文節からの相対的な距離を反映した統計的日本語係り受け解析，” 電子情報通信学会論文誌，vol.J93-D，no.6，pp.1036-1047，Jun. 2010.
13) 工藤 拓，松本 裕治，“カーネル法を用いた言語処理における高速化手法，” 情報処理学会論文誌，vol.45，no.9，pp.2177-2185，Sep. 2004.
14) 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，Dec. 1999.
15) 岩立将和, 浅原正幸, 松本裕治，“係り受け解析器の部分解析精度評価とその応用，” 情報処理学会 研究報告 2009-NL-189，pp.41-48，Jan. 2009.
16) Kotaro Kitagawa，Kumiko Tanaka-Ishii. “Tree-Based Deterministic Dependency Parsing -An Application to Nivre’s Method-，” In Proc. of ACL 2010, pp.189-193, 2010.
17) 飯田 龍，乾 健太郎，松本 裕治，“文脈的手がかりを考慮した機械学習による日本語ゼロ代名詞の先行詞同定，” 情報処理学会論文誌，vol.45，no.3，pp.906-918，Mar. 2004.
18) 岩立将和, 浅原正幸, 松本裕治，“トーナメントモデルをを用いた日本語係り受け解析，” 自然言語処理，vol.15，no.6，pp.169-185，Oct. 2008.
19) 山田 寛康，“Shift-Reduce 法に基づく日本語固有表現抽出，” 情報処理学会 研究報告 2007-NL-179，pp.13-18，May. 2007.
20) 岡野原 大輔，辻井 潤一，“Shift-Reduce 操作に基づく未知語を考慮した形態素解析，” 第 14 回言語処理学会年次大会論文集，pp.77-80，Mar. 2008.

付 録

A.1 提案手法の解析アルゴリズムの擬似コード

```
// 出力:  estlink:  推定された係り先文節の添字番号が格納された配列
function bilateral_parsing( w, B[], N )
estlink = [-1] * N; // 文節数分 −1 が入った配列
(estlink) = bilateral_sr_parsing( w, B, N, estlink );
(estlink) = bilateral_comp_parsing( w, B, N, estlink );
```

図 5 擬似コード - Bilateral Parsing
Fig. 5 Pseudo code for the bilateral parser.

```
// 出力:  estlink:  推定された係り先文節の添字番号が格納された配列
function bilateral_sr_parsing( w, B[], N, estlink[] )
push(stack, -1);  // −1 は番兵
push(stack,  0);
for (j = 1; j < N; j++) {
  i = pop(stack);
  while ( (i != -1)
          && ( ( (i == N - 2) && (j == N - 1) ) ||
               ( w · ω(⟨i, j⟩, B) > 0 ) ) ) {
    estlink[i] = j;  // 推定された係り先文節の添字番号を代入
    i = pop(stack);
  }
  push(stack, i);
  push(stack, j);
}
```

図 6 擬似コード - Bilateral Parsing の決定的な解析側の係り受け同定
Fig. 6 Pseudo code for the deterministic parsing of the bilateral parser.

```
// 出力:  estlink:  推定された係り先文節の添字番号が格納された配列
function bilateral_comp_parsing( w, B[], N, estlink[] )
for (i = N - 3; i >= 0; i--) {
  if (estlink[i] != -1)
    continue;  // すでに係り先が決定している場合は係り先の探索は不要
  max_scr_idx = i + 1;
  max_scr = w · ω(⟨i, i + 1⟩, B);
  j = estlink[i + 1];
  while (j != -1) {
    scr = w · ω(⟨i, j⟩, B);
    if (scr > max_scr) {
      max_scr_idx = j;
      max_scr = scr;
    }
    j = estlink[j];
  }
  estlink[i] = max_scr_idx;
}
```

図 7 擬似コード - Bilateral Parsing の相対的な比較による解析側の係り受け同定
Fig. 7 Pseudo code for the preference-based parser of the bilateral parser.