係り受け解析を伴った日本語文の語順整序

吉田 和史[†]　大野 誠寛[‡]　加藤 芳秀[††]　松原 茂樹[†]

[†] 名古屋大学大学院情報科学研究科　[‡] 名古屋大学情報基盤センター
[††] 名古屋大学情報連携統括本部

1　はじめに

日本語は語順が比較的自由であるため，語順を強く意識しなくても，意味の通じる文を書くことができる．しかし，語順に関する選好がないわけではないため，文法的には間違っていないものの読みにくい語順をもった文が作成されることがある．例えば，以下の 2 つの例文では，

1. 鈴木さんが佐藤さんが何日かかってもどうしても解けなかった問題をすぐ解いてしまった。
2. 佐藤さんが何日かかってもどうしても解けなかった問題を鈴木さんがすぐ解いてしまった。

例文 1 はそのままでは読みにくいが [1]，例文 2 のように文節を並べ替えることにより読みやすくすることができる．

本論文では，読みにくい日本語文に対して，より読みやすくなるように文節を並べ替える手法を提案する．これまでにも語順整序に関する研究はいくつか行われている．日本語を対象とした研究として，内元ら [2] は，日本語における語順決定に関する様々な要因に基づいて，統計的に語順を整える手法を提案している．また，横林ら [3] は，構文情報を用いて修飾節を入れ替える手法を提案している．外国語を対象とした研究として，Filippova ら [4] は，ドイツ語の言語的特徴を用いて語順を生成する手法を提案している．これらの手法はいずれも事前に係り受け解析を施すことを想定しており，入力文が読みにくい語順である場合に係り受け解析の精度が低下することに伴い，語順整序の精度も低下してしまう可能性がある．

一方，本手法は，係り受け構造が付与されていない文を入力とし，係り受け解析と語順整序を同時に行う．係り受けと語順の適切さを同時に考慮することにより，読みやすい語順を精度よく同定することが期待できる．新聞記事を用いた実験を行い，本手法の有効性を確認した．

2　日本語における語順と係り受け

これまでに言語学分野において，日本語の語順に関する研究調査が行われており，語順を決定する基本的要因が詳細に整理されている [1]．例えば，長い修飾句を持つ文節は前方に位置する傾向が強いといったことが指摘されている．例文 1 は,「鈴木さんが」とその係り先「解いてしまった」が遠く離れているため,「鈴木さんが」の係り先が分かりにくくなっており，読みにくい文となっている．この例は，例文 1 の係り受け構造が分かれば，例文 2 のように読みやすく語順を変更できる可能性があることを示唆している．一方，係り受け解析は一般に，新聞記事など読みやすい文に付与された係り受け構造から学習を行っているため，入力文が読みにくい語順である場合に精度が低下する可能性が高まる．そのため，例文 1 は，例文 2 のように語順を変更した後に解析した方が高精度に解析できる可能性がある．このように語順整序と係り受け解析は互いに依存しているといえる．

3　語順整序手法

本手法では，文法的には間違っていないものの読みにくい文が入力されることを想定し，その文に対して，係り受け解析を行うと同時に，より読みやすくなるような語順を同定する．なお，入力文は形態素解析と文節まとめ上げが施されていることとする．

本手法は，入力文に対する語順と係り受け構造のすべてのパターンから，最尤のパターンを探索することにより係り受け解析と語順整序の同時処理を実現する．なお，本手法では，文節の言い換えは行わず，文節を並べ替えることのみを行う．

3.1　語順整序のための確率モデル

本手法では，入力文の文節列を $B=b_1 \cdots b_n$ とするとき，$P(S|B)$ を最大とする構造 S を求める．構造 S は，語順整序後の語順 $O = \{o_{1,2}, \cdots, o_{1,n}, \cdots, o_{i,j}, \cdots, o_{n-1,n}\}$ と係り受け構造 $D = \{d_1, \cdots, d_{n-1}\}$ の二項組として定義され，$S = \langle O, D \rangle$ と書く．ここで，$o_{i,j}$ $(1 \leq i < j \leq n)$ は，2 文節間 b_i と b_j の語順整序後の順序を表し，文節 b_i が先か（$o_{i,j} = 1$），後か（$o_{i,j} = 0$）のいずれかの値をとる．また d_i は，文節 b_i を係り元の文節とする係り受け関係とする．

ある $S = \langle O, D \rangle$ に対する $P(S|B)$ を次式により計算する．

$$P(S|B) = P(O, D|B) = \sqrt{P(O|B) \times P(D|O, B)} \times \sqrt{P(D|B) \times P(O|D, B)} \quad (1)$$

この式は，以下の 2 つの式の両辺で積を取ることにより導いたものである．

$$P(O, D|B) = P(O|B) \times P(D|O, B) \quad (2)$$

$$P(O, D|B) = P(D|B) \times P(O|D, B) \quad (3)$$

なお，これらは，$P(O,D|B)$ を乗法定理により式変形を行う際に，確率式の前件部に移す順番を変えることにより導くことができる．

ここで，2 文節間の語順 $o_{i,j}$ は他の 2 文節間の語順とは互いに独立であり，かつ，係り受け関係 d_i も他の係り受け関係とは互いに独立であると仮定する．この仮定にもとづき，以下のように近似する．

$$P(O|B) \cong \prod_{i=1}^{n-1} \prod_{j=i+1}^{n} P(o_{i,j}|B) \quad (4)$$

$$P(D|O,B) \cong \prod_{i=1}^{n-1} P(d_i|O,B) \quad (5)$$

$$P(D|B) \cong \prod_{i=1}^{n-1} P(d_i|B) \quad (6)$$

$$P(O|D,B) \cong \prod_{i=1}^{n-1} \prod_{j=i+1}^{n} P(o_{i,j}|D,B) \quad (7)$$

$P(o_{i,j}|B)$ は，文節列 B において文節 b_i と文節 b_j の語順が $o_{i,j}$ になる確率を，$P(d_i|O,B)$ は，文節列 B を語順整序結果 O に従って並べ替えた後の文において，文節 b_i を係り元とする係り受け関係が d_i になる確率を，$P(d_i|B)$ は，文節列 B において文節 b_i を係り元とする係り受け関係が d_i になる確率を，$P(o_{i,j}|D,B)$ は，係り受け構造が D である文節列 B において，文節 b_i と文節 b_j の語順が $o_{i,j}$ になる確率を表す．これらの確率はいずれも最大エントロピー法により推定する．

$P(d_i|O,B)$ を推定する際は，文献 [5] の素性のうち，読点および括弧に関する素性を除くすべての素性を利用した．$P(d_i|B)$ を推定する際には，$P(d_i|O,B)$ の推定時に使用した素性のうち，文節の順番についての情報を使うことなく取得可能な素性を用いた．$P(o_{i,j}|D,B)$ を推定する際は，b_i と b_j が同一文節に係る場合については，文献 [2] で用いられた素性のうち，並列関係や意味素性に関する素性を除くすべての素性を用いた．b_i と b_j が同一文節に係らない場合は，同一文節に係る場合に使用した素性のうち，受け文節に関する素性を除くすべての素性を用いた．$P(o_{i,j}|B)$ を推定する際は，$P(o_{i,j}|D,B)$ の推定時に使用した素性のうち，係り受け情報を使うことなく取得可能な素性を用いた．

3.2 探索アルゴリズム

入力文 B に対して考えられる，O と D から成る構造 S のパターンは膨大な数であるため，効率的な探索アルゴリズムが求められる．しかし，O と D は互いに依存しているため，単純には，最適解を効率的に探索することはできない．本研究では，従来の係り受け解析で利用されてきた CYK 法を拡張し，$P(O,D|B)$ を最大とする O と D の近似解を効率よく探索する．

本研究では，文法的には間違っていない入力文を，意味を変えることなく，読みやすくなるように語順を整えることを想定している．この想定から，解を効率的に探索する上で，以下の条件を利用することができる．

1. 文の係り受け構造は，入力時の語順に対して，日本語の構文的制約（後方修飾性，非交差性，係り先の唯一性）[5] を満たす必要がある．
2. 文の係り受け構造は，語順整序後の語順に対して，日本語の構文的制約を満たす必要がある．
3. 文の係り受け構造は，語順整序の前後で同一である必要がある．

条件 1 と 3 より，入力文の語順において日本語の構文的制約を満たす係り受け構造に，D の探索空間を絞ることができる．さらに，これら絞り込んだ係り受け構造から条件 2 と 3 に基づいて導出される語順に，O の探索空間を絞ることができる．すなわち，ある係り受け構造に対して，その係り受け構造を維持しつつ，語順整序後の語順でも日本語の構文的制約を満たすように並べ替えられた語順を探索すればよい．

一方，CYK 法を利用することにより，入力文に対して，日本語の構文的制約を満たす係り受け構造を効率的に探索できることが知られている．そこで本研究では，従来の係り受け解析における CYK 法を拡張し，入力文の語順において日本語の構文的制約を満たす係り受け構造を探索すると同時に，その係り受け構造から導出可能な語順（係り受け構造を維持しつつ，語順整序後の語順でも日本語の構文的制約を満たすように変更した語順）を効率的に探索する．

3.2.1 語順整序アルゴリズム

Algorithm 1 に本手法の語順整序アルゴリズムを示す．本手法では，文節長 n の入力文に対して $n \times n$ の三角行列 $M_{i,j}(1 \leq i \leq j \leq n)$（図 1 の左図参照）を用意し，$i$ 行 j 列目の $M_{i,j}$ に，部分文節列 $B_{i,j} = b_i \cdots b_j$ に対する，語順 $O_{i,j}$ と係り受け構造 $D_{i,j}$ から成る最尤の構造 $argmax_{S_{i,j}} P(S_{i,j}|B_{i,j})$ を書き込む．本節では，説明の都合上，$S_{i,j}$ を，係り受け関係 $d_x(i \leq x \leq j)$ の系列で表すこととし，例えば，$d_i d_{i+1} \cdots d_j^0$ により，語順は b_i が 1 番目，b_{i+1} が 2 番目，…，b_j が最後となり，かつ，係り受け構造は $\{d_i, d_{i+1}, \cdots, d_{j-1}\}$ となる構造を意味することとする．なお，係り先を明示する必要があるときは，d_x^y により，文節 b_x が b_y に係る係り受け関係を示すこととする．また，d_j^0 は，部分文節列の最終文節 b_j の係り先はないことを意味する．

まず，4〜6 行目で，対角線要素 $M_{i,i}$ に d_i^0 を格納する．次に，7〜16 行目で，対角線要素 $M_{i,i}$ を始点として，対角線に沿って，右上方向に順に $M_{i,j}$ を書き込んでいく．

$M_{i,j}$ に書き込む最尤構造は以下のように探索する．まず，10〜12 行目において，*Concat* 関数により，$M_{i,k}$ と $M_{k+1,j}$ から最尤構造の候補を生成し，構造候補集合 $C_{i,j}$ に追加することを繰り返す．*Concat* 関数は，$M_{i,k}$ と $M_{k+1,j}$ の各内部の語順と係り受け構造は変更することなく，語順に関しては $M_{i,k}$ の後に $M_{k+1,j}$ を単純につなげ，係り受け構造に関しては $M_{i,k}$ と $M_{k+1,j}$ の最終文節同士を係り受け関係で結ぶことにより，$M_{i,j}$ に書き込む最尤構造の候補を 1 つを生成する関数である．すなわち，$M_{i,k} =$

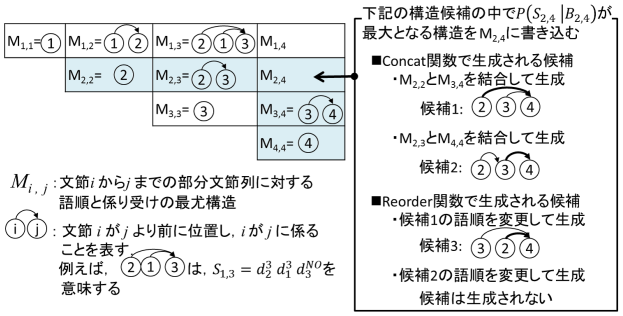

図 1: 探索アルゴリズムの実行例

$d_i \cdots d_{k-1} d_k^0$ と $M_{k+1,j} = d_{k+1} \cdots d_{j-1} d_j^0$ を入力とするとき，$d_i \cdots d_{k-1} d_k^j d_{k+1} \cdots d_{j-1} d_j^0$ を返す．11 行目における $push(A, a)$ とは，集合 A に要素 a を追加する関数である．

次に，13 行目において，*Reorder* 関数を用いて，語順を並べ替えた構造候補を生成し，それらを構造候補集合に追加する．*Reorder* 関数は，10〜12 行目で得られた構造候補集合から，1 個ずつ構造候補を取り出し，その係り受け構造を維持しつつ，語順整序後の語順でも日本語の構文的制約を満たすように語順を並べ替えた構造を生成し，それらの集合を返す関数である．なお，1 個の構造候補からは，0 個以上の構造候補が新たに生成される．さらに，14 行目において，部分文節列 $b_i \cdots b_j$ に対する, 語順 $O_{i,j}$ と係り受け構造 $D_{i,j}$ の最尤構造 $argmax_{S_{i,j} \in C_{i,j}} P(S_{i,j}|B_{i,j})$ を $M_{i,j}$ に書き込む．最後に，$M_{1,n}$ が埋まると，入力文に対する語順と係り受け構造の最尤構造として $M_{1,n}$ を出力する．

なお，本アルゴリズムにおいて，13 行目の処理を行わないと，従来の係り受け解析における CYK 法となる．

3.2.2 語順整序アルゴリズムの実行例

図 1 に，$n = 4$ のときの語順整序アルゴリズムの実行例を示す．図 1 の左図は 4×4 の三角行列であり，順に $M_{1,1}, M_{2,2}, M_{3,3}, M_{4,4}, M_{1,2}, M_{2,3}, M_{3,4}, M_{1,3}$ まで書き込まれており，次に，$M_{2,3}$ を書き込む処理が行われている様子を示している．図 1 の右図は，$M_{2,3}$ に書き込む最尤構造を求める際の処理を示す．まず，アルゴリズムの 10〜12 行目において，*Concat* 関数により構造候補を 2 つ生成する．具体的には，候補 1 は $M_{2,2}$ と $M_{3,4}$ の最終文節同士を，候補 2 は $M_{2,3}$ と $M_{4,4}$ の最終文節同士を係り受け関係で結ぶことにより生成される．次に，13 行目において，*Reorder* 関数を用いて，候補 1 と候補 2 から，係り受け構造を維持したまま，語順整序後の語順でも日本語の構文的制約を満たすように語順を並べ替えた構造を生成する．具体的には，候補 1 からは語順が異なる構造として候補 2 が一つ生成され（b_4 に係る b_2 と b_3 の順序を入れ替えて生成），候補 3 からは係り受け構造の形から語順が異なる構造は生成されない．このようにして生成された 3 つの構造候補のうち，$P(S_{2,4}|B) = P(O_{2,4}, D_{2,4}|B_{2,4})$ を最大とする構造を $M_{2,4}$ に書き込む．

Algorithm 1 語順整序アルゴリズム

```
1: input B_{1,n} = b_1 ··· b_n // 入力文節列
2: set M_{i,j}(1 ≤ i ≤ j ≤ n) // 三角行列
3: set C_{i,j} // 構造候補集合
4: for i = 1 to n do
5:     M_{i,i} = d_i^0
6: end for
7: for d = 1 to n − 1 do
8:     for i = 1 to n − d do
9:         j = i + d
10:        for k = i to j − 1 do
11:            push(C_{i,j}, Concat(M_{i,k}, M_{k+1,j}))
12:        end for
13:        C_{i,j} = C_{i,j} ∪ Reorder(C_{i,j})
14:        M_{i,j} = argmax_{S_{i,j} ∈ C_{i,j}} P(S_{i,j}|B_{i,j})
15:    end for
16: end for
17: return M_{1,n}
```

4 評価実験

4.1 実験概要

本実験では，京大テキストコーパス [6] に収録されている新聞記事文に対して，係り受け構造を維持しつつ語順を変更した文をテストデータとして用いた．テストデータには，人間が作文した文を使用することが考えられるが，本実験では，問題の焦点を語順に絞ることを考慮し，文意は取れるものの読みにくい文を擬似的に作成することとした．図 2 にテストデータの作成例を示す．文末から順に，複数の文節から係られる文節（「勢力だ。」や「通った」）を起点として，その文節に係る部分係り受け構造の順序をランダムに変更することを繰り返すことにより作成した．読点は文の途中の区切りに挿入される記号であり，読点の位置を変えると文の意味が異なる可能性が生じる．そこで本研究では学習データとテストデータの文中から読点を取り除いた．また，文中に句読点以外の記号が含まれる文は本研究の対象外とし，テストデータから除外した．

このようにして，京大テキストコーパスの 1 月 9 日分の新聞記事から，擬似的に作成した文（865 文，7,620 文節）をテストデータとした．なお，学習データには，7 日分（1 月 1 日，3〜8 日）の新聞記事（7,976 文）を用いた．

語順整序の評価では，文献 [2] と同様に，文単位正解率（語順整序後の語順が元の文と完全に一致している文の割合）と 2 文節単位正解率（2 文節ずつ取り上げた時の文節の順序関係が元の文のそれと一致しているものの割合）を測定した．係り受け解析の評価では，文献 [5] と同様に，文単位正解率（解析結果の係り受け構造が正解と完全に一致している文の割合）と係り受け単位正解率（解析結果と正解で一致している係り受け関係の割合）を測定した．

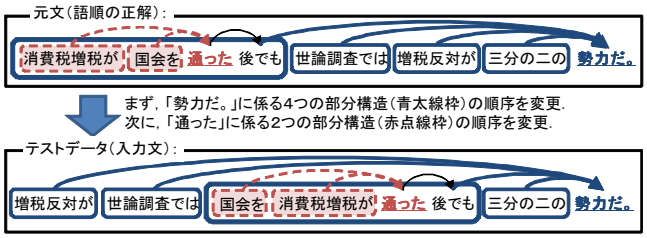

図 2: テストデータの作成例

表 1: 実験結果（語順整序）

	2 文節単位正解率	文単位正解率
本手法	77.3% (30,190/38,838)	25.7% (222/865)
ベースライン 1	75.4% (29,279/38,838)	23.8% (206/865)
ベースライン 2	74.8% (29,067/38,838)	23.5% (203/865)
テストデータ	61.5% (23,886/38,838)	8.0% (69/865)

比較のために，2 つのベースラインを設けた．いずれも，まず係り受け解析を行い，その後に文献 [2] の手法により語順整序を行うものである．係り受け解析に文献 [5] の手法を用いる場合をベースライン 1，CaboCha[7] を用いる場合をベースライン 2 とする．なお，両ベースラインにおいて，語順を推定する際に使用した素性は，本手法において $P(o_{i,j}|D,B)$ を推定する際に使用した素性と同一である．また，ベースライン 1 において，係り受け確率を推定する際に使用した素性は，本手法において $P(d_i|O,B)$ を推定する際に使用した素性と同一である．

4.2 実験結果

本手法及び各ベースラインの語順整序結果を表 1 に示す．最下位行は，テストデータの語順（語順整序前の語順）で測定した語順正解率である．2 文節単位と文単位のいずれの指標においても，本手法は最も高い正解率を達成した．本手法と両ベースラインとの間でマクネマー検定を実施したところ，文単位正解率では有意差は認められなかったが，2 文節単位正解率では有意差が認められた（$p < 0.05$）．本手法による語順整序結果の成功例（1 文全体の語順が正解と完全に一致した例）を図 3 に示す．読みにくい文に対して，読みやすい語順に修正できていることがわかる．

次に，係り受け解析の実験結果を表 2 に示す．本手法の文単位正解率は，両ベースラインと比べて，有意に高い結果となった（$p < 0.05$）．一方，係り受け単位正解率では，本手法はベースライン 2 と比べ，有意に低い結果となったが，ベースライン 1 との間では有意差は認められなかった（$p < 0.05$）．以上より，読みにくい文に対する語順整序および係り受け解析において，本手法が有効であることを確認した．

5 おわりに

本論文では語順整序と係り受け解析を同時に実行する手法について述べた．日本語の構文的成約を利用した探索空間の絞り込みを行い，従来の係り受け解析で利用されてきた CYK 法を改良したアルゴリズムによって，効率的に最適解の探索を行う．京大コーパスを使用した評価実験により，提案手法の有効性を確認した．今後は人間によって作成された文をテストデータとして使用し，評価実験を行う予定である．

謝辞 本研究は一部，科研費挑戦的萌芽研究 No.24650066，及び，科研費若手研究 (B) No.25730134 により実施した．

例1
【入力文】
3回目の 交渉を 九日に 持つ ことを 披露宴後 会見した 野茂は 明らかにした。
【語順整序結果】
披露宴後 会見した 野茂は 九日に 3回目の 交渉を 持つ ことを 明らかにした。

例2
【入力文】
戦争を いつの 世代も 美化すべきでないと 戦争を 体験した ことは 私自身 ないが 思う。
【語順整序結果】
私自身 戦争を 体験した ことは ないが いつの 世代も 戦争を 美化すべきでないと 思う。

例3
【入力文】
軽い 練習を リヤド市内で 七日 午後 日本は 行った。
【語順整序結果】
日本は 七日 午後 リヤド市内で 軽い 練習を 行った。

図 3: 語順整序結果の成功例

表 2: 実験結果（係り受け解析）

	係り受け単位正解率	文単位正解率
本手法	78.4% (5,293/6,755)	35.3% (305/865)
ベースライン 1	79.2% (5,350/6,755)	31.6% (273/865)
ベースライン 2	81.2% (5,487/6,755)	32.1% (278/865)

参考文献

[1] 日本語記述文法研究会．現代日本語文法 7．くろしお出版, 2009.

[2] 内元ら．コーパスからの語順の学習．自然言語処理, Vol. 7, No. 4, pp. 163–180, 2000.

[3] 横林ら．係り受けの複雑さの指標に基づく文の書き換え候補の生成と推敲支援への応用．情処学論, Vol. 45, No. 5, pp. 1451–1459, 2004.

[4] K. Filippova and M. Strube. Generating constituent order in German clauses. In *Proc. of the 45th Annual Meeting of the Association of Computational Linguistics*, pp. 320–327, 2007.

[5] 内元ら．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．情処学論, Vol. 40, No. 9, pp. 3397–3407, 1999.

[6] 黒橋, 長尾．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 回年次大会論文集, pp. 115–118, 1997.

[7] 工藤, 松本．チャンキングの段階適用による日本語係り受け解析．情処学論, Vol. 43, No. 6, pp. 1834–1842, 2002.