

# 係り受け解析との統合に基づく日本語文の語順整序

吉田 和史<sup>1,a)</sup> 大野 誠寛<sup>2,b)</sup> 加藤 芳秀<sup>3,c)</sup> 松原 茂樹<sup>1,d)</sup>

**概要:** 本稿では、読みにくい語順の日本語文に対して、読みやすい語順となるように文節を並べ替える手法を提案する。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。具体的には、語順を整えた後に係り受け解析を施すことを表した確率モデルと、その逆の手順を表した確率モデルとの加重相乗平均をとったモデルにより、最尤の係り受け構造と語順を同定する。重みを変えることにより、入力文の語順の適切さに応じた尤度計算を行うことができる。また、本手法は、CYK 法を拡張することにより、係り受け構造と語順の最尤解を効率的に探索することができる。新聞記事から、日本語母語話者でも書きそうな読みにくい語順を持つ文を 552 文作成し、それらを用いて評価実験を行った結果、本手法の有効性を確認した。

**キーワード:** 語順整序, 係り受け解析, 推敲支援, 語順, 生成

## 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、実際には語順に関して選好が存在しているため、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解けなかった問題をすぐ解いてしまった。
- (2) 佐藤さんが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。

読みにくい語順を自動的に整えるという語順整序に関する研究は、推敲支援や文生成などに応用することを目的として、これまでもいくつか行われている。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、日本語の推敲支援のために、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。その他、日本語だけでなく、外国語における適切な語順を推定する研究もいくつか存在する [4], [5], [6], [7], [8]。これらの手法は、いずれも事前に係り受け解析を施し、正確な構文情報が得られることを想定している。しかし、入力文が読みにくい語順である場合、係り受け解析の精度は低下する傾向にあり、その影響を受けて、語順整序の精度も低下するという問題がある。なお、最近では、統計的機械翻訳の性能向上を目的とした語順整序に関する研究が盛んに行われている [9], [10], [11], [12], [13]、これらの研究は、原言語と目標言語間の語順の違いを捉えるために、双方の言語情報を利用している。したがって、単言語の可読性を向上させることを目的とした研究とは問題設定が異なる。

本論文では、推敲支援のための要素技術として、読みにくい語順をもった日本語文に対して、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。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。係り受けと語順の尤度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できる。評価実験では、新聞記事文の語順を機械的に変更した文から人手で選別することにより、日本語母語話者でも書きそうな読みにくい語順を持つ文を 552 文作成して、それらに対して本手法により語順整序を実行した。元の新聞記事文の語順とどの程度一致しているか測定した結果、本手法は、比較のために設定した2つのベースラインと比べ、高い一致度を達成しており、本手法の有効性を確認した。

<sup>1</sup> 名古屋大学大学院情報科学研究科  
Graduate School of Information Science, Nagoya University, Japan

<sup>2</sup> 名古屋大学情報基盤センター  
Information Technology Center, Nagoya University, Japan

<sup>3</sup> 名古屋大学情報連携統括本部  
Information & Communications, Nagoya University, Japan

a) yoshida@db.ss.is.nagoya-u.ac.jp

b) ohno@nagoya-u.jp

c) yoshihide@icts.nagoya-u.ac.jp

d) matubara@nagoya-u.jp

本論文の構成を以下に示す. 2節で日本語における語順整序と係り受け解析の関係性を述べる. 3節では提案する語順整序手法を示す. 4節で評価実験について報告し, 続く5節では, 提案手法について考察を与える.

## 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高ま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高い. 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, 意味は伝わるものの読みにくい語順をもった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, 読みやすい語順を同定する. 係り受け解析と語順整序の同時実行は, 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実現する. なお, 本研究では, 入力文は形態素解析と文節まとめ上げが事前に施されていることとし, 文節の言い換えは行わず, 文節間の並べ替えのみ行う. また, 入力文には倒置現象は含まれないものとする.

### 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.  $S$  は, 語順整序後の語順  $O = \{o_{1,2}, o_{1,3}, \dots, o_{1,n}, o_{2,3}, \dots, o_{i,j}, \dots, o_{n-1,n}\}$  と係り受け構造  $D = \{d_1, d_2, \dots, d_{n-1}\}$  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  $b_i$  と  $b_j$  との間における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  $P(S|B)$  は式 (1) により計算する.

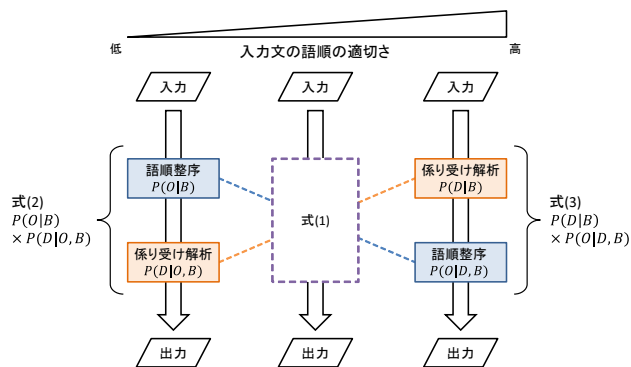


図 1 式 (1)~(3) の関係

Fig. 1 Relationships among formulas (1) to (3).

$$P(S|B) = P(O, D|B) = \{P(O|B) \times P(D|O, B)\}^\alpha \times \{P(D|B) \times P(O|D, B)\}^{1-\alpha} \quad (0 \leq \alpha \leq 1) \quad (1)$$

式 (1) は, 以下に示す式 (2) と式 (3) の加重相乗平均をとったものであり \*1, 重み  $\alpha$  ( $0 \leq \alpha \leq 1$ ) によって, どちらを重視するかという度合いが設定される.

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

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

ここで, 式 (2) は語順を整えた後に係り受け解析を行うという順番で, 式 (3) は, 係り受け解析を施した後に語順整序を行うという順番で, それぞれ  $P(O, D|B)$  を乗法定理により展開したものである. 図 1 に, 式 (1)~(3) の関係を示した概念図を示す. 入力文の語順の適切さが低いほど, 最初に語順を整えた上で係り受け解析を行った方が  $S = \langle O, D \rangle$  を精度よく同定できると考えられ, 式 (2) により  $P(O, D|B)$  を計算した値を重視することが考えられる. 逆に, 入力文の語順の適切さが高いほど, まず係り受け解析を行い, その上で, 語順を整える方がよいと考えられ, 式 (3) により計算した値を重視することが考えられる. 一方, 推敲支援を必要とする人が書いた文がどの程度, 適切な語順を持っているかは明らかではない. そこで, 本手法では, 式 (2) と式 (3) を重視する度合いを重みにより調整することを考え, 式 (1) により  $P(O, D|B)$  を計算することとした. 入力文の語順の適切さに応じて  $\alpha$  を設定することにより, 高い精度での語順の推定が期待できる.

なお, 式 (1)~(3) は, 真の確率分布を求めることができれば同じ値になると考えられるが, 実際には, コーパスから学習して近似的に推定することになるため, 異なる値となる. また,  $y = a^x$  は,  $0 \leq a \leq 1$  のとき, 区間  $0 \leq x \leq 1$  において単調減少となるため, 式 (1) では,  $\alpha$  が小さいほど, 式 (2) が重視されることになる.

式 (1) の最右辺における各確率の計算は, 2文節間の語

\*1 事前実験の結果, 加重相乗平均をとるよりも高い精度を示したため, 加重相乗平均をとることとした.

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とにより \*2, 以下のように近似できる。

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

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

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

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

$P(o_{i,j}|B)$  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。

$P(o_{i,j}|D, B)$  を推定する際は、文献 [2] で用いられた素性のうち、並列関係に関する素性を除くすべての素性を用いる。 $P(o_{i,j}|B)$  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る。 $P(d_i|O, B)$  を推定する際は、文献 [14] 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する。 $P(d_i|B)$  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る。

### 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、最尤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、 $P(O, D|B)$  を最大にする  $O$  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

\*2 これらは実際には互いに独立しているわけではないが、本研究では、計算の都合上、独立を仮定した。なお、 $o_{i,j}$  や  $d_i$  の組み合わせは、存在し得る組み合わせのみを探索するため、矛盾する組み合わせが  $O$  や  $D$  の解として選ばれることはない。

の構文的制約（後方修飾性、非交差性、係り先の唯一性）[14] 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 \*3。

条件 (1) と条件 (3) より、 $D$  の探索空間は、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) と条件 (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、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は、CYK 法により探索できる。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

図 2 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 4 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  $D_{i,j}$  の最尤構造  $\text{argmax}_{S_{i,j}} P(S_{i,j}|B_{i,j})$  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（順序付き集合）で表すこととする。例えば、 $S_{1,3} = d_3^3 d_1^3 d_3^0$  は、 $b_2$  が 1 番目、 $b_1$  が 2 番目、 $b_3$  が最後という語順で、かつ、 $\{d_1^3, d_2^3, d_3^0\}$  という係り受け構造であることを意味する。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、7~15 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、 $\text{concatReorder}$  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。次に、13 行目において、部分文節列  $b_i \cdots b_j$  に対する、最尤構造  $\text{argmax}_{S_{i,j} \in C_{i,j}} P(S_{i,j}|B_{i,j})$  を  $M_{i,j}$  に書き込む。

\*3 本研究では、係り受け構造を 2 文節間の係り受け関係の（順序なし）集合で定義している。2 つの係り受け構造が、集合として同じものであれば、両者の係り受け構造は同一であるとする。

```

1: input  $B_{1,n} := b_1 \cdots b_n$  // 入力文節列
2: set  $M_{i,j} := null$  ( $1 \leq i \leq j \leq n$ ) // 三角行列
3: set  $C_{i,j} := null$  ( $1 \leq i \leq j \leq n$ ) // 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:       $C_{i,j} := C_{i,j} \cup \text{concatReorder}(M_{i,k}, M_{k+1,j})$ 
12:    end for
13:     $M_{i,j} := \text{argmax}_{S_{i,j} \in C_{i,j}} P(S_{i,j} | B_{i,j})$ 
14:  end for
15: end for
16: return  $M_{1,n}$ 

```

図 2 語順整序アルゴリズム

Fig. 2 Word reordering algorithm.

```

1: function  $\text{concatReorder}(S_1, S_2)$ 
2: begin
3:   set  $C := null$  // 構造候補集合
4:   set  $z := \text{last}(S_2)$  //  $\text{last}(S_2)$  は  $S_2$  の最終文節番号を表す.
5:
6:   // 接続プロセス
7:   // 係り受けに関する結合
8:   //  $S_1$  の最終文節の係り先を  $b_z$  に変更したものを  $S'_1$  とする.
9:    $S'_1 := \text{changeLastDep}(z, S_1)$ 
10:  // 構造間の接続による語順決定
11:   $C := \{S'_1 S_2\}$ 
12:
13:  // 語順変更プロセス
14:  for each  $d \in \{d_x^y \mid y = z, d_x^y \in S_2\}$  do
15:    //  $S_2$  を  $d$  の直後で分割し, 左側を  $S_2^L$ , 右側を  $S_2^R$  とする.
16:     $(S_2^L, S_2^R) := \text{divide}(d, S_2)$ 
17:     $C := C \cup \{S_2^L S'_1 S_2^R\}$ 
18:  end for
19:
20:  return  $C$ 
21: end

```

図 3  $\text{concatReorder}$  関数

Fig. 3 Function:  $\text{concatReorder}$ .

最後に,  $M_{1,n}$  が埋まると, 16 行目で, 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.

図 3 に,  $\text{concatReorder}$  関数のアルゴリズムを示す.  $\text{concatReorder}$  関数は, 引数として, 2 つの構造  $S_1 (= M_{i,k})$  と  $S_2 (= M_{k+1,j})$  を受け取り,  $M_{i,j}$  に書き込む構造の候補集合  $C$  を返す関数である. この関数は, 接続プロセスと語順変更プロセスの 2 つのプロセスから構成される.

接続プロセスでは, まず 9 行目において,  $S_1$  の最終文節と  $S_2$  の最終文節とを係り受け関係で結ぶことによって, 係り受けに関する結合を行う. 具体的には,  $\text{changeLastDep}$  関数によって,  $S_1$  の最終文節の係り先を  $S_2$  の最終文節  $b_z$  に変更した構造を作成し, それを  $S'_1$  とする. 例えば,  $S_1 = d_i d_{i+1} \cdots d_{k-1} d_k^0$  であるとき,  $\text{changeLastDep}(S_1, z)$  は  $d_i d_{i+1} \cdots d_{k-1} d_k^z$  を返す. 次に, 11 行目で 2 つの構造間

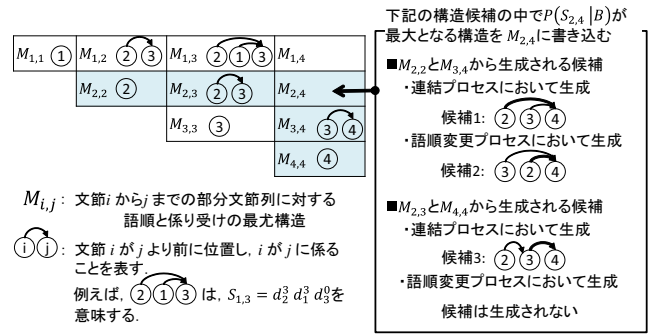


図 4 探索アルゴリズムの実行例

Fig. 4 Execution example of our search algorithm.

を接続することにより, 生成する構造候補の語順を決定する. 具体的には,  $S'_1$  と  $S_2$  の各内部の構造 (語順と係り受け) は変更することなく,  $S'_1$  の後に  $S_2$  を単純に接続させた " $S'_1 S_2$ " を  $M_{i,j}$  に書き込む構造の候補として,  $C$  に追加する.

次に語順変更プロセスにおいて, 接続プロセスで生成された構造候補の語順を並べ替えることにより, 0 個以上の構造候補を新たに生成する. 語順の変更は以下の制約に基づいて実行される.

- (1) 生成される構造は, 語順変更の前後で係り受け構造を維持しつつ, 変更後の語順においても日本語の構文的制約を満たしていなければならない.
- (2)  $S'_1$  と  $S_2$  の各内部の構造 (語順と係り受け) は変更してはならない.

このことは, 14~18 行目の処理により実現される. まず 14 行目で,  $S_2$  の内部において,  $S_2$  の最終文節に係っている文節を探索する. そして, そのような文節が見つかるたびに, 16~17 行目で, その文節の直後に  $S'_1$  を移動させた構造 (17 行目の " $S_2^L S'_1 S_2^R$ ") を生成し,  $C$  に追加することを繰り返す. したがって, 語順変更プロセスで生成される候補の数は,  $S_2$  の内部において,  $S_2$  の最終文節に係る文節の数に等しくなる.

なお, 図 2 において,  $\text{concatReorder}$  関数を  $\text{concat}$  関数に変更すると, 従来の係り受け解析における CYK 法となる.  $\text{concat}$  関数は,  $\text{concatReorder}$  関数において, 語順変更プロセス (図 3 の 13~18 行目) を除いた処理を行う関数である.

### 3.2.2 語順整序アルゴリズムの実行例

図 4 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 4 の左図は  $4 \times 4$  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,  $M_{2,4}$  を書き込む処理が行われている様子を示している. 図 4 の右図は,  $M_{2,4}$  に書き込む最尤構造を求める際の処理を表す. まず, 図 2 の 10~12 行目において,  $\text{concatReorder}$  関数により構造候補が 3 つ生成される. 具体的には, まず,  $\text{concatReorder}(M_{2,2}, M_{3,4})$  の連

接プロセスにおいて、 $M_{2,2}$ の最終要素  $d_2^0$  を  $d_2^1$  に変更した  $M'_{2,2}$  と、 $M_{3,4}$  を接続することにより、候補1が生成される。次に、 $concatReorder(M_{2,2}, M_{3,4})$ の語順変更プロセスにおいて、 $M'_{2,2}$ を  $b_3^0$ の直後に移動することにより、候補2が生成される。最後に、 $concatReorder(M_{2,3}, M_{4,4})$ の接続プロセスにより候補3が生成される。なお、 $concatReorder(M_{2,3}, M_{4,4})$ の語順変更プロセスでは、候補3の係り受け構造の形から、語順が異なる候補は生成されない。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。

## 4. 評価実験

日本語文の語順整序における本手法の有効性を評価するため、新聞記事を用いて語順整序実験を実施した。新聞記事中の文から擬似的に作成した読みにくい語順の文に対して本手法を適用し、元の新聞記事文の語順をどの程度再現できるかを評価した。

### 4.1 評価用データの作成

推敲支援への応用を念頭に置くと、母語話者が自然に作成した読みにくい語順の文に対して本手法を適用し、その有効性を評価することが考えられる。しかし、母語話者が自然に作成した読みにくい語順の文と、その語順を読みやすく人手で修正した文のペアを大量に集めることは容易ではなく、現時点において、そのようなデータは存在していない。また、母語話者が自然に作成した文には、語順以外にも読みやすさを低下する要因が存在しており、問題の焦点を語順に絞った評価を行うことは難しいといえる。

そこで、本研究では、新聞記事中の文は読みやすい語順で書かれているものと想定し、新聞記事文から、文意は取れるものの読みにくい語順の文を擬似的に作成することにより、評価用データを収集した。ただし、語順を機械的にランダムに変更しただけでは、母語話者が到底書きそうにない語順となる可能性があるため、人の判断を介在させることとした。すなわち、ランダムに語順を変更した文に対して、人手で判断し、母語話者でも書きそうな文である場合に限り、評価用データとして採用した。具体的な新聞記事データとしては、京大テキストコーパス [15] (Version 2) を用いた。

評価用データの構築手順を図5に示す。以下に、各手順の説明を示す。

(1) 京大テキストコーパスから1文を取り出す。ただし、以下の文は対象外とした。

- 文中に句読点以外の記号が含まれる文<sup>\*4</sup>
- 3.2節の条件(1)と条件(3)を満たす形では、語順を

<sup>\*4</sup> 本研究では、開始記号と終了記号の組への対処は考慮していないため、対象外とした。

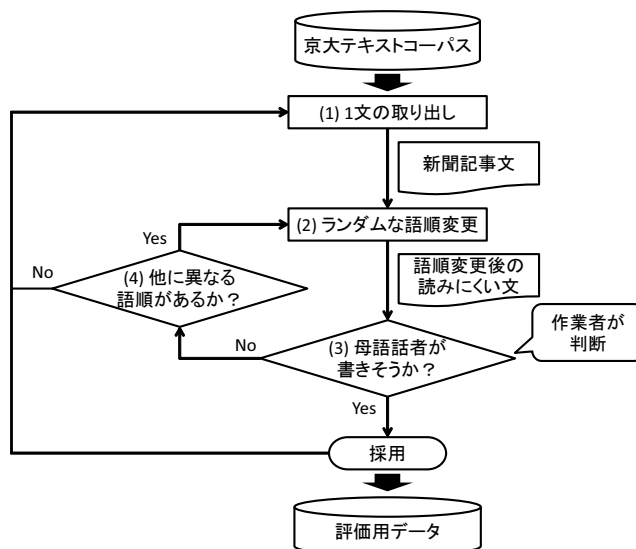


図5 評価用データの構築手順

Fig. 5 Construction procedure of evaluation data.

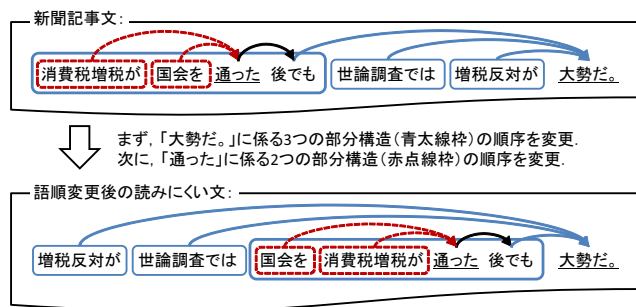


図6 ランダムな語順変更の例

Fig. 6 Example of random reordering.

変更できない文

- (2) 3.2節の条件(1)と条件(3)を満たしつつ、語順をランダムに変更することにより、読みにくい語順の文を自動生成する。このランダムな語順変更の例を図6に示す。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。なお、元の文の語順や、これまでに不採用となった語順とは異なる語順を持った文が生成されるまで繰り返す。
- (3) 手順(2)で自動生成された文が母語話者でも書きそうな文であるか否かを作業員1名が判断する。Yesならば、その語順をもった文を評価用データとして採用し、手順(1)に戻る。Noならば、その語順は不採用とし、手順(4)に進む。なお、作業員には日本語の語順に関する傾向をまとめたマニュアルとして文献[1]を提示し、適宜参照してもらった。
- (4) 元の文の語順、及び、これまでに不採用となった語順の他に、異なる語順を持つ文を手順(2)により新たに生成できるか否かを判定する。Yesならば、手順(2)

表 1 評価用データの規模

Table 1 Size of evaluation data.

文数	552
文節数	4,906

に進む。Noならば、手順(1)に戻る。

上記の手順を、京大テキストコーパスに収録されている毎日新聞1995年1月9日の記事中の文に対して適用し、評価用データを構築した。構築した評価用データの規模を表1に示す。

#### 4.2 実験概要

本手法は式(1)における重み $\alpha$ を事前に決定しておく必要がある。そのため、前節で構築した評価用データ552文を用いた5分割交差検定を実施した。すなわち、552文をランダムに5セットに分割し、そのうちの4セットをヘルドアウトデータとして用いて $\alpha$ を決定した後で、その $\alpha$ により、残りの1セットに対して語順整序を実施することを、5回繰り返した。ヘルドアウトデータを用いた $\alpha$ の決定では、以下で述べる2文節単位の語順整序正解率を最大とする $\alpha$ の値を小数点第二位まで求めた。具体的には、まず0.1刻みで変化させて $\alpha$ を小数点第一位まで求め、その $\alpha$ を中心とした前後0.1の区間において、次は0.01刻みで $\alpha$ を変化させることにより、語順整序正解率を最大とする $\alpha$ を探索した。式(4)~(7)の各確率を推定するための学習データには、京大テキストコーパスに収録された7日分(1月1日、3~8日)の新聞記事文(7,976文)を固定して用いた。なお、学習のための最大エントロピー法のツールとしては、文献[16]のものを利用した。また、文節が移動することに伴って、その文節に付随した読点の位置が変わると、文の意味が変わる可能性があるため、本研究では読点を取り扱わないこととし、学習データと評価用データの文中から読点を予め取り除いた。

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。

比較のために、以下2つのベースラインを設けた。

**ベースライン1** 文献[14]の手法により係り受け解析を行い、その後、文献[2]の手法により語順整序を行う。

**ベースライン2** CaboCha[17]により係り受け解析を行い、その後、文献[2]の手法により語順整序を行う。

なお、両ベースラインの実験では、本手法の確率モデルの学習データと同じものを用いて各モデルを学習し、上述の評価用データ全552文に対して語順整序を各1回実施した。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

表 2 各セットの語順整序結果

Table 2 Experimental results for each set.

	$\alpha$	2文節単位正解率	文単位正解率
セット1	0.66	83.02% (3,790/4,565)	32.73% (36/110)
セット2	0.75	83.83% (4,308/5,139)	29.09% (32/110)
セット3	0.66	83.30% (3,862/4,636)	32.73% (36/110)
セット4	0.66	81.89% (3,496/4,269)	28.83% (32/111)
セット5	0.75	86.91% (4,018/4,623)	31.53% (35/111)

表 3 実験結果(語順整序)

Table 3 Experimental results (word reordering).

	2文節単位正解率	文単位正解率
本手法	83.82% (19,474/23,232)	30.98% (171/552)
ベースライン1	82.39% (19,140/23,232)	26.99% (149/552)
ベースライン2	83.35% (19,365/23,232)	26.63% (147/552)
語順整序なし	76.78% (17,838/23,232)	0% (0/552)

用した素性と同一である。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。また、ベースライン2のCaboChaは、学習データにより学習し直している。

#### 4.3 実験結果

交差検定における各セットに対する本手法の語順整序結果(両正解率)と、その時に用いた $\alpha$ の値を表2に示す。例えば、セット1の行の $\alpha$ の値は、セット2から4までの4セットを合わせたヘルドアウトデータにおいて、2文節単位の語順整序正解率を最大とする $\alpha$ として求めたものになる。 $\alpha$ の値は、ヘルドアウトデータによって異なり、0.66または0.75となった。

本手法及び各ベースラインの語順整序結果を表3に示す。本手法の両正解率は、5つの各セットにおける正解率のマイクロ平均を取ることにより求めた。最下位行は、評価用データの語順(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\*5、文単位正解率では両ベースラインとの間に有意差が認められた( $p < 0.05$ )。2文節単位正解率に関しては、ベースライン2との間には有意差が認められなかったものの( $p > 0.05$ )、ベースライン1との間に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7に示す。読みにくい語順を持った入力文に対して、読みやすい語順に修正できていることがわかる。

\*5 本論文の以下で示す統計的検定は全て同じ方法で実施した。

<p>例1</p> <p>【入力文】 麻薬売買にも手を越境前の生活費を得るため出す。</p> <p>【語順整序結果】 越境前の生活費を得るため麻薬売買にも手を出す。</p>
<p>例2</p> <p>【入力文】 生産量は少なくなるが高品質のブドウをブドウの樹液を濃縮し収穫するためだ。</p> <p>【語順整序結果】 生産量は少なくなるがブドウの樹液を濃縮し高品質のブドウを収穫するためだ。</p>
<p>例3</p> <p>【入力文】 コペンハーゲンからの報道によるとチェチェン紛争への軍事介入を理由に七日デンマーク政府はロシアとの昨年九月に結んだ軍事協力協定を一時的に凍結した。</p> <p>【語順整序結果】 コペンハーゲンからの報道によるとデンマーク政府は七日チェチェン紛争への軍事介入を理由に昨年九月に結んだロシアとの軍事協力協定を一時的に凍結した。</p>

図 7 語順整序結果の成功例

Fig. 7 Examples of sentences correctly reordered by our method.

表 4 入力時の順序関係が適切な場合と不適な場合に分けて測定した、2文節単位の語順整序正解率

Table 4 Agreement rates calculated separately for the case that the word order between two *bunsetsus* in the input sentence is appropriate and the case that the order is inappropriate.

	$R_{F \rightarrow T}$	$R_{T \rightarrow T}$
本手法	66.30% (3,576/5,394)	88.77% (15,834/17,838)
ベースライン 1	52.65% (2,840/5,394)	91.38% (16,300/17,838)
ベースライン 2	51.67% (2,787/5,394)	92.94% (16,578/17,838)

#### 4.4 語順整序結果の分析

高い精度での語順整序を実現するためには、入力文中のあらゆる2文節間の順序関係のうち、適切なものは変更しないということと、不適切なものを適切に変更するということの両面で高い精度を達成する必要がある。そこで本節では、入力文中の2文節間の順序関係を適切な場合と不適な場合に分けて、各場合に対する本手法およびベースラインの性能を分析する。

表 4 に、評価用データ 552 文の各文における全ての2文節間の順序関係を、入力時において正解の語順と一致するものと、一致しないものに分けて、2文節単位の語順整序正解率を測定した結果を手法ごとに示す。 $R_{F \rightarrow T}$  は、入力時において正解と一致しない2文節間の順序関係のうち、正しく語順を変更できたものの割合を、 $R_{T \rightarrow T}$  は、入力時において正解と一致する2文節間の順序関係のうち、正しい語順のまま出力できたものの割合をそれぞれ表している。

本手法の  $R_{F \rightarrow T}$  は、両ベースラインと比べて、最も高い結果となった。したがって、本手法は、両ベースラインよりも、2文節間の順序関係が入力時において不適切な場合に、その順序関係を適切に変更する能力が高いといえる。例えば、図 7 に示す例では、入力時において不適切な2文

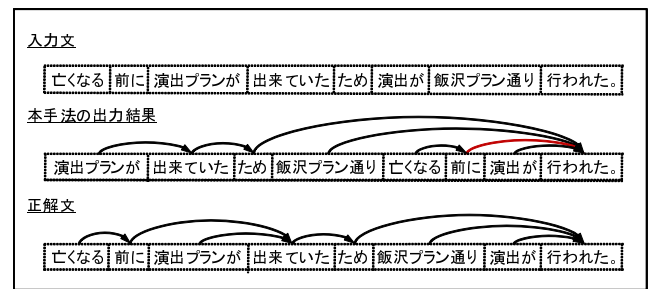


図 8 語順整序結果の誤り例

Fig. 8 Example of sentences incorrectly reordered by our method.

節間の順序関係に対して、本手法はいずれも正しく変更できたが、両ベースラインでは失敗していた。

一方で本手法は、両ベースラインと比べて、入力時には正しかった2文節間の順序関係を誤って変更することが多かった。本手法とベースライン2との比較において、入力時には正しかった2文節間の順序関係を、本手法だけが誤って変更していた例を図 8 に示す。ベースライン2はすべての語順、係り受け関係の同定に成功していたが、本手法は、入力時点で正解していた8つの2文節間の順序関係に対して、語順整序を施すことによって、それらを誤って変更していた。この語順整序誤りを引き起こした原因は、本手法が係り受け解析においても、「前に」の係り先が「行われた。」であると誤って解析していることに隠されていると考えられる。図 8 の入力文に対して単純に係り受け解析を施す場合、「前に」の係り先を正しく同定することはそれほど難しいことではないと考えられる。その理由は、入力時の語順においては、「前に」と「行われた。」の距離は遠く離れており、かつ、それら2文節の間に、「前に」の係り先として正しい文節「出来ていた」が存在しているためである。実際に、ベースライン2で用いた CaboCha では、「前に」の係り先が「出来ていた」であると正しく同定できている。しかし、本手法は、係り受け構造を探索すると同時に、「前に」が「行われた。」に係っていても自然であるような語順も探索するため、この影響を受けて、「前に」の係り先を誤ることになり、それに伴い、語順整序も誤ったと考えられる。

図 8 の例は、係り受けと語順を同時に探索するという本手法の特徴から生じる副作用が表れた例といえるが、一方で本手法は、重み  $\alpha$  を設定することにより、探索される構造の尤度を変更することができるという特徴ももつ。そのため、入力時には正しかった2文節の順序関係を誤って変更することが多かった文に対しては、式 (3) の重みを大きくする、すなわち、 $\alpha$  をさらに高く設定することによって、正しく語順整序できる可能性がある。そこで、与える  $\alpha$  を 0.66 から 0.83<sup>\*6</sup> に変更し、再び語順整序したところ、

\*6  $\alpha$  を 0.01 刻みで変更しながら語順整序を行い、適切な出力結果が得られる  $\alpha$  を探索した。

表 5 実験結果 (係り受け解析)

Table 5 Experimental results (dependency parsing).

	係り受け単位正解率	文単位正解率
本手法	83.39% (3,631/4,354)	40.04% (221/552)
ベースライン 1	84.75% (3,690/4,354)	36.78% (203/552)
ベースライン 2	86.08% (3,748/4,354)	37.50% (207/552)

すべての係り受け関係の同定に成功し、以下の語順が得られた。

演出プランが 亡くなる 前に 出来ていた ため 飯沢  
プラン通り 演出が 行われた。

$\alpha$  を変更した後の語順整序結果では、文単位でこそ正解しなかったものの、「亡くなる」と「前に」の配置は改善された。このことは、文ごとに最適な  $\alpha$  を設定することができれば、係り受け構造と語順を同時に探索することによる副作用の影響を小さくできることを示唆している。

## 5. 考察

### 5.1 係り受け解析に関する考察

本手法は、語順整序と同時に係り受け解析を行っており、係り受け解析器とみることもできる。そこで、4 節の実験結果に対して、本手法と両ベースラインの係り受け解析精度を測定し、比較評価した。

係り受け解析の評価には、文献 [14] 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) を用いた。

係り受け解析の実験結果を表 5 に示す。本手法の両正解率は、語順整序結果と同様に、5 つの各セットにおける正解率のマイクロ平均である。文単位正解率では、本手法は両ベースラインよりも上回ったものの、有意差は認められなかった ( $p > 0.05$ )。一方、係り受け単位正解率では、両ベースラインと比べて有意に低い結果となった ( $p < 0.05$ )。

### 5.2 最適な $\alpha$ における語順整序性能

4 節の実験では、ヘルドアウトデータを用いて式 (1) の重み  $\alpha$  を決定しており、そこで決定した  $\alpha$  は、テストデータに対して最大の語順整序正解率を達成するものであるとは限らない。そこで本節では、評価用データに対して最適な  $\alpha$  を設定したときの語順整序性能を評価する。そのため、評価用データ全 552 文に対して、2 文節単位の語順整序正解率が最大となる  $\alpha$  を決定し、その時の語順整序結果と、4 節の両ベースラインの結果とを比較評価した。なお、評価用データ全 552 文を用いて  $\alpha$  を決定すること以外は、4.2 節と同じ設定により実験した。

実験結果を表 6 に示す。 $\alpha = 0.66$  のときに、本手法の 2 文節単位正解率は最大となった。このときの本手法の語順整序正解率は、2 文節単位と文単位のいずれにおいて

表 6 最適な  $\alpha$  を用いた本手法の語順整序結果

Table 6 Word reordering results by our method using the optimal  $\alpha$ .

	2 文節単位正解率	文単位正解率
本手法 ( $\alpha = 0.66$ )	84.12% (19,542/23,232)	31.16% (172/552)

も、表 3 の両ベースラインのものとは比べて有意に高かった ( $p < 0.05$ )。このことから、重み  $\alpha$  をより適切に定めることができれば、ベースライン 2 に対しても、有意に高い 2 文節単位正解率を達成できる可能性があることがわかる。

なお、本節の実験では、評価用データ全体に対して最適な  $\alpha$  を 1 つ決定したが、表 2 をみると、ヘルドアウトデータごとに最適な  $\alpha$  は変わっていることが分かる。このことを突き詰めると、文ごとに最適な  $\alpha$  を決定することができれば、さらに正解率が向上する可能性があることを示唆している。

### 5.3 入力文の語順の適切さと $\alpha$ の関係

本手法は、入力文の語順の適切さに応じて式 (1) の重み  $\alpha$  を変化させることにより、高精度な語順整序を実現するものである。この手法を提案するにあたり、3.1 節では、式 (2) が先に語順を整えてから係り受け解析を施すことを、式 (3) がその逆の手順を表しているとし、入力文の語順が不適切であるほど、式 (2) を重視する方が、すなわち、式 (1) において小さい値の  $\alpha$  を与える方が、高精度な語順整序が期待できるという仮説を述べた。本節では、この仮説を検証するため、語順の適切さが異なる 4 つのデータを用意し、それぞれに対して最適な  $\alpha$  を求める実験を行った。

実験で用いた 4 つのデータを、語順の適切さが高いと考えられる順に以下に示す。

#### (1) 新聞記事データ

4.1 節で構築した評価用データ 552 文の各文について、元の新聞記事文の語順に戻したデータである。

#### (2) 評価用データ

4.1 節で構築した評価用データ 552 文である。

#### (3) 完全ランダムデータ

4.1 節の評価用データの構築手順 (2) と同じ方法により、(1) の新聞記事データ 552 文の各文に対して、語順を自動的にランダムに変更したデータである。ただし、ランダムに語順を変更する際に、元文と同じ語順になることを許した。

#### (4) 制約付きランダムデータ

語順をランダムに変更する際に、元文と同じ語順は生成しないという制約を加えていること以外は、(3) の完全ランダムデータと同じ方法で作成したデータである。

以上の各データに対して、5.2 節の実験と同じ方法で、2 文節単位の語順整序正解率が最大となる  $\alpha$  を求めた。



表 7 語順の適切さが異なる 4 つのデータに対する実験結果  
Table 7 Experimental results on four kinds of data which have different word order readability.

データ	データの語順 (語順整序なし)		$\alpha$	本手法の語順整序結果	
	2 文節単位正解率	文単位正解率		2 文節単位正解率	文単位正解率
新聞記事データ	100% (23,232/23,232)	100% (552/552)	0.92	86.51% (20,098/23,232)	40.40% (223/552)
評価用データ	76.78% (17,838/23,232)	0% (0/552)	0.66	84.12% (19,542/23,232)	31.16% (172/552)
完全ランダムデータ	74.46% (17,298/23,232)	18.12% (100/552)	0.52	82.58% (19,184/23,232)	32.07% (177/552)
制約付きランダムデータ	60.40% (14,032/23,232)	0% (0/552)	0.41	80.66% (18,671/23,232)	24.46% (135/552)

実験結果を表 7 に示す。2~3 列目は各データの語順をそのまま、語順整序結果とみなして測定した語順整序正解率を、4 列目は、各データにおいて、本手法の 2 文節単位の語順整序正解率が最大となった時の  $\alpha$  の値を、5~6 列目は、その時の本手法の語順整序正解率を示している。入力文の語順の適切さの度合いは、そのままの語順で測定した語順整序正解率で量れると考えると、表 7 から、入力文の語順が不適切なデータであるほど、最適な  $\alpha$  の値が小さくなっていることがわかる。この実験結果は、3 節で述べた仮説が妥当であることを裏付けているといえる。また、本手法が入力文の語順の適切さに応じた処理を実現できていることを示唆しており、高精度な語順整序の達成につながったものと考えられる。

## 6. おわりに

本論文では係り受け解析との統合に基づく日本語文の語順整序手法を提案した。本手法は、先に語順を整えてから係り受け解析を施すことを表した確率モデルと、その逆の手順を表した確率モデルとの加重相乗平均をとることにより、入力文の語順の適切さに応じた尤度計算を実現している。また、本手法は、従来の係り受け解析で利用されてきた CYK 法を拡張したアルゴリズムによって、効率的に解を探索する。京大テキストコーパスを使用した評価実験により、本手法の有効性を確認した。今後は、被験者実験を実施し、出力文の読みやすさを主観的に評価する予定である。

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。

## 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] 内元清貴，村田真樹，馬青，内山将夫，関根聡，井佐原均：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] 横林博，菅沼明，谷口倫一郎：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，情報処理学会論文誌，Vol. 45，No. 5，pp. 1451-1459 (2004)。  
 [4] Filippova, K. and Strube, M.: Generating constituent order in German clauses, In *Proceedings of the 45th An-*

*nual Meeting of the Association of Computational Linguistics (ACL2007)*, pp. 320-327 (2007).  
 [5] Harbusch, K., Kempen, G., van Breugel, C. and Koch, U.: A generation-oriented workbench for performance grammar: Capturing linear order variability in German and Dutch, In *Proceedings of the 4th International Natural Language Generation Conference (INLG2006)*, pp. 9-11 (2006).  
 [6] Kruijff, G. M., Kruijff-Korbayová, I., Bateman, J. and Teich, E.: Linear order as higher-level decision: Information structure in strategic and tactical generation, In *Proceedings of the 8th European Workshop on Natural Language Generation (ENLG2001)*, pp. 74-83 (2001).  
 [7] Ringger, E., Gamon, M., Moore, R. C., Rojas, D., Smets, M. and Corston-Oliver, S.: Linguistically informed statistical models of constituent structure for ordering in sentence realization, In *Proceedings of the 20th International Conference on Computational Linguistics (COLING2004)*, pp. 673-679 (2004).  
 [8] Shaw, J. and Hatzivassiloglou, V.: Ordering among premodifiers, In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL '99)*, pp. 135-143 (1999).  
 [9] Goto, I., Utiyama, M. and Sumita, E.: Post-ordering by parsing for Japanese-English statistical machine translation, In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL2012)*, pp. 311-316 (2012).  
 [10] Elming, J.: Syntactic reordering integrated with phrase-based SMT, In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING2008)*, pp. 209-216 (2008).  
 [11] Ge, N.: A direct syntax-driven reordering model for phrase-based machine translation, In *Proceedings of Human Language Technologies: The 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT2010)*, pp. 849-857 (2010).  
 [12] Christoph, T. and Hermann, N.: Word reordering and a dynamic programming beam search algorithm for statistical machine translation, *Computational Linguistics*, Vol. 29, No. 1, pp. 97-133 (2003).  
 [13] Nizar, H.: Syntactic preprocessing for statistical machine translation, In *Proceedings of the 11th Machine Translation Summit (MT SUMMIT XI)*, pp. 215-222 (2007).  
 [14] 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).  
 [15] 黒橋禎夫，長尾真：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).  
 [16] Zhang, L.: Maximum entropy modeling toolkit for

Python and C++, [http://homepages.inf.ed.ac.uk/s0450736/maxent\\_toolkit.html](http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html) (2008). [Online; accessed 1-March-2008].

- [17] 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. 1834-1842 (2002).