

統計的機械翻訳のための 統語に基づく単純な事前並べ替え手法

星野 翔^{1,†1,a)} 宮尾 祐介^{1,†2} 須藤 克仁^{2,†3} 林 克彦^{2,†4} 永田 昌明²

受付日 2018年6月6日, 採録日 2018年12月4日

概要: 本論文は, 英語と日本語のように語順が大きく異なる言語対における統計的機械翻訳の精度向上のため, 統語に基づく単純な事前並べ替え手法を提案する. まず, 句構造構文解析器を用いて入力文を構文解析および2分木化して, 2分木化句構造木を得る. 次に, 線形サポートベクタマシンを2値分類器として用いて, 2分木の各ノードに反転または非反転の並べ替えラベルを付与する. その後, 構文木に付与された並べ替えラベルに従い, 入力文を並べ替え, 統計的機械翻訳システムを用いて翻訳する. 類似の手法は過去に幾度となく試行されているが, 提案手法は, 2値分類器の学習に必要なオラクル並べ替えラベルおよび分類器の素性テンプレートを同時に改良する. 大規模特許データを用いる英日・日英翻訳実験において, 我々の提案手法は先行研究の事前並べ替え手法の翻訳精度を大幅に改善できることを示す.

キーワード: 統計的機械翻訳, 統語に基づく事前並べ替え

A Simple Syntax-based Preordering Method for Statistical Machine Translation

SHO HOSHINO^{1,†1,a)} YUSUKE MIYAO^{1,†2} KATSUHITO SUDOH^{2,†3} KATSUHIKO HAYASHI^{2,†4}
MASAAKI NAGATA²

Received: June 6, 2018, Accepted: December 4, 2018

Abstract: We propose a simple syntax-based preordering method that improves translation accuracy of distant language pairs, such as English and Japanese, using statistical machine translation. Our method reorders a source-side binary constituent tree by assigning reordering labels, whether the order of child nodes under a binary node should be reversed, using linear support vector machine as a binary classifier. While this idea has been repeatedly implemented in the task of preordering, the way how to obtain oracle reordering labels used for training the classifier remains in a nontrivial open problem. We introduce a procedure to obtain the oracle reordering labels as well as a set of features that improves binary classification accuracy on the task of predicting reordering labels. The tree reordered according to the classified labels is used to yield reordered source sentence, which is fed to a standard statistical machine translation system to generate translation. Experimental results in English-to-Japanese and Japanese-to-English patent translation show that our proposal substantially improves a previously proposed method in terms of translation accuracy.

Keywords: statistical machine translation, syntax-based preordering

¹ 国立情報学研究所
National Institute of Informatics, Chiyoda, Tokyo 101-8430, Japan

² 日本電信電話株式会社 NTT コミュニケーション科学基礎研究所
NTT Communication Science Laboratories, NTT Corporation, Seika, Kyoto 619-0237, Japan

^{†1} 現在, 株式会社みらい翻訳
Presently with Mirai Translate, Inc.

^{†2} 現在, 東京大学
Presently with The University of Tokyo

^{†3} 現在, 奈良先端科学技術大学院大学
Presently with Nara Institute of Science and Technology

^{†4} 現在, 大阪大学
Presently with Osaka University

a) shhshn.shohoshino@gmail.com

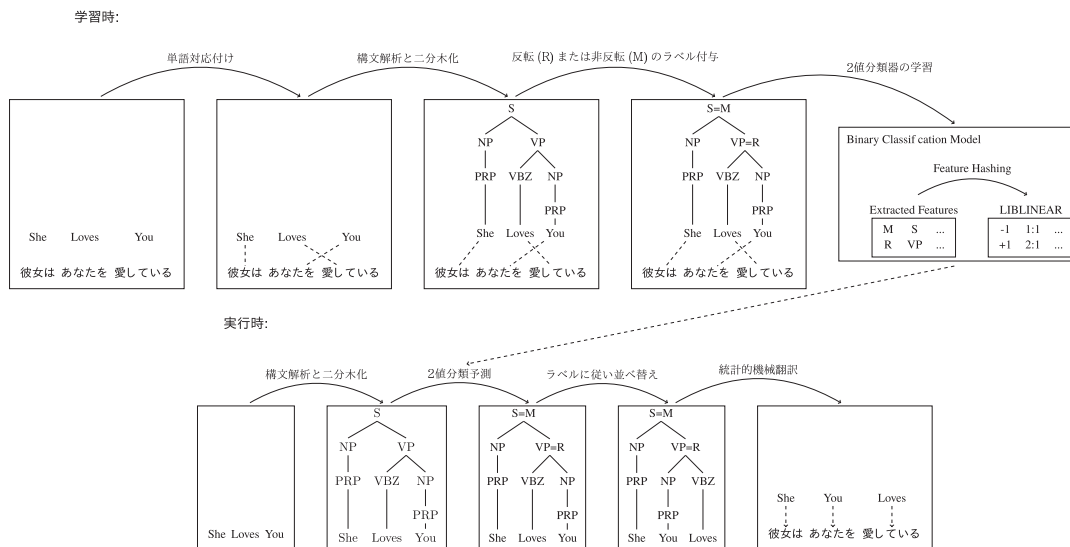


図 1 学習時と実行時における提案手法の概要
 Fig. 1 Overview of the proposed method at training and runtime.

1. はじめに

統計的機械翻訳は、入力文から訳文への語順の並べ替えを必要とするため、適切な語順を持つ訳文の生成が課題となっている。特に英語と日本語のような語順が大きく異なる言語対では、複雑な並べ替えが必要となるため翻訳性能が大幅に劣化する。そのような並べ替え問題に対応するため、統語情報を用いない語彙化並べ替え [39] から、統語に基づく統計的機械翻訳 [43]、事前並べ替え [4]、[42]、階層的な句に基づく統計的機械翻訳 [3]、事後並べ替え [37] など様々に研究されている。しかしながら、語順が大きく異なる言語対に有効でありながら、単純で理解しやすく、即座に使用可能な既存手法は数少ない。

そこで本研究は、統語に基づく単純な事前並べ替え手法を提案する。提案手法は、句構造と 2 値分類器に基づく事前並べ替え手法 [25] を拡張したものである。まず、句構造構文解析器を用いて入力文を構文解析および二分木化して、二分木化句構造木を得る。次に、線形サポートベクターマシンを 2 値分類器として用いて、二分木の各ノードに反転または非反転の並べ替えラベルを付与する。その後、構文木に付与された並べ替えラベルに従い、入力文を並べ替え、統計的機械翻訳システムを用いて翻訳する。

我々の提案する手法は、先行研究の手法 [25] を 2 つの点について改良する。提案手法では、2 値分類器の学習に用いるオラクル並べ替えラベル（正解ラベル）の求め方を、ヒューリスティクスによる手法から、Kendall's τ [21] を用いた理論的な定式化で置き換える。さらに、新しい正解ラベルのために素性エンジニアリングを行い、適切な素性テンプレートを提案する。

大規模特許データを用いる英日・日英翻訳実験において、RIBES [19] と BLEU [31] のスコアを評価尺度として、我々

の提案手法は先行研究の事前並べ替え手法 [25] から翻訳精度を大幅に改善できることを示す。

本研究の提案手法は既発表 [16] であるが、本論文は提案手法の詳細を明らかにする追加実験を行い、論旨を明確化するものである。

2. 統語に基づく事前並べ替え手法

図 1 に提案手法の概要を示す。先行研究の 2 値分類器に基づく事前並べ替え手法 [25] を踏襲しており、2 値分類器の学習方法が違いとなっている。我々の手法は、学習時と実行時の 2 つの異なるパイプラインを持つ。

学習時においては、まず対訳コーパスから単語対応を求める。次に構文解析器を用いて、対訳コーパスのうち入力文側の二分木化句構造木を推定する。その後、句構造木の非終端ノードそれぞれに、反転または非反転の 2 値を持つオラクル並べ替えラベルを、単語対応から算出し割り当てる。最後に、句構造木から抽出した素性とオラクル並べ替えラベルを用いて 2 値分類器を学習する。

実行時においては、まず構文解析器を用いて入力文の二分木化句構造木を推定する。次に学習済みの 2 値分類器を用いて、句構造木の非終端ノードそれぞれに反転または非反転の並べ替えラベルを推定する。その後、推定された並べ替えラベルに基づいて入力文を並べ替える。最後に並べ替えた入力文を統計的機械翻訳システムで翻訳する。

2.1 句構造と 2 値分類器に基づく事前並べ替え

ここで、先行研究である句構造と 2 値分類器に基づく事前並べ替え手法 [25] の概略を、異なる定式化を用いて示す*1。

*1 なお先行研究では、2 値分類器を 6 クラスの多クラス分類に置き換えた、二分木から三分木への拡張方法も議論されているが、提案手法は二分木と 2 値分類器の組合せに準じている。

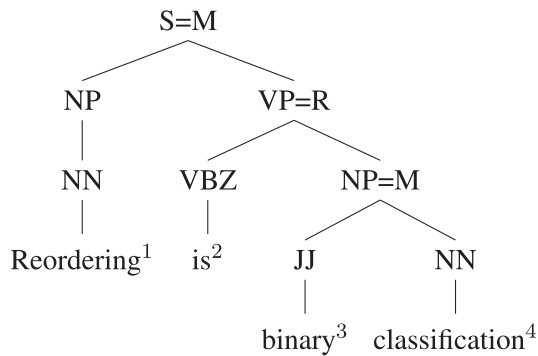


図 2 句構造木の各ノードに 2 値の並べ替えラベルが付与される。ラベル R は反転を、ラベル M は非反転を意味する

Fig. 2 Each node is assigned a binary reordering label. The label R indicates reversed and the label M indicates monotone.

とある n 単語の文と、その文に対応する 2 分木化句構造木が存在するとき、各ノードを

$$v(i, p, j)$$

と表記する。ここで、整数 i, p, j は $1 \leq i \leq p < p+1 \leq j \leq n$ を満たす。つまり、このノードは i 番目の語から j 番目の単語へのスパン (i, j) を持ち、またそのスパン全体は、 i 番目の単語から p 番目の単語への左スパン (i, p) と $(p+1)$ 番目の単語から j 番目の単語への右スパン $(p+1, j)$ に分割できることを表している。

次に、各ノードを反転・非反転のいずれかの操作で並べ替えることを考える*2。並べ替えを行う確率を

$$P(x | \theta(v(i, p, j)))$$

と表記する。ここで、並べ替えラベル x は $x \in \{R, M\}$ を満たし、ラベル R は反転（子ノードの順番を逆転させる操作）、ラベル M は非反転（子ノードの順番を逆転させない操作）を表す。関数 θ は、ノード v を入力とする素性関数を表す。素性関数で用いられる素性テンプレートの詳細については 2.4 節で説明する。

図 2 に例を示す。ここでは、4 単語 ($n = 4$) の文が S, VP, NP の 3 ノードを持つ句構造木に対応している。このうち、NP ノード $v(3, 3, 4)$ は $binary^3$ に対応する左スパン $(3, 3)$ と $classification^4$ に対応する右スパン $(4, 4)$ を持つ。NP ノードの並べ替えラベルは $P(x | \theta(v(3, 3, 4)))$ として計算でき、この例では $x = M$ と分類されている。VP ノード $v(2, 2, 4)$ と根ノードである S ノード $v(1, 1, 4)$ についても同様の方法で並べ替えラベルが計算される。

全ノードに並べ替えラベルを計算した後、ラベル R が付与されたノードのみを反転させる。たとえば、図 2 の

*2 3.1 節で後述するように、実際には線形サポートベクタマシンを 2 値分類器として用いる。しかし、提案手法は特定の分類器に限られるものではなく、たとえば先行研究 [25] では、最大エントロピー分類器が用いられている。

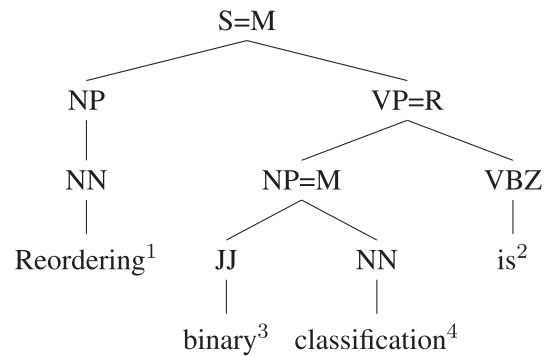


図 3 並べ替えラベルに従って並べ替えられた句構造木は、*Reordering binary classification is* という文を表す

Fig. 3 The tree reordered according to the assigned labels, which resembles the reordered sentence: *Reordering binary classification is*.

句構造木を並べ替えて、図 3 の句構造木を得る。この新たな句構造木は、入力文を並べ替えた *Reordering binary classification is* という文に対応している。この文は英語を日本語と語順が近くなるよう並べ替えた文と見なすことができる。その後、並べ替えた文を統計的機械翻訳システムを用いて翻訳する。

2.2 オラクル並べ替えラベルの算出手順

これまで概略を示した、提案手法が踏襲している先行研究の手法 [25] では、各ノードに付与された並べ替えラベルの集合が 1 文全体の並べ替え結果と対応しているため、並べ替えラベルを予測する 2 値分類器の性能が事前並べ替えの成否を左右する。具体的には、1 文全体の並べ替え結果が最適となるよう、適切なオラクル並べ替えラベル（正解ラベル）を与えて 2 値分類器を学習させる必要がある。しかし、そのようなオラクル並べ替えラベルのデータは存在せず、人手での付与は作成コストと一貫性の問題から困難である。

そこで我々は、句構造木と単語対応*3を入力として、系統的にオラクル並べ替えラベルを付与するための手順を提案する。この手順どおりに付与された並べ替えラベルは、2 値分類器の学習データに用いられるほか、評価データとして用いることもできる。

我々は、あるノード $v(i, p, j)$ が与えられたとき、スパン (i, j) についての Kendall's τ [21] が最大となるラベルをオラクル並べ替えラベルと定義する。事前並べ替えは入力文を並べ替えて訳文の語順に近づけるタスクであり、その近さを順位相関係数として表すことができることから、Kendall's τ は、統計的機械翻訳の評価尺度 [1], [18], [38] お

*3 3.2 節で、この単語対応の正解率が並べ替えの正解率にどの程度影響しているか検証する。一般に、単語対応の正解率と並べ替えの正解率は相互依存関係にある。たとえば、事前並べ替えを単語対応の前処理に適用することで、単語対応の正解率の改善が可能だと報告されている [7]。

よび並べ替えの目的関数 [28], [44] として広く用いられている。

なお既存研究では Kendall's τ に代わる評価尺度として fuzzy reordering score [38] や fuzzy reordering score を改良した chunk fragmentation [28] が用いられており、それぞれ有効性が示されている。しかし Kendall's τ と異なり、いずれもスコア値が左スパンと右スパンの内容 (特に両端) に依存するため、後述する反転・非反転の単純な操作ではオラクル並べ替えラベルを算出できない。そのため、本研究は Kendall's τ を用いている。他の評価尺度を提案手法に取り入れることは将来の課題としたい。

数列 $\mathbf{x} = x_1, \dots, x_n$ が与えられたとき、Kendall's τ は \mathbf{x} と昇順で整列後の \mathbf{x} の類似度 τ を

$$\tau(\mathbf{x}) = \frac{4c(\mathbf{x})}{n(n-1)} - 1$$

と定義する。ここで、順方向の組の個数を表す $c(\mathbf{x})$ を

$$c(\mathbf{x}) = \sum_{i,j \in [1,n], i < j} \delta(x_i < x_j)$$

と定義する。 $\delta(x_i < x_j)$ は、順方向 $x_i < x_j$ のとき 1 を、それ以外は 0 を返す Kronecker のデルタ関数を表す。 $\tau(\mathbf{x}) = 1$ は数列 \mathbf{x} が昇順で整列済みであることを表し、 $\tau(\mathbf{x}) = -1$ は数列 \mathbf{x} が降順で整列済みであることを表す。

事前並べ替えでは、数列 \mathbf{x} の代わりに単語対応 \mathbf{a} を

$$\mathbf{a} = a_1, \dots, a_n$$

と定義して用いる。ここで、 $a_x = y$ は入力文の x 番目の単語が訳文の y 番目の単語に対応することを表す*4。さらに、2分木におけるスパン (i, j) に対応する部分的な単語対応を

$$\mathbf{a}(i, j) = \{a_k \mid i \leq k \leq j\}$$

と定義する。

ノード $v(i, p, j)$ に対応する並べ替えラベルを算出するため、スパン (i, j) についての局所スコアを

$$\begin{aligned} s(v(i, p, j)) \\ = \tau(\mathbf{a}(i, p) \cdot \mathbf{a}(p+1, j)) - \tau(\mathbf{a}(p+1, j) \cdot \mathbf{a}(i, p)) \end{aligned}$$

と定義する。ここで、記号 \cdot は単語対応の連結を表す。このスコアは、子ノードの単語対応を左から右に連結した場合の順方向の組の個数 $\tau(\mathbf{a}(i, p) \cdot \mathbf{a}(p+1, j))$ と、子ノードの単語対応を右から左に連結した場合の順方向の組の個数 $\tau(\mathbf{a}(p+1, j) \cdot \mathbf{a}(i, p))$ を比較している。

この局所スコアが、 $s(v(i, p, j)) < 0$ を満たす場合は R の

*4 実際には、grow-diag-final-and ヒューリスティクスで得られる 1 対多の単語対応について、 y の中央値をとることで、1 対 1 の単語対応にあらかじめ変換する。

並べ替えラベルを、 $s(v(i, p, j)) > 0$ を満たす場合は M の並べ替えラベルを付与する。いずれにも該当しない $s = 0$ の場合、並べ替えラベルの違いが並べ替え結果に影響を与えない本質的に曖昧な事例であると考えられるため、2値分類器の学習データに用いず事例を除外する。たとえば、あるノードについて子ノードがまったく同じ単語対応 $a_x = y$ を共有している場合、 $s = 0$ に該当し、学習データから除外される*5。

ここまで、局所的なスコアを用いてオラクル並べ替えラベルを算出する手順を示したが、局所的な並べ替えラベルの付与が 1 文全体について最適であることを、次に示す。

2.3 オラクル並べ替えラベルの算出根拠

我々は、文レベルの Kendall's τ が再帰的に計算可能であることを示した既存研究 [28], [44] を援用して、提案手法が局所的に算出するオラクル並べ替えラベルが 1 文全体について最適であることを示す。

ここで、 $c(\mathbf{a}(i, j))$ は

$$\begin{aligned} c(\mathbf{a}(i, j)) = & c(\mathbf{a}(i, p)) + c(\mathbf{a}(p+1, j)) \\ & + \sum_{k \in [i,p], l \in [p+1,j]} \delta(a_k < a_l) \end{aligned}$$

と分解可能であり、3つの項は互いに独立である。つまり、 $\mathbf{a}(i, p)$ の並べ替えは、第 1 項 $c(\mathbf{a}(i, p))$ にしか影響せず、他の 2 項には影響を与えない。第 2 項および第 3 項についても同様である。

そこで、各項を独立に最大化すると、 $c(\mathbf{a}(i, j))$ は最大となる。第 1 項および第 2 項は再帰的に計算できるため、第 3 項の最大化を考える。この第 3 項は、提案手法における局所スコア $s(v(i, p, j))$ と対応している。そのため、局所スコアを最大化すると、 $c(\mathbf{a}(i, j))$ が最大となる。 $c(\mathbf{x})$ は $\tau(\mathbf{x})$ に比例するため、 $c(\mathbf{a}(i, j))$ を最大化すると、 $\tau(\mathbf{x})$ 、つまり Kendall's τ が最大となる。よって、局所的なスコアを最大化する ($s(v(i, p, j)) < 0$ が反転、 $s(v(i, p, j)) > 0$ が非反転) オラクル並べ替えラベルが、1 文全体について最適であることが示される。

なお、オラクル並べ替えラベルは $\tau(\mathbf{a}) \geq 0$ を保証するが、句構造の制約があるため $\tau(\mathbf{a}) = 1$ は保証されない。図 4 に、どのノードを反転させても $\tau(\mathbf{a}) = 1$ が満たされない例を示す。

2.4 素性

表 1 に提案手法がノード $v(i, p, j)$ における素性関数 θ で用いる素性テンプレートを示す。素性テンプレートには、

*5 これらの並べ替えラベルの個数は、データに依存する。一例として、3.2 節で使用する 8,000 文の学習データからは以下の値が得られる。Giza データでは、 R ラベルが 69,178 個、 M ラベルが 102,916 個で、残りの 43,896 個が除外される。Nile データでは、 R ラベルが 90,724 個、 M ラベルが 70,583 個で、残りの 54,683 個が除外される。

スパン素性とツリー素性の2種類がある. このうちスパン素性は, 左スパン (i, p) と右スパン $(p+1, j)$ の違い, たとえば主辞が左右のどちらに属しているかをとらえる役割を持つ. スパン素性では, x 番目から y 番目の単語を, インデックス $x:y$ を用いて表記する. 特に, 非終端記号は $t_{x:y}$,

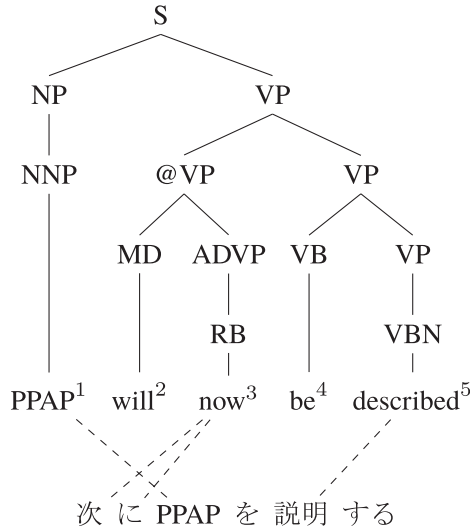


図 4 句構造の制約で $\tau(\mathbf{a}) < 1$ となっている稀な例. 英語の句構造木と日本語文の単語対応を点線で示す

Fig. 4 Rare case of $\tau(\mathbf{a}) < 1$ under tree structure: Dashed lines represent word alignment between the English parse tree and its translation in Japanese.

表 1 ノード $v(i, p, j)$ における素性テンプレート. 整数 d, l, r は $d \geq 0 \wedge l = \max(i, p-d) \wedge r = \min(p+1+d, j)$ を満たす

Table 1 Feature templates for the node $v(i, p, j)$, where integers d, l , and r satisfy $d \geq 0 \wedge l = \max(i, p-d) \wedge r = \min(p+1+d, j)$.

スパン素性	ツリー素性
$t_{i:p}, t_{p+1:j}, w_{i:p}, w_{p+1:j}$,	$\sigma(v(i, p, j))$,
$t_{i:p} \circ t_{p+1:j}, w_{i:p} \circ w_{p+1:j}$,	$\sigma_r(v(i, p, j))$,
$t_{i:p} \circ t_{p+1:j} \circ w_{i:p} \circ w_{p+1:j}$,	$\sigma_t(v(i, p, j))$,
$t_{l:p}, t_{p+1:r}, w_{l:p}, w_{p+1:r}$,	$\sigma_w(v(i, p, j))$
$t_{l:p} \circ t_{p+1:r}, w_{l:p} \circ w_{p+1:r}$,	
$t_{l:p} \circ t_{p+1:r} \circ w_{l:p} \circ w_{p+1:r}$,	

表 2 図 2 中の VP ノード $v(2, 2, 4)$ における素性のインスタンスを示す

Table 2 Feature instance examples for the VP node $v(2, 2, 4)$ in Fig. 2.

素性テンプレート	インスタンス
$t_{2:2}$	VBZ
$t_{3:4}$	JJ_NN
$t_{3:3}$	JJ
$w_{2:2}$	is
$w_{3:4}$	binary_classification
$w_{3:3}$	binary
$\sigma(v(2, 2, 4))$	(VP(VBZis)(NP(JJbinary)(NNclassification)))
$\sigma_r(v(2, 2, 4))$	0VP 1VBZ 1NP 2JJ 2NN 0VP_VBZ 0VP_NP 1NP_JJ 1NP_NN
$\sigma_t(v(2, 2, 4))$	(VP(VBZ)(NP(JJ)(NN)))
$\sigma_w(v(2, 2, 4))$	((is)((binary)(classification)))

単語表層は $w_{x:y}$, さらに素性の組合せは記号 \circ を用いて表記する. さらに, あらゆる長さのスパンを内包することを意図して, 左スパンと右スパンそれぞれの部分集合を用いる. これら部分集合を

$$d \geq 0 \wedge l = \max(i, p-d) \wedge r = \min(p+1+d, j)$$

を満たす整数 d, l, r を用いて, 左部分スパン (l, p) と右部分スパン $(p+1, r)$ と表記する. 整数 d には, 重複しない部分スパンをすべて生成するため, 0 以上かつ部分スパンの最大長以下の値を順に与える. 左部分スパンは左スパンの部分集合かつ右端インデックスが p に固定されている点特徴的で, 同様に, 右部分スパンも右スパンの部分集合かつ左端インデックスが $p+1$ に固定されている点特徴的である. このため, 左右の部分スパンは, 単純な n-gram ではない. これら左右部分スパンは, 左右スパンと同じく, 左スパンと右スパンの違いをとらえる役割を持つ.

ツリー素性は, S 式を用いて木構造を文字列に変形する関数と, 深さを付与した非終端記号の組合せで, 各ノードを木構造として表現する. ツリー素性では, ノード $v(i, p, j)$ の S 式を $\sigma(v(i, p, j))$ と表記する. 特に, 非終端記号のみの S 式を $\sigma_t(v(i, p, j))$, 単語表層のみの S 式を $\sigma_w(v(i, p, j))$ と表記する. 加えて, ノード $v(i, p, j)$ 中の非終端記号に着目して, 非終端記号の親子の組合せ (子ノードが存在しない場合は親ノードのみ) を, 根ノードからの深さを表す整数と組み合わせて列挙し, $\sigma_r(v(i, p, j))$ と表記する. これらの素性のうち, $\sigma(v(i, p, j))$, $\sigma_t(v(i, p, j))$, $\sigma_w(v(i, p, j))$ の 3 つは単一の素性を表すが, $\sigma_r(v(i, p, j))$ については各組合せが 1 つの素性を表す.

例として, 表 2 に図 2 中の VP ノード $v(2, 2, 4)$ についての素性インスタンスを示す. この VP ノードは, 左スパン (is^2) と右スパン $(binary^3 classification^4)$ を持つ. 各テンプレートは文字列素性を表しており, たとえば, テンプレート $t_{2:2}$ は非終端記号の文字列 VBZ の素性として発火する. 実際には, これらの素性単体に加えて, 各素性を組み合わせた文字列を用いる. たとえば, 組合せ $t_{2:2} \circ t_{3:4}$

表 3 データ分割の概略を示す。人手で単語対応付けした 8,000 文と単語対応付けされていない 100 万文の両方を 2 値分類器の学習に用いる

Table 3 Data Splitting: Both the 8,000 manually annotated sentences and one million unannotated sentences are used to train binary classifier.

元データ	学習データ NTCIR-7/8			開発データ NTCIR-8		評価データ NTCIR-9/10		
ラベル名	training			未使用	tuning	dev	test9	test10
	人手で 単語対応付け	単語対応付けなし 無作為抽出	1,908,615		人手で 単語対応付け			
行数	8,000	1,000,000		1,000	1,000			
合計		2,916,615		269,669	2,000		2,000	2,300

は文字列 VBZ と文字列 JJ_NN を組み合わせた 1 つの文字列の素性として発火する。

事前並べ替えの素性としては、ブラケット化トランスダクション文法に基づく事前並べ替え手法 [28] が、各ノードにおけるスパン長、左右子ノードのスパン長の関係、単語表層、非終端記号、Giza を用いて得られる単語クラス、さらにフレーズテーブルを辞書として用いている。前述の手法をさらに発展させた研究 [26] は、これらの素性の組合せを大量に追加しており、単語クラスの獲得に Brown クラスタリング [2] を用いている。また提案手法と同じく 2 値分類器を用いる事前並べ替え手法 [25] では、主辞、非終端記号、各ノードのスパン周辺の単語を用いている。

提案手法の素性テンプレートは、構文解析の文献を参考に、これら既存の素性のうち単語表層と非終端記号の組合せを発展させて開発したものである。スパン素性は、スパン境界周辺の単語素性 [15] から着想しているが、左右スパンの境界周辺の単語をスライドさせて用いる点が異なる。またツリー素性は、非局所素性 [17] から着想しているが、構文木全体を用いる点が異なる。これらの改良を加えた結果、並べ替えの正解率を評価尺度とする予備実験において、着想元の構文解析の素性からの正解率の向上が見られた。

3. 実験

提案手法を評価するため、特に並べ替えの難易度が高いとされている英語と日本語の言語対を用いて、以下の順番で実験を行う。

- まず、教師ありおよび教師なしの単語対応器を比較することにより、並べ替えラベルの割当てに用いる 2 値分類器の学習に必要な単語対応データの質と量を調査する (3.2 節)
- 次に、素性テンプレートの除去試験により、提案手法が用いる素性テンプレートの貢献度を確かめる (3.3 節)
- その後、多分木の代わりに 2 分木を用いることによる提案手法の性能への影響度合いを分析する (3.4 節)
- 最後に、評価データを用いて提案手法の翻訳性能を調査し (3.5 節)、既存手法と比較する (3.6 節)

3.1 実験設定

実験では、NTCIR 特許機械翻訳テストコレクションに含まれる英語と日本語の対訳データを用いる。この対訳データは合計で 300 万文対を超えるが、NTCIR-9 [13] および NTCIR-10 [12] のワークショップで用いられた実験設定に従って、NTCIR-7 と NTCIR-8 の学習データ、NTCIR-8 の開発データ、NTCIR-9 と NTCIR-10 の評価データを使用する。表 3 に我々の実験で使用するデータの切り出し方法を示す。まず学習データのうち、構文解析に失敗した 269,669 文は除く。次に、学習データから 8,000 文と 100 万文をそれぞれ重ならないよう無作為抽出する。開発データ 2,000 文は、前半の 1,000 文を機械翻訳システムのチューニングのために用い、後半の 1,000 文 (dev) を提案手法の開発に用いる。この開発データの後半 1,000 文 (dev) および学習データから無作為抽出した 8,000 文には人手で単語対応を付与する。

機械翻訳システムには、標準的なフレーズベースの統計的機械翻訳システムである Moses 3 [23] を用いる。以下に詳細を列挙する。言語モデルは SRILM 1.7.0 [36] を用いて 6-gram の設定で学習する。単語対応は MGIZA [10] を用いて自動推定し、grow-diag-final-and ヒューリスティクスで対称化する。機械翻訳システムは、語彙化並べ替え [39] を用いる設定で、歪み制限のハイパーパラメータを人手で与えて、最小誤り率学習手法 (MERT) [29] を用いてチューニングする。デコード時の探索アルゴリズムにはスタックをデフォルト値のサイズ 100 で使用し、その他の設定も最大フレーズ長が 20 などデフォルト値に準じる。事前並べ替え手法は、統計的機械翻訳システムに与えるデータの前処理として適用する。

機械翻訳システムの評価には、RIBES [18] と BLEU [31] をそれぞれデフォルト設定で用いる*6。RIBES は大局的な語順の正しさを、BLEU は局所的な語順の正しさを評価することをそれぞれ意図している。この 2 つの評価尺度は、人間が作成した参照訳と機械翻訳システムの出力の類似度を、[0, 100]% の値域でスコア付けし、高スコアであるほど

*6 RIBES と BLEU のスコア付けと統計的検定には Travatar [27] を用いた。

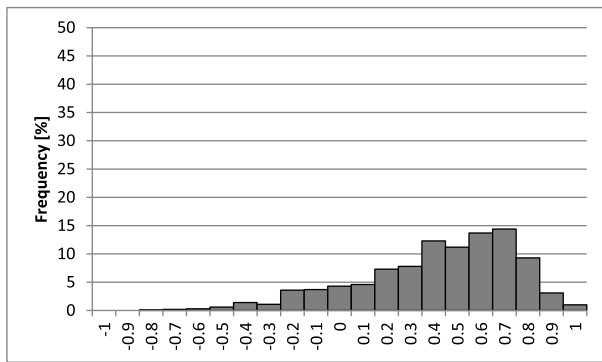


図 5 日英翻訳の開発データ (*dev*). τ 平均値は 0.4172

Fig. 5 Japanese-to-English development data (*dev*): average $\tau = 0.4172$.

システム訳が人間訳に類似していることを表す。

事前並べ替えのためのデータ前処理として、NTCIR の対訳データを Berkeley parser 1.7 [32], [33] を用いて 2 分木化句構造木として構文解析する。特に日本語については、Haruniwa [9] の 2 分木モデル、Comainu 0.7.0 [45], MeCab 0.996 [24], UniDic 2.1.2 [5] の各ツールを用いることで、日本語版の Berkeley parser を構築して、英語と同様に構文解析する。Berkeley parser は英語における構文解析精度が F1 スコアで 90.2, Haruniwa は日本語における構文解析精度が F1 スコアで 80.29 であると報告されており、後者の構文解析精度は相対的に低いことが知られている。

2 値分類器の構築には、LIBLINEAR 1.96 [8] の線形サポートベクタマシンを用いる。その際、パラメータ C は 1.0 に固定し、バイアス項は用いていない。さらに、2.4 節で導入した文字列素性は、実際には素性ハッシュ化 [35] を用いて、30 ビットの符号なし整数値に変換して用いる。素性ハッシュ化により、2 値分類器の学習に必要なメモリ使用量を削減し、また学習速度を向上させることができる。

3.2 単語対応器の比較

提案手法が用いる 2 値分類器の学習のために、より良い学習データ (単語対応データ) を生成することを目的として、2 種類の単語対応器を調査する。第 1 の調査では、教師なし単語対応器 Giza++ [30] を学習データ (前述の約 300 万文対) に適用し、得られた単語対応の出力を *Giza* データと名付ける。第 2 の調査では、教師あり単語対応器 Nile [34] を学習データから無作為抽出して人手で単語対応付けした 8,000 文対で学習する。その後、学習済みの Nile を学習データに適用し、得られた単語対応の出力を *Nile* データと名付ける。なお、Giza++ と Nile の 2 つの単語対応器の精度を人手で単語対応付けした開発データ (*dev*) を用いて評価すると、Giza++ は F1 スコアで 50.1, Nile は F1 スコアで 86.9 となった。

次に、*Giza* データと *Nile* データの一部を用いて 2 値分類器を学習し、それぞれを *Giza* 分類器および *Nile* 分類器

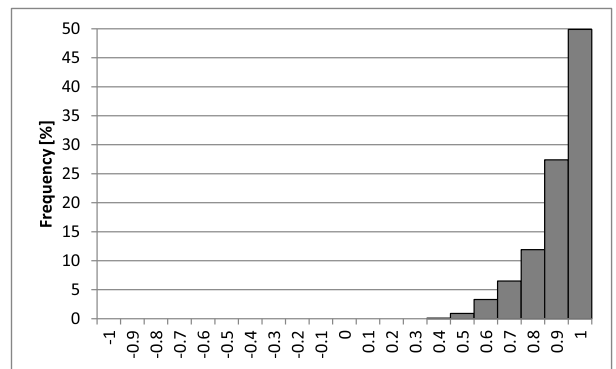


図 6 オラクル並べ替えラベルを算出する提案手法を適用した開発データ. τ 平均値は 0.9091

Fig. 6 Development data after applying the proposed procedure to obtain oracle reordering labels: average $\tau = 0.9091$.

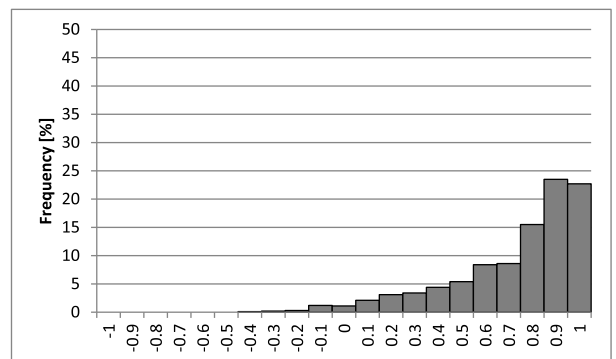


図 7 Giza 分類器を適用した開発データ. τ 平均値は 0.7320

Fig. 7 Development data reordered by the Giza classifier: average $\tau = 0.7320$.

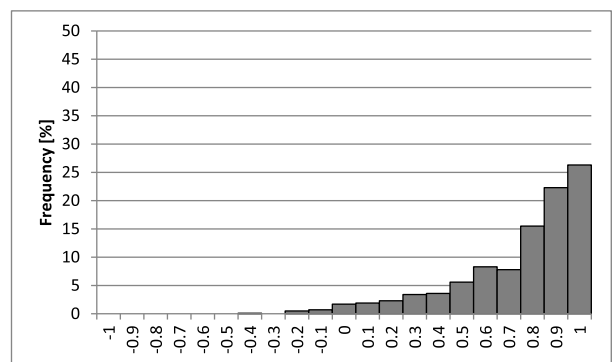


図 8 Nile 分類器を適用した開発データ. τ 平均値は 0.7478

Fig. 8 Development data reordered by the Nile classifier: average $\tau = 0.7478$.

と名付ける。具体的には、学習データ 300 万文対のうち、人手で単語対応付けした 8,000 文と、その他の 100 万文の、合計 100 万 8,000 文を上限として 2 値分類器の学習データに用いる。メモリ使用量と学習時間を抑えるために、全データは用いない。学習後の *Giza* 分類器および *Nile* 分類器の性能を調査するため、人手で単語対応付けした開発データ (*dev*) をそれぞれの分類器で並べ替え、Kendall's

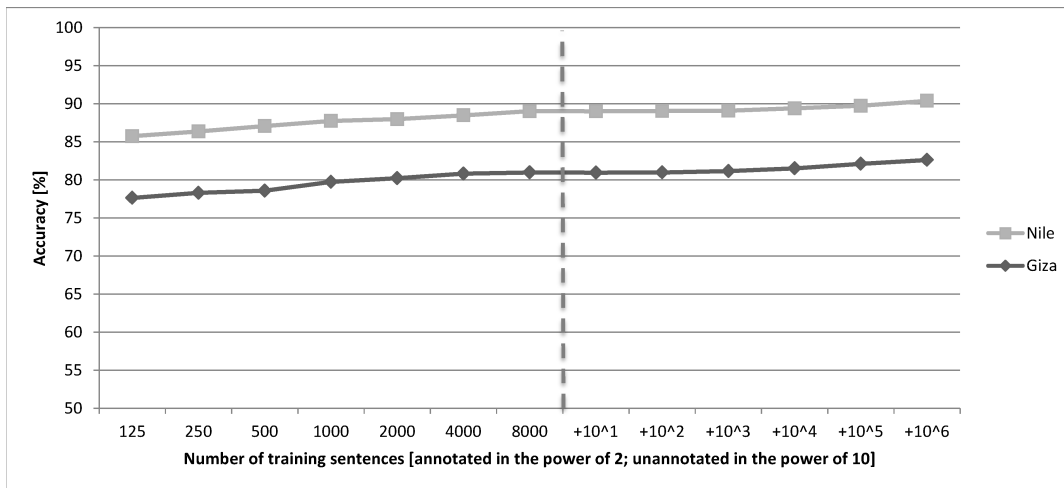


図 9 開発データ (dev) における Giza 分類器と Nile 分類器それぞれの 2 値分類正解率の学習曲線

Fig. 9 Binary classification accuracy of the learning curves of the Giza and Nile classifiers for the development data (dev).

τ の分布 [18] と平均値を用いて内的評価する。Kendall's τ の分布では、横軸は文レベルの Kendall's τ の値、縦軸は文の数の総数に対する割合を示すことで、並べ替え後の語順と正解語順の類似度を可視化する。

図 5 に Kendall's τ 分布で可視化した事前並べ替え前の開発データを示す。Kendall's τ の平均値 0.4 は、語順の相関が弱いことを意味し、英語と日本語の言語対での並べ替えタスクの難しさを表している。他方で、図 6 には提案手法におけるオラクル並べ替えラベルを用いた事前並べ替え後の開発データを示す。Kendall's τ の平均値 0.9 は、語順の相関が非常に強いことを意味する。図 5 と図 6 の間の違いは、オラクル並べ替えラベルによって語順の違いの大部分が解消されたことを明確に示す。

図 7 と図 8 に、Kendall's τ 分布で可視化した、それぞれ Giza 分類器と Nile 分類器で事前並べ替えした開発データを示す。両者の比較により、Nile 分類器は Giza 分類器より多くの整列済みの並べ替え結果 ($\tau(x) = 1$) が得られていることが分かる。

図 9 に Giza 分類器と Nile 分類器を学習した際の学習曲線を示す。縦軸は開発データにおける正解率、横軸は学習データ量を文の数で示している。学習データには、人手で単語対応付けした 8,000 文対と単語対応付けられていない 100 万文対の両方を使用しており、単語対応付けした 8,000 文対は 2 の累乗、単語対応付けられていない 100 万文対は 10 の累乗の割合でデータ量を変化させている。この学習曲線から、Nile 分類器が Giza 分類器より優れていることが分かる。

表 4 に提案手法を開発データ (dev) における 2 値分類の正解率で内的評価した結果および翻訳精度で外的評価した結果を示す。ここでは、Giza 分類器と Nile 分類器の違いと、歪み制限 (DL と記載) 3 種類の違いに着目する。歪

表 4 開発データ (dev) の日英翻訳における Giza 分類器と Nile 分類器の結果。太字は bootstrap resampling [22] において最高スコアと統計的有意性 ($p < 0.01$) がないことを表す

Table 4 Results of the Giza and Nile classifiers for the Japanese-to-English translation of the development data (dev). Bold text denotes no significance ($p < 0.01$) from the highest score obtained within bootstrap resampling [22].

手法	DL	正解率	dev	
			RIBES	BLEU
Giza 分類器	0	82.60	76.72	31.38
Giza 分類器	10	82.60	76.71	31.64
Giza 分類器	20	82.60	76.22	31.11
Nile 分類器	0	90.36	76.80	31.45
Nile 分類器	10	90.36	76.89	31.81
Nile 分類器	20	90.36	76.32	31.77

み制限は、0 以上 20 以下の値が使われることが多いため、0, 10, 20 の 3 候補を探索した。この結果から、Nile 分類器と歪み制限 10 の組合せが最善であると分かったため、以降の実験では同様の設定を用いる。

3.3 素性テンプレートの除去試験

表 5 に、先行研究 [25] の素性テンプレート、提案手法の素性テンプレート、および提案手法の素性テンプレートの一部を除去した場合のそれぞれを提案手法に適用し、開発データ (dev) における 2 値分類の正解率と翻訳精度で比較した結果を示す。この比較から、提案手法の素性テンプレートすべてを用いる場合が最善であると分かった。また、スパン素性がツリー素性より有効であることも確認できた。その原因としては、非終端記号の素性が句構造の統語ラベルを含んでいるため、単語表層と非終端記号の組合せのみで十分強力なシグナルとなっていることが考えられ

表 5 開発データ (*dev*) の日英翻訳における除去試験結果. 太字は bootstrap resampling [22] において最高スコアと統計的有意性 ($p < 0.01$) がないことを表す

Table 5 Ablation tests in the Japanese-to-English translation of the development data (*dev*). Bold text denotes no significance ($p < 0.01$) from the highest score obtained within bootstrap resampling [22].

素性テンプレート	DL	正解率	dev	
			RIBES	BLEU
全素性	10	90.36	76.89	31.81
スパン素性以外	10	86.97	73.02	30.26
$\sigma(v(i, p, j))$ 以外	10	90.18	76.83	31.71
$\sigma_r(v(i, p, j))$ 以外	10	90.29	76.28	31.67
$\sigma_t(v(i, p, j))$ 以外	10	90.33	76.73	31.91
$\sigma_w(v(i, p, j))$ 以外	10	90.31	76.94	31.62
先行研究 [25] の素性	10	84.38	66.59	27.39

表 6 開発データ (*dev*) の日英翻訳における予測した並べ替えラベルとオラクル並べ替えラベルの結果. 太字は bootstrap resampling [22] において最高スコアと統計的有意性 ($p < 0.01$) がないことを表す

Table 6 Results of predicted and oracle classifiers for the Japanese-to-English translation of the development data (*dev*). Bold text denotes no significance ($p < 0.01$) from the highest score obtained within bootstrap resampling [22].

手法	DL	正解率	dev	
			RIBES	BLEU
Nile 分類器	10	90.36	76.89	31.81
2 分木制約ありのオラクル	10	100	82.36	33.58
2 分木制約なしのオラクル	10		84.65	36.62

る. さらに, 提案手法の素性テンプレートの重要性が, 先行研究の素性テンプレートとの違いによって実験的に示されている.

3.4 予測した並べ替えラベルとオラクル並べ替えラベルの比較

句構造木と 2 値分類器に基づく提案手法の妥当性を評価するため, 2 値分類誤りと 2 分木制約のそれぞれが提案手法に与えている影響を開発データ (*dev*) における翻訳精度で示す. 2 値分類誤りについては, 予測した並べ替えラベル (Nile) と 2.2 節で述べた提案手法を適用することで得られるオラクル並べ替えラベル (2 分木制約ありのオラクル) を比較することで, 現在の予測精度に基づく提案手法と, その上限との差を明らかにする. 2 分木制約については, 前述のオラクル並べ替えラベル (2 分木制約ありのオラクル) と, 単語対応を安定的ソートで並べ替え [40], その単語対応に基づき文を直接並べ替えた結果 (2 分木制約なしのオラクル) を比較する.

表 6 に, 予測した並べ替えラベル, オラクル並べ替えラ

表 7 英日翻訳における最終結果. 太字は bootstrap resampling [22] において最高スコアと統計的有意性 ($p < 0.01$) がないことを表す

Table 7 Final testing results for the English-to-Japanese translation. Bold text denotes no significance ($p < 0.01$) from the highest score obtained within bootstrap resampling [22].

手法	DL	test9			test10	
		RIBES	BLEU	RIBES	BLEU	
事前並べ替えを用いないベースライン						
Moses	0	64.91	25.83	65.16	27.64	
Moses	10	69.40	31.34	69.33	32.57	
Moses	20	70.74	32.05	70.82	32.91	
提案手法						
Giza 分類器	0	78.68	34.62	78.75	35.61	
Giza 分類器	10	78.72	34.82	78.88	36.28	
Nile 分類器	0	79.05	35.03	79.11	36.25	
Nile 分類器	10	79.04	35.42	79.29	36.59	

表 8 日英翻訳における最終結果. 太字は bootstrap resampling [22] において最高スコアと統計的有意性 ($p < 0.01$) がないことを表す

Table 8 Final testing results for the Japanese-to-English translation. Bold text denotes no significance ($p < 0.01$) from the highest score obtained within bootstrap resampling [22].

手法	DL	test9			test10	
		RIBES	BLEU	RIBES	BLEU	
事前並べ替えを用いないベースライン						
Moses	0	66.95	26.36	67.50	27.17	
Moses	10	68.95	29.41	69.64	30.20	
Moses	20	69.88	30.12	70.22	30.51	
提案手法						
Giza 分類器	0	77.49	33.08	77.49	33.65	
Giza 分類器	10	77.44	33.28	77.42	33.77	
Nile 分類器	0	77.74	32.97	77.89	33.91	
Nile 分類器	10	77.97	33.55	78.07	34.13	

ベル, 文を直接並べ替えた結果のそれぞれについて, 開発データ (*dev*) の翻訳精度を示す. それぞれに違いがあるが, 予測した並べ替えラベルとオラクル並べ替えラベルの間の違いは, 2 値分類誤りに起因する翻訳精度の劣化が有意に存在することを表している. また同時に, 2 分木制約に起因する翻訳精度の劣化も有意に存在することを表している. これらの結果から, 2 値分類誤りまたは 2 分木制約に起因する翻訳精度の劣化を防ぐことで提案手法が相当改善できることが示唆される. 具体的な改善方法は将来の課題としたい.

3.5 評価データにおける実験結果

表 7 と表 8 に, 英日翻訳と日英翻訳での提案手法の性能を, 評価データにおける翻訳精度でそれぞれ示す. この

表 9 英日翻訳における提案手法と既存手法の結果. 太字は bootstrap resampling [22] において最高スコアと統計的有意性 ($p < 0.01$) がないことを表す

Table 9 Results of the proposed method and previous methods for the English-to-Japanese translation. Bold text denotes no significance ($p < 0.01$) from the highest score obtained within bootstrap resampling [22].

手法	DL	test9		test10		学習時間
		RIBES	BLEU	RIBES	BLEU	
事前並べ替えなしの Moses	20	70.74	32.05	70.82	32.91	
規則に基づく既存手法 [19]	10	75.39	34.08	75.68	34.69	
先行研究の手法 [25]	10	77.04	34.29	77.21	35.48	1 日間
提案手法 (オラクル並べ替えラベルのみ)	10	76.88	34.13	77.20	35.64	1 日間
提案手法 (素性テンプレートのみ)	10	78.82	35.29	78.94	36.58	1 日間
提案手法	10	79.04	35.42	79.29	36.59	1 日間

表 10 日英翻訳における提案手法と既存手法の結果. 太字は bootstrap resampling [22] において最高スコアと統計的有意性 ($p < 0.01$) がないことを表す

Table 10 Results of the proposed and previous methods for the Japanese-to-English translation. Bold text denotes no significance ($p < 0.01$) from the highest score obtained within bootstrap resampling [22].

手法	DL	test9		test10		学習時間
		RIBES	BLEU	RIBES	BLEU	
事前並べ替えなしの Moses	20	69.88	30.12	70.22	30.51	
先行研究 [25] の並べ替えラベル+素性テンプレート	10	68.37	29.13	69.34	30.09	1 日間
提案手法 (オラクル並べ替えラベルのみ)	10	66.24	28.46	65.88	29.00	1 日間
提案手法 (素性テンプレートのみ)	10	77.49	33.25	77.77	34.31	1 日間
提案手法	10	77.97	33.55	78.07	34.13	1 日間

結果から, RIBES スコアと BLEU スコアの両面で, 提案手法がベースラインである事前並べ替え前の Moses の翻訳結果を有意に改善したことが分かる. 特に, 提案手法で Giza 分類器に代わり Nile 分類器を使用することでいっそうの改善が得られたため, 提案手法は単語対応付けが高精度であるほどより効果的であると示唆される. さらに, 提案手法は歪み制約なし ($DL = 0$) でも高い翻訳精度を達成しており, 事前並べ替え単独での並べ替えに成功していることが分かる.

3.6 既存手法との比較

表 9 および表 10 に, 提案手法を先行研究 [25] の事前並べ替え手法と規則に基づく事前並べ替え手法 [19] のそれぞれと比較した結果を示す. 表中のラベルのうち「提案手法 (オラクル並べ替えラベルのみ)」はオラクル並べ替えラベルと先行研究の素性テンプレートの組合せ, 「提案手法 (素性テンプレートのみ)」は, 先行研究の並べ替えラベルと提案手法の素性テンプレートの組合せ, 「提案手法」は, オラクル並べ替えラベルと提案手法の素性テンプレートの組合せを, それぞれ表している.

提案手法は英日翻訳と日英翻訳の両方で既存手法を上回った. 特に, 先行研究の手法は日英翻訳でベースラインである事前並べ替え前の Moses を下回る結果となり, 提案手法と顕著な違いが見られた. 翻訳精度が向上した原因

を分析すると, 新たな素性テンプレートの貢献が大きかった. さらに, オラクル並べ替えラベルと組み合わせることで, すべてのデータ・評価尺度で最高精度を達成している.

4. 関連研究

4.1 句構造と 2 値分類器に基づく事前並べ替え

提案手法は, 2.1 節で概略を説明した先行研究 [25] の手法に基づいている. この手法はオラクル並べ替えラベルを用いておらず, 代わりに, 並べ替えラベルを生成するためのヒューリスティクスを用いて学習を行っているが, その妥当性は今日まで検証されていなかった. 彼らのヒューリスティクスは, 学習データを Giza++ で単語対応付けし, 入力文側の単語を訳文側の単語インデックスを用いて整理する. 次に, 整理後の入力文から単語対応が重複するものを除き, 残りをを用いて 2 値分類器を学習する. 2 値分類器のラベルには, 単語対応から得られる反転・非反転のラベルを用いる. 素性には, 主辞, 非終端記号, 各ノードのスパン周辺の単語を用いている.

提案手法における Kendall's τ を用いた定式化を適用することで, 先行研究のヒューリスティクスの妥当性を検証する. 彼らのヒューリスティクスは, 単語対応が重複する事例を除いているため, 学習データ中の $\tau(\mathbf{a}) = 1$ または $\tau(\mathbf{a}) = -1$ を満たす事例のみを 2 値分類器の学習に用いていると見なせる. そのため, このヒューリスティクス

が生成する並べ替えラベルはオラクル並べ替えラベルと必ずしも一致しない。2.2 節の提案手法はこの点を問題視しているため、先行研究の枠組みのうちヒューリスティクスのみを受け継いでいない。計算量は、構文木を事前に計算しておくことができるため、Kendall's τ に基づく並べ替えラベルの算出と各ノードにおける 2 値分類の合計で $O(n^2 \log n) + O(n)$ である。

4.2 依存構造に基づく事前並べ替え

依存構造に基づく事前並べ替え手法 [11], [20] も提案されている。依存構造では、単純な 2 分木ではなく n 分木が対象となるため、並べ替えラベルの算出に $O(n!)$ を要し、合計で $O(n!) + O(n)$ の計算量となる。ランキングサポートベクタマシンを用いた順位付けによって近似的に計算する手法 [44] では、並べ替えラベルの算出に CKY アルゴリズムを用いて、合計で $O(n^3) + O(n)$ の計算量となる。

4.3 反転トランスダクション文法に基づく事前並べ替え

反転トランスダクション文法 [41] に基づく事前並べ替え手法 [6], [14], [26], [28] も提案されている。特に反転トランスダクション文法をより単純化したブラケット化トランスダクション文法 [41] に基づく手法 [6], [26], [28] は、既存の構文解析器を必要としない点が特徴的である。素性には、各ノードにおけるスパン長、左右子ノードのスパン長の関係、単語表層、非終端記号、Giza++ または Brown クラスタリング [2] を用いて得られる単語クラス、さらにフレーズテーブルを辞書として用いている。計算量は、CKY アルゴリズムを用いるため合計で $O(n^3)$ であったが、異なる定式化で $O(n^2)$ に改善した手法 [26] も提案された。とはいえ、構文木を事前に計算できないため、提案手法より長い学習時間が必要となる。

5. おわりに

本論文では、統計的機械翻訳のための統語に基づく事前並べ替え手法を提案した。提案手法は、句構造と 2 値分類器を用いて構文木の各ノードに反転・非反転のラベルを付与する先行研究の手法 [25] を、オラクル並べ替えラベルと素性テンプレートの 2 つの側面で改良した。オラクル並べ替えラベルには、Kendall's τ が最大になるように反転または非反転のラベルを付与した。また、素性テンプレートを改良し、2 値分類の正解率が大幅に改善することを確認した。なお先行研究の手法と同じく、提案手法は単純かつ句構造構文解析器が利用可能なすべての言語対に適用可能である。大規模な特許データを用いた英日・日英翻訳実験で、提案手法は先行研究の手法からの翻訳精度の大幅な改善を達成した。そのような高性能を言語依存性が強い手法を取り入れず達成できたことから、提案手法の有効性が示された。

謝辞 本論文の構成に協力いただいた、NTT コミュニケーション科学基礎研究所の森下陸氏に感謝する。

参考文献

- [1] Birch, A. and Osborne, M.: LRscore for Evaluating Lexical and Reordering Quality in MT, *Proc. Joint 5th Workshop on Statistical Machine Translation and MetricsMATR*, pp.327–332 (2010).
- [2] Brown, P.F., de Souza, P.V., Mercer, R.L., Della Pietra, V.J. and Lai, J.C.: Class-Based n -gram Models of Natural Language, *Computational Linguistics*, Vol.18, No.4, pp.467–479 (1992).
- [3] Chiang, D.: Hierarchical Phrase-Based Translation, *Computational Linguistics*, Vol.33, No.2, pp.201–228 (2007).
- [4] Collins, M., Koehn, P. and Kucerova, I.: Clause Restructuring for Statistical Machine Translation, *Proc. 43rd Annual Meeting of the Association for Computational Linguistics*, pp.531–540 (2005).
- [5] Den, Y., Nakamura, J., Ogiso, T. and Ogura, H.: A Proper Approach to Japanese Morphological Analysis: Dictionary, Model, and Evaluation, *Proc. 6th International Conference on Language Resources and Evaluation*, pp.1019–1024 (2008).
- [6] DeNero, J. and Uszkoreit, J.: Inducing Sentence Structure from Parallel Corpora for Reordering, *Proc. 2011 Conference on Empirical Methods in Natural Language Processing*, pp.193–203 (2011).
- [7] Ding, C., Utiyama, M. and Sumita, E.: Improving fast_align by Reordering, *Proc. 2015 Conference on Empirical Methods in Natural Language Processing*, pp.1034–1039 (2015).
- [8] Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R. and Lin, C.-J.: LIBLINEAR: A Library for Large Linear Classification, *Journal of Machine Learning Research*, Vol.9, pp.1871–1874 (2008).
- [9] Fang, T., Butler, A. and Yoshimoto, K.: Parsing Japanese with a PCFG Treebank Grammar, *Proc. 20th Meeting of the Association for Natural Language Processing*, pp.432–435 (2014).
- [10] Gao, Q. and Vogel, S.: Parallel Implementations of Word Alignment Tool, *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, pp.49–57 (2008).
- [11] Genzel, D.: Automatically Learning Source-side Reordering Rules for Large Scale Machine Translation, *Proc. 23rd International Conference on Computational Linguistics*, pp.376–384 (2010).
- [12] Goto, I., Chow, K.P., Lu, B., Sumita, E. and Tsou, B.K.: Overview of the Patent Machine Translation Task at the NTCIR-10 Workshop, *Proc. 10th NTCIR Workshop Meeting*, pp.260–286 (2013).
- [13] Goto, I., Lu, B., Chow, K.P., Sumita, E. and Tsou, B.K.: Overview of the Patent Machine Translation Task at the NTCIR-9 Workshop, *Proc. 9th NTCIR Workshop Meeting*, pp.559–578 (2011).
- [14] Goto, I., Utiyama, M., Sumita, E. and Kurohashi, S.: Preordering Using a Target-Language Parser via Cross-Language Syntactic Projection for Statistical Machine Translation, *Transactions on Asian and Low-Resource Language Information Processing*, Vol.14, No.3, pp.13:1–13:23 (2015).
- [15] Hall, D., Durrett, G. and Klein, D.: Less Grammar, More Features, *Proc. 52nd Annual Meeting of the As-*

- sociation for Computational Linguistics, pp.228–237 (2014).
- [16] Hoshino, S., Miyao, Y., Sudoh, K., Hayashi, K. and Nagata, M.: Discriminative Preordering Meets Kendall's τ Maximization, *Proc. 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pp.139–144 (2015).
- [17] Huang, L.: Forest Reranking: Discriminative Parsing with Non-Local Features, *Proc. 2008 Conference of the 46th Annual Meeting of the Association for Computational Linguistics and the Human Language Technologies*, pp.586–594 (2008).
- [18] Isozaki, H., Hirao, T., Duh, K., Sudoh, K. and Tsukada, H.: Automatic Evaluation of Translation Quality for Distant Language Pairs, *Proc. 2010 Conference on Empirical Methods in Natural Language Processing*, pp.944–952 (2010).
- [19] Isozaki, H., Sudoh, K., Tsukada, H. and Duh, K.: Head Finalization: A Simple Reordering Rule for SOV Languages, *Proc. Joint 5th Workshop on Statistical Machine Translation and MetricsMATR*, pp.244–251 (2010).
- [20] Jehl, L., de Gispert, A., Hopkins, M. and Byrne, B.: Source-side Preordering for Translation using Logistic Regression and Depth-first Branch-and-Bound Search, *Proc. 14th Conference of the European Chapter of the Association for Computational Linguistics*, pp.239–248 (2014).
- [21] Kendall, M.G.: A New Measure of Rank Correlation, *Biometrika*, Vol.30, No.1/2, pp.81–93 (1938).
- [22] Koehn, P.: Statistical Significance Tests for Machine Translation Evaluation, *Proc. 2004 Conference on Empirical Methods in Natural Language Processing*, pp.388–395 (2004).
- [23] Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A. and Herbst, E.: Moses: Open Source Toolkit for Statistical Machine Translation, *Proc. 45th Annual Meeting of the Association for Computational Linguistics: Interactive Poster and Demonstration Sessions*, pp.177–180 (2007).
- [24] Kudo, T., Yamamoto, K. and Matsumoto, Y.: Applying Conditional Random Fields to Japanese Morphological Analysis, *Proc. 2004 Conference on Empirical Methods in Natural Language Processing*, pp.230–237 (2004).
- [25] Li, C.-H., Li, M., Zhang, D., Li, M., Zhou, M. and Guan, Y.: A Probabilistic Approach to Syntax-based Reordering for Statistical Machine Translation, *Proc. 45th Annual Meeting of the Association of Computational Linguistics*, pp.720–727 (2007).
- [26] Nakagawa, T.: Efficient Top-Down BTG Parsing for Machine Translation Preordering, *Proc. 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pp.208–218 (2015).
- [27] Neubig, G.: Travatar: A Forest-to-String Machine Translation Engine based on Tree Transducers, *Proc. 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp.91–96 (2013).
- [28] Neubig, G., Watanabe, T. and Mori, S.: Inducing a Discriminative Parser to Optimize Machine Translation Reordering, *Proc. 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp.843–853 (2012).
- [29] Och, F.J.: Minimum Error Rate Training in Statistical Machine Translation, *Proc. 41st Annual Meeting of the Association for Computational Linguistics*, pp.160–167 (2003).
- [30] Och, F.J. and Ney, H.: A Systematic Comparison of Various Statistical Alignment Models, *Computational Linguistics*, Vol.29, No.1, pp.19–51 (2003).
- [31] Papineni, K., Roukos, S., Ward, T. and Zhu, W.-J.: BLEU: A Method for Automatic Evaluation of Machine Translation, *Proc. 40th Annual Meeting of the Association for Computational Linguistics*, pp.311–318 (2002).
- [32] Petrov, S., Barrett, L., Thibaux, R. and Klein, D.: Learning Accurate, Compact, and Interpretable Tree Annotation, *Proc. 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pp.433–440 (2006).
- [33] Petrov, S. and Klein, D.: Improved Inference for Unlexicalized Parsing, *Proc. 2007 Conference of Human Language Technologies and the North American Chapter of the Association for Computational Linguistics*, pp.404–411 (2007).
- [34] Riesa, J., Irvine, A. and Marcu, D.: Feature-Rich Language-Independent Syntax-Based Alignment for Statistical Machine Translation, *Proc. 2011 Conference on Empirical Methods in Natural Language Processing*, pp.497–507 (2011).
- [35] Shi, Q., Petterson, J., Dror, G., Langford, J., Smola, A. and Vishwanathan, S.: Hash Kernels for Structured Data, *Journal of Machine Learning Research*, Vol.10, pp.2615–2637 (2009).
- [36] Stolcke, A.: SRILM – An Extensible Language Modeling Toolkit, *Proc. 7th International Conference on Spoken Language Processing*, pp.901–904 (2002).
- [37] Sudoh, K., Wu, X., Duh, K., Tsukada, H. and Nagata, M.: Post-ordering in Statistical Machine Translation, *Proc. Machine Translation Summit XIII*, pp.316–323 (2011).
- [38] Talbot, D., Kazawa, H., Ichikawa, H., Katz-Brown, J., Seno, M. and Och, F.J.: A Lightweight Evaluation Framework for Machine Translation Reordering, *Proc. 6th Workshop on Statistical Machine Translation*, pp.12–21 (2011).
- [39] Tillman, C.: A Unigram Orientation Model for Statistical Machine Translation, *Proc. 2004 Conference of Human Language Technologies and the North American Chapter of the Association for Computational Linguistics*, pp.101–104 (2004).
- [40] Tromble, R. and Eisner, J.: Learning Linear Ordering Problems for Better Translation, *Proc. 2009 Conference on Empirical Methods in Natural Language Processing*, pp.1007–1016 (2009).
- [41] Wu, D.: Stochastic Inversion Transduction Grammars and Bilingual Parsing of Parallel Corpora, *Computational Linguistics*, Vol.23, No.3, pp.377–403 (1997).
- [42] Xia, F. and McCord, M.: Improving a Statistical MT System with Automatically Learned Rewrite Patterns, *Proc. 20th International Conference on Computational Linguistics*, pp.508–514 (2004).
- [43] Yamada, K. and Knight, K.: A Syntax-based Statistical Translation Model, *Proc. 39th Annual Meeting of the Association for Computational Linguistics*, pp.523–530 (2001).
- [44] Yang, N., Li, M., Zhang, D. and Yu, N.: A Ranking-

based Approach to Word Reordering for Statistical Machine Translation, *Proc. 50th Annual Meeting of the Association for Computational Linguistics*, pp.912–920 (2012).

- [45] 小澤俊介, 内元清貴, 伝 康晴: 長単位解析器の異なる品詞体系への適用, *自然言語処理*, Vol.21, No.2, pp.379–401 (2014).



星野 翔 (正会員)

2011年工学院大学情報学部コンピュータ科学科卒業。2017年総合研究大学院大学複合科学研究科情報学専攻単位取得後退学。同年みらい翻訳入社。機械翻訳の研究に従事。言語処理学会会員。



宮尾 祐介 (正会員)

1998年東京大学理学部情報科学科卒業。2001年同大学助手, のち助教。2010年国立情報学研究所准教授。同研究所教授を経て, 2018年より東京大学大学院情報理工学系研究科教授。博士(情報理工学)。構文解析とその応用の研究に従事。人工知能学会, 言語処理学会, ACL各会員。



須藤 克仁 (正会員)

2000年京都大学工学部情報学科卒業。2002年同大学大学院情報学研究科修士課程修了。同年NTT入社。2017年より奈良先端科学技術大学院大学准教授。博士(情報学)。音声言語処理, 機械翻訳の研究に従事。日本音響学会, 言語処理学会, ACL各会員。



林 克彦 (正会員)

2013年奈良先端科学技術大学院大学情報科学研究科博士課程修了。同年NTTコミュニケーション科学基礎研究所入所。2018年より大阪大学産業科学研究所助教。博士(工学)。自然言語処理の研究に従事。人工知能学会, 言語処理学会, ACL各会員。



永田 昌明 (正会員)

1987年京都大学大学院工学研究科修士課程修了。同年NTT入社。現在, NTTコミュニケーション科学基礎研究所主幹研究員(上席特別研究員)。博士(工学)。自然言語処理の研究に従事。電子情報通信学会, 人工知能学会, 言語処理学会, ACL各会員。