

# 構文情報が機械翻訳に及ぼす影響の分析

丹生 伊左夫<sup>1,a)</sup> Graham Neubig<sup>1,b)</sup> 小林 和也<sup>1</sup> Sakriani Sakti<sup>1</sup> 戸田 智基<sup>1</sup> 中村 哲<sup>1</sup>

**概要:** 統計的機械翻訳では広く使われるフレーズベース翻訳以外にも、階層的フレーズベース翻訳、構文情報を用いた翻訳などの方式が提案されている。そこで、本稿では京都フリー翻訳タスクに対して、フレーズベース翻訳、階層的フレーズベース翻訳さらに統語ベース翻訳の手法を、BLEU と RIBES という2つの自動翻訳評価尺度で調査した。さらに、機械翻訳を実行する前に、原言語文を目的言語に近い語順に並び替える手法の効果についても分析した。その結果、実験で用いたタスクにおける英日翻訳で、各翻訳手法の効果と特徴を確認した。

## 1. はじめに

対訳データを利用してモデルを構築し、翻訳を行う統計的機械翻訳は、大量のデータが利用可能になるとともに、ますます実用化されている。特に、対訳データから得られるフレーズ間の翻訳確率を利用し、目的言語として適切な語順になるよう並び替えを行うフレーズベース機械翻訳 (phrase-based machine translation: PBMT) [1] が主流として利用されてきた。しかし、英語と日本語のように語順が大きく異なる言語間において、フレーズベース翻訳の精度が比較的低いことも知られている。そこで、翻訳精度向上のために、特に並び替えに着目した様々な枠組みの統計翻訳手法が提案されている。

その代表として、フレーズベース翻訳の手軽な学習という利点を残しつつ、並び替えの精度を高めた階層的フレーズベース翻訳 (hierarchical phrase-based machine translation: Hiero) [2] がある。さらに、原言語側の文を解析し、その結果に基づいて翻訳を行う統語ベース機械翻訳 (syntax-based machine translation) [3] も存在する。また、原言語を目的言語らしい語順に並び替えてから PBMT や Hiero で翻訳を行う事前並び替え (preordering) [4] もあり、構文解析結果を人手のルールで並び替える手法 [5][6] や、学習に基づく手法 [4][7] が提案されている。

このように様々な手法が提案されている一方で、評価実験は同じ枠組みで翻訳を行う手法との比較や、代表的な翻訳手法であるフレーズベース翻訳との比較が大半である。つまり、統計的機械翻訳全体における大域的な手法の位置

付けがはっきりと見えておらず、どの枠組みの手法が全体としてどれほど効果的なのか明確でない。そこで、本稿では英語から日本語への翻訳で、京都関連の Wikipedia 記事を対象とした京都フリー翻訳タスク (KFTT)\*<sup>1</sup> に対して、様々な枠組みの手法の翻訳精度や特徴を分析し、特に構文情報を考慮した手法に着目して各翻訳手法の位置付けを行う。

結果として、フレーズベース翻訳よりも構文情報を考慮した翻訳手法の方が翻訳精度が高いことを確認した。また、統語ベースと事前並び替えを比較したところ、BLEU [8] による自動評価と許容性による人手評価に関しては、Neubig が提案した学習に基づく事前並び替え法 [9] が最も高い精度を実現した。しかし、決定的な差が見えておらず、各手法の長所と短所を詳細な分析によって明らかにする。

また、代表的な構文解析器やアライメント手法を適用した際に、翻訳精度がどのように変化するかについても調査を行う。

## 2. 機械翻訳手法

機械翻訳において、原言語文を目的言語文に翻訳する際、語彙選択と並び替えを正確に行う必要がある。これらの問題を解決すべく、様々な翻訳手法が提案されており、本節で代表的な手法について紹介する。

### 2.1 フレーズベース機械翻訳

統計翻訳で最も代表的な手法として、Koehn らのフレーズベース翻訳 (PBMT) [1] がある。この手法は、原言語文と目的言語文の対訳データから単語アライメントを取り、アライメント結果を基に複数の単語からなるフレーズを

<sup>1</sup> 奈良先端科学技術大学院大学  
Nara Institute of Science and Technology

a) isao-ni@is.naist.jp

b) neubig@is.naist.jp

\*<sup>1</sup> <http://www.phontron.com/kfft/index-ja.html>

抽出し、各フレーズにスコア付けを行う。そして、スコアの付与されたフレーズ同士の辞書を利用して、原言語のフレーズを目的言語のフレーズへ変換する。ただし、スコア付けは、各フレーズの翻訳確率と単語の翻訳確率、そしてペナルティを考慮して行われる。さらに、並べ替えモデルを用いてフレーズの並び替えに対してスコア付けを行い、翻訳結果の自然性を言語モデルで評価する。訳出の際、この3つのモデルのスコアを考慮し、その重み付き和が最大となる候補を選択する。

このフレーズベース翻訳は、翻訳対象である2言語間の対訳データさえあれば翻訳可能であり、構文解析器が存在しない言語間でも適用可能である。しかし、文の構造を考慮しない手法であるが故に、単語の並び替えが効果的に行えない傾向にある。文法的に似た言語間では大きな問題とならないが、英語と日本語のように語順が大きく異なる言語間では、高い翻訳精度を得られない。

## 2.2 階層的フレーズベース翻訳

フレーズベース翻訳の汎用性を残して、並び替えの精度を高めた方法として、階層的フレーズベース翻訳 (Hiero) [2] が Chiang により提案されている。フレーズベース翻訳のように単語列を扱うだけでなく、下記のようなルールも利用する。

[X1] *visit* [X2] → [X1] は [X2] を訪れる

X1 や X2 に該当する単語列をまず翻訳し、ルールの X1 と X2 の代わりに代入していくことで翻訳を行う。つまり、適用する各ルールと、X1, X2 に当てはまる様々な候補の確率とルール自体の確率、さらに言語モデルを考慮することで、最適な翻訳文を選択する。従って、並び替えを個別なモデルで行うのではなく、並び替えと語彙選択を同時に行うルールを利用することで、フレーズベース翻訳よりも精度の高い目的言語文の語順を得ることが可能となる。

階層的フレーズベース翻訳は、通常のフレーズベース翻訳よりも並び替えを効果的に行えると報告されているが、翻訳に多くの時間を要し、各モデルのサイズも大きくなる [2]。

## 2.3 統語ベース機械翻訳

前節で述べたフレーズベース翻訳と階層的フレーズベース翻訳とは異なり、文の構文情報を利用した統計的機械翻訳手法も提案されている [7]。特に本節では、原言語側だけで構文情報を利用する T2S (tree-to-string) 翻訳や F2S (forest-to-string) 翻訳を紹介する。

これらの翻訳方式は原言語文を構文解析し、構文木の情報を用いて翻訳を行う。また、翻訳は Hiero と同じように変数を含むルールを利用するが、単語列ではなく、原言語

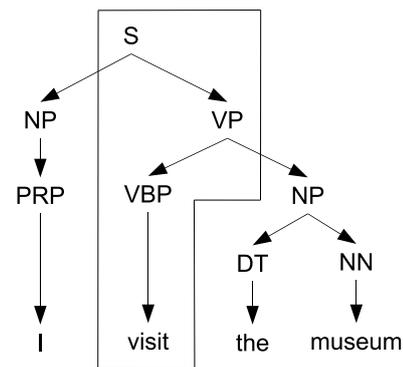


図 1 構文解析例

文の部分木の構造として利用する。

例えば、図 1 の場合、木構造中の 2 つの NP が、前節の Hiero の説明で用いたルールの X1 と X2 に該当し、線で囲まれた部分木が “visit” に該当する。部分木の構造を保ちながら、置き換え可能な 2 つの NP をルールに直接含み、原言語文の構文解析結果と適合する翻訳ルールを用いて目的原言語文を生成し、言語モデルでスコア付けを行う。

一文に対して 1 つの構文木を構築し、翻訳に用いる手法が T2S であり、構文木の複数の解釈を 1 つのグラフ構造で保持し、確率を付与して訳文を選択する手法が F2S である。F2S 翻訳は一文ずつに複数の構文情報を考慮することで、構文解析器が最適と判断した解析結果だけに縛られない訳出が可能である。

統語ベース翻訳は、構文解析器が付与する豊富な情報を利用して並び替えを正確に行え、訳出候補の数も少ないため探索空間が小さく翻訳に要する時間が短い。しかし一方で、解析結果に適合した訳だけを出力するため、構文解析器の精度に大きく依存する。さらに、精度の良い構文解析器が利用できない言語に対しては、構文情報を用いて翻訳を行えないという短所もある。

## 3. 事前並び替え手法

フレーズベース翻訳は、2.1 節で述べたように計算量やモデルの問題から、大域的な並び替えに弱い翻訳手法である。語順の大きく異なる言語間で、フレーズベース翻訳という枠組みを変えずに並び替えの精度を向上させる方法として、事前並び替えという手法が存在する [4]。翻訳を行う前に、予め原言語文を目的言語に近い語順に並び替える手法であり、本節ではその代表的な手法について紹介する。

### 3.1 Head Finalization

英語と日本語の統語構造はそれぞれ SVO 型と SOV 型言語であるため、語順が大きく異なる。そこで、日本語が主要部終端型 (head-final) の言語であることに着目し、シンプルなルールで並び替える方法として Head Finalization [6] が、Isozaki らにより提案されている。Head Finalization

は、構文木の各非終端ノードにおいて、その子ノードの中で主要部を末尾に移動させるものである。また、日本語の助詞に相当する語として特殊な記号を挿入し、the や a などの冠詞を削除し、英語の複数形を単数形に直すというシンプルな手法である。

単純なアルゴリズムで高速に動作する上に、フレーズベース翻訳と組み合わせることで翻訳精度の向上も報告されている [6]。しかし、主要部を単純に終端に移動させるだけでは対応できない文もある。例えば、“I have no time” をそのまま翻訳できる語順とならない。Head Finalization を適用しても “I *va1* no time *va2* have” となり、「私は無い時間を持っている」として適切ではない日本語表現になってしまう。また、構文解析結果に基づいて並び替えを行うため、構文解析器の解析誤りが直接並び替えに影響を及ぼす。

### 3.2 Lader

Head Finalization は人手のルールを利用した事前並び替え手法であったが、人手ルールを用いない手法として Neubig らによりアライメントデータから学習可能な並べ替え手法 Lader [9] が提案されている。具体的には、アライメント付きデータから並べ替えの精度が最大となるように、括弧反転トランスダクション文法 (bracketing transduction grammar: BTG) [10] を教師なしで学習する手法である。これに基づいて、入力文の並べ替えを行ってから通常のフレーズベース手法で翻訳を行う。なお、識別学習に基づく手法を用いているため、原言語文の品詞や構文解析結果などの情報を柔軟に取り入れることで、精度の向上が実現可能であると報告されている。

この手法は両言語に精通している言語学者により作成されたルールを必要としないことから、汎用的に利用できる事前並び替え手法である。その一方、対象言語間のアライメントデータがないと高精度な並べ替えを実現できないため、アライメントデータを作成する必要がある。

## 4. 実験

本節では、上記の翻訳方式と事前並び替え法を比較するために行った実験について述べる。

### 4.1 実験条件

実験で用いたデータである KFTT に関して、学習データ (Train) と開発データ (Dev)、テストデータ (Test) の詳細を表 1 に示す。

英語側のトークン化には Stanford Parser [11]<sup>\*2</sup> を、日本語側の単語分割には KyTea [12]<sup>\*3</sup> を利用した。また、原言語である英語の文に対して用いる構文解析器について、

<sup>\*2</sup> <http://nlp.stanford.edu/software/lex-parser.shtml>

<sup>\*3</sup> <http://www.phontron.com/kytea/index-ja.html>

表 1 京都フリー翻訳タスクのデータ内訳

Dataset	Lang	Words	Sentences	Average sentence length
Train	En	9.11M	405k	22.5
	Ja	9.42M	405k	23.2
Dev	En	24.3k	1.17k	20.8
	Ja	26.8k	1.17k	23.0
Test	En	26.7k	1.16k	23.0
	Ja	28.3k	1.16k	24.5

T2S と F2S では Egret [13]<sup>\*4</sup> を利用し、事前並び替えの手法である Head Finalization では Enju [14]<sup>\*5</sup> を用いた。

そして、各手法において対訳データ間のアライメントを取るツールとして、GIZA++ [15]<sup>\*6</sup> を用いて実験した。ただし、事前並び替え手法である Lader で学習するためのアライメントデータは KFTT の一部として配布されている人手アライメントを用いた。また、目的言語である日本語の言語モデルは IRSTLM<sup>\*7</sup> を用いて 5-gram で学習し、各素性の重みは BLEU が最大となるように MERT [16] を用いて調整した。各実験の翻訳精度は、BLEU と RIBES [17] の 2 つの自動翻訳評価尺度で測った。

構文情報を利用しない手法として、Moses [18] に実装されているフレーズベース翻訳 (PBMT) と階層的フレーズベース翻訳 (Hiero) を実験し、PBMT はデフォルトの設定を用いた。Hiero のルールに当てはまる候補の探索ではビーム幅を 5000 に設定し、max-chart-span は 20 で実験した。

そして、統語ベース翻訳を行う手法として、Travatar [19]<sup>\*8</sup> に実装されている構文木から単語列への翻訳 (T2S) と、複数の構文木を考慮する翻訳 (F2S) を実験した。T2S と F2S は、全ての実験において、Travatar の pop limit を 5000 に設定した。

事前並び替え手法として、Head Finalization と Lader をそれぞれ適用し、翻訳は PBMT と Hiero で実験を行った。Head Finalization は我々が論文に沿って再実装したものを用いた。Lader<sup>\*9</sup> の学習と適用の際、単語や単語クラス、Egret により推定された品詞と構文木を素性に利用した。

### 4.2 翻訳方式の比較

前節で述べた各翻訳手法の翻訳精度を表 2 に示す。ただし、表中の太字は、危険率 5% の下、ブートストラップ・リサンプリング法 [20] を用いて、各手法と精度の最も高い手法 (Lader で事前並び替えした PBMT) を比較した結果、統計的有意差がない数値を表している。

<sup>\*4</sup> <https://code.google.com/p/egret-parser/>

<sup>\*5</sup> <http://www.nactem.ac.uk/enju/index.ja.html>

<sup>\*6</sup> <https://code.google.com/p/giza-pp/>

<sup>\*7</sup> <http://hlt.fbk.eu/en/irstlm>

<sup>\*8</sup> <http://www.phontron.com/travatar/>

<sup>\*9</sup> <http://www.phontron.com/lader/>

表 2 京都フリー翻訳タスクに対する各翻訳手法の精度

Preordering	Method	BLEU	RIBES
None	PBMT	21.79	68.21
	Hiero	20.32	67.30
	T2S	22.56	<b>71.43</b>
	F2S	23.30	<b>71.34</b>
Head Finalization	PBMT	22.60	65.23
	Hiero	20.70	66.07
Lader	PBMT	<b>23.90</b>	<b>71.08</b>
	Hiero	22.24	70.07

表 3 各翻訳手法で用いる言語モデル, 翻訳モデル, 並び替えモデルのサイズ (バイト)

Method	LM	TM	RM
PBMT	99M	421M	176M
Hiero	99M	561M	-
T2S / F2S	99M	<b>241M</b>	-
HF-PBMT	99M	411M	175M
HF-Hiero	99M	571M	-
Lader-PBMT	99M	423M	180M
Lader-Hiero	99M	585M	-

まず, 構文情報を用いない PBMT と Hiero に比べて, 構文情報を考慮する手法すべてにおいて, BLEU と RIBES の両方が向上した. さらに, 事前並び替えを行うことで PBMT と Hiero の両方の翻訳精度が向上し, Lader で事前並び替えを行い, PBMT で翻訳した場合は, 統語ベース翻訳よりも BLEU が高い結果となった.

KFTT における英日翻訳に関して, T2S や F2S よりも Lader で事前並び替えした方が BLEU が高くなった理由の一つとして, Lader は KFTT のデータで学習されたモデルを利用しているため, KFTT に対しては精度良く並び替えを行えたことが翻訳精度に影響したと考えられる. そして, RIBES に関して, フレーズベース翻訳よりも Lader を用いた方が高くなっていることから, 上記の理由も影響し, より大域的な並び替えを実現できていると判断できる. 一方, Lader で事前並び替えした PBMT や Hiero よりも, T2S や F2S の方が RIBES が高いことから, 構文情報を考慮した方が並び替えの精度向上に寄与していると考えられる.

また, モデルのサイズが翻訳のスピードとメモリ効率に影響を与えるため, 上記の実験で用いた言語モデル (language model: LM) と翻訳モデル (translation model: TM), また PBMT で用いる並び替えモデル (reordering model: RM) のそれぞれについて, 各モデルのサイズを表 3 に示す. ただし, 翻訳の性能だけを調査するために, 全ての実験を通して同一の言語モデルで統一した.

表 3 より, PBMT で用いる翻訳モデルに比べて, 翻訳ルールを付与した Hiero の翻訳モデルの方がサイズが大きくなっている. 一方で, T2S や F2S 翻訳が, PBMT や Hiero の翻訳モデルより小さいサイズでありながら, 高い

表 4 構文解析器の違いによる翻訳精度の比較

Parser	Method	BLEU	RIBES
Stanford parser	T2S	21.18	70.61
	F2S	22.56	71.43
Egret	T2S	22.56	71.43
	F2S	23.30	71.34

翻訳精度を得られることが確認された. また, 事前並び替えを用いた翻訳において, PBMT と Hiero の両方で, 翻訳モデルは大きなサイズの変化が見られなかった.

#### 4.3 構文解析器の影響

本節では, 構文解析器の違いによる翻訳精度の違いについて記す. 木構造に基づく 2 つの翻訳手法 T2S と F2S は, 原言語文を構文解析することから, 構文解析の精度が翻訳精度に直接影響を及ぼす. そこで, この影響度合いを確認するために, 構文解析器として Egret と Stanford Parser を変えた場合の翻訳精度を表 4 に示す. ただし, Stanford Parser は複数の構文木をグラフ構造で出力可能な仕様となっていないため, T2S 翻訳のみを実験した.

結果として, Stanford Parser を用いて構文解析した場合よりも, Egret を用いて構文解析した方が翻訳精度が高いことが確認された. 表 4 から, 構文解析器は翻訳精度に大きく影響を及ぼすことが確認されたが, 機械翻訳との相性としては Egret で構文解析した方が統語ベース機械翻訳で高い性能を得られることがわかった. この結果は, 必ずしも構文解析の精度が Egret の方が高いことを示すわけではないが, Egret が用いている潜在的なクラスに基づく構文解析法は, 様々なタスクにおいて高い精度を実現しているという報告があり [21], 本実験のデータに対しても同じ結果になっている可能性が高いと考えられる.

#### 4.4 アライメントの影響

各翻訳手法において, 翻訳モデルを作成する際に, 原言語文と目的言語文の対訳データから得られるアライメント精度が, 翻訳精度に影響するのを確認する. アライメントツールとして, GIZA++に加えて Nile [22]\*<sup>10</sup>で実験を行った. なお, Nile は教師あり学習に基づくため, 正解アライメントとして KFTT で配布されているものを利用した. Nile は構文情報を利用することで, 高い翻訳精度を実現しており, 英語側の構文解析器として Egret を用い, 日本語側の構文解析器として Eda[23] と Travatar ツールキットの係り受け解析結果を句構造木へ変更するスクリプトを利用した. データの 10% を学習データから取り除いてアライメント精度を評価したところ, F 値で測ったアライメント精度は GIZA++ で 58.32%, Nile で 64.22% であり, Nile の構文情報の利用と教師あり学習により大幅なアライメント精度の向上が実現されていることが分かる.

\*<sup>10</sup> <https://code.google.com/p/nile/>

表 5 アライメントの違いによる翻訳精度の比較

Method	GIZA++		Nile	
	BLEU	RIBES	BLEU	RIBES
PBMT	21.79	68.21	21.64	68.11
Hiero	20.32	67.30	22.15	69.72
T2S	22.56	71.43	22.88	71.92
F2S	23.30	71.34	22.62	72.45

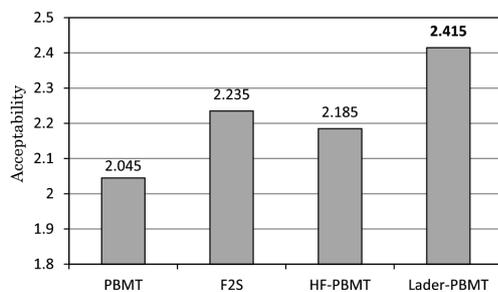


図 2 各翻訳手法における主観評価結果

このアライメント精度の向上が翻訳にも影響するのかを調査するために、GIZA++と Nile のアライメントを用いた翻訳モデルの学習と評価を行った。実験はフレーズベース翻訳として PBMT と Hiero、統語ベース翻訳として T2S と F2S を行い、その結果を表 5 に示す。この結果から、フレーズベース翻訳について、PBMT は BLEU と RIBES の両方であまり変化がなかったが、Hiero は大幅な向上が見られた。統語ベース翻訳について、BLEU に関しては T2S で向上し、F2S で減少が見られたが、RIBES に関しては共に向上する結果となった。特に、F2S では大幅な RIBES の向上が確認された。

#### 4.5 主観評価による翻訳精度の比較

フレーズベース翻訳と統語ベース翻訳、そして 2 種類の事前並び替えを用いた翻訳の計 4 手法で、BLEU の最も高かった PBMT、F2S、HF-PBMT、そして Lader-PBMT について、主観評価を行った結果を本節で記す。各手法の原言語文の長さ 1 から 30 単語の 200 文に対して、1 から 5 の 5 段階許容性評価 [24] を付与してもらった。評価者は英語を母国語にしている一名で、日本語に熟達している。

各手法における主観評価結果の平均値を図 2 に示す。平均値が高いほど、主観評価の結果が良いことを表し、また、ブートストラップ・リサンプリング法を用いて Lader-PBMT と他の 3 手法で、危険率 5% の下、有意差が見られた。

図 2 に示すように、Lader-PBMT が主観評価で最も高い結果となった理由として、この主観評価は評価効率のため単語数 30 以下の文のみで実験を行っていることが考えら

れる。さらに長い文での評価も行えば、より RIBES が高く、大域的な並び替えに強い F2S が上回る可能性もあると考えられる。

## 5. 詳細分析

本節では、各翻訳手法の特徴を記すと共に、機械翻訳例を用いて、結果の分析を行う。

まず、各手法の全体的な特徴として、PBMT はフレーズ単位で正しく翻訳できている場合が多いが、フレーズの並び替えで間違いが生じている傾向があった。つまり、短い文は比較的適切に翻訳できているが、長い文ではより多くの並び替え誤りが見られた。表 6 の Example 1 に示すように、PBMT において各フレーズは正確に翻訳できていることがわかる。しかし、「藤原氏 (Fujiwara clan)」と「流れを汲む (descended from)」と「公家 (court nobility)」の各フレーズの並び替えに失敗し、結果として適切でない翻訳結果が出力されている。一方、構文解析に基づく翻訳では、構文解析器が正しく構文木を構築したため、これに基づいて適切な並び替えを行い、翻訳に成功している。さらに、Head Finalization で事前並び替えした英文は、“saionji family *va0* fujiwara clan from descended court nobility *va2* were.” となり、精度良く並び替えできた結果、翻訳結果も適切な文を出力している。そして、Lader で事前並び替えした例文は、“the saionji family the fujiwara clan from descended nobility court were.” となり、Head Finalization と同様、精度の高い翻訳結果が得られた。

また、構文情報を考慮した T2S や F2S に関しては、学習データの構文解析結果として得られる情報を利用して、もし学習データで出現した単語に付与された品詞と、テストデータで同じ単語に付与された品詞が異なる場合、未知語として処理されてしまう。翻訳結果を確認しても、T2S と F2S を用いた場合に、未知語に分類され、正しく翻訳されなかった箇所が多く存在した。例えば、表 6 の Example 2 に示すように、“amateru” という単語の翻訳に際して、フレーズベース翻訳を用いた PBMT と HF-PBMT、Lader-PBMT は全て漢字の「天照」やカタカナの「アマテル」に翻訳できているが、F2S だけが未知語と判断し、結果として翻訳されず英単語のまま出力されていた。一方で、このような未知語処理の問題はあるものの、並び替えは比較的正確に実行できていることが確認できた。Example 1 と 2 の両方で、F2S の並び替えは高い精度で並び替えを実現できている。

そして、事前並び替えを用いた翻訳は、PBMT よりも精度よく並び替えを行えている傾向にあった。しかし、事前に並び替えた語順に忠実に翻訳を行うことが多いため、事前並び替えで精度良く並び替えが行われなければ、その結果に依存して翻訳結果も精度が低下する。具体例として、Example 3 の HF-PBMT は翻訳結果の語順が適切でなく、

表 6 各手法の機械翻訳例

	Method	Acceptability	Translated sentence
Example 1	Source sentence	-	The Saionji Family were court nobility descended from the Fujiwara clan.
	Reference	-	西園寺(さいおんじ)家は、藤原氏の流れを汲む公家。
	PBMT	1	西園寺家の流れを汲む公家は藤原氏。
	F2S	5	西園寺家は藤原氏の流れを汲む公家であった。
	HF-PBMT	5	西園寺家は藤原氏の流れを汲む公家である。
Lader-PBMT	5	西園寺家は藤原氏の流れを汲む公家。	
Example 2	Source sentence	-	A male god called "Amateru" was widely worshipped.
	Reference	-	「アマテル」と称された男神が広く祀られていた。
	PBMT	4	男性神として「天照(アマテル)」が広く信仰されている。
	F2S	3	amateru と呼ばれる男神が広く信仰されている。
	HF-PBMT	3	『天照(アマテル)と呼ばれる男神は広く信仰されていた。』
Lader-PBMT	4	「天照(アマテル)」と呼ばれる男神広く信仰されていた。	
Example 3	Source sentence	-	1217 - Studied under Myozen, a disciple of Eisai, at Kennin-ji Temple.
	Reference	-	建保5年(1217年)建仁寺にて栄西の弟子明全に師事。
	PBMT	1	-建保5年(1217年)明全に師事し、栄西の弟子に建仁寺である。
	F2S	2	建保5年(1217年)-明全、栄西の弟子に師事し、建仁寺であった。
	HF-PBMT	3	建仁寺で、栄西の弟子 myozen-建保5年(1217年)に師事している。
Lader-PBMT	5	建保5年(1217年)-建仁寺で、栄西の弟子、明全に師事した。	

訳出に失敗していることがわかる。そこで、問題を確認するために、Head Finalization で並び替えられたテストデータの原言語文を見ると、“kennin-ji temple at , - , eisai of disciple myozen under studied 1217 va2 .” と並び替えられており、事前並び替えの時点で、語順が適切でない結果となっていた。さらに、Enju で構文解析した結果から文中の主要部を確認したところ、“1217” が誤って主要部と判断され、構文解析の段階で失敗していた。このように、構文解析誤りが原因の一つとなり、結果として、翻訳精度の低下と、BLEU や RIBES が低下する原因になっている可能性がある。一方、Lader による並び替えでは、“1217 - kennin-ji temple at , eisai of a disciple , myozen under studied .” となり、適切な語順への並び替えが反映され、正確に訳出されている。

## 6. おわりに

本稿は、英語から日本語への翻訳について、様々な翻訳方式と事前並び替え法を比較した。また、統語ベース翻訳に関して、構文解析器の及ぼす影響についても調査し、翻訳モデルを学習するために必要なアライメント手法の影響も実験した。そして、構文情報に基づく翻訳の諸課題を指摘し、さらなる翻訳精度向上の可能性を示唆した。

本稿では、英語から日本語への翻訳で様々な手法を比較したが、他の言語対や分野における翻訳の場合も同様の結果となるとは限らないため、今後実験すべき課題である。そして、統語ベース翻訳では前節で述べたように、未知語処理の問題があることと、実験から構文解析器の違いが翻訳精度に大きく影響することがわかっている。そこで、未

知語処理の問題を解消し、構文解析に基づく翻訳とより相性の良い構文解析器を用いることで、翻訳精度の向上が期待できる。

## 参考文献

- [1] Phillip Koehn, Franz Josef Och, and Daniel Marcu. Statistical phrase-based translation. In *Proc. HLT*, pages 48–54, Edmonton, Canada, 2003.
- [2] David Chiang. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2), 2007.
- [3] Kenji Yamada and Kevin Knight. A syntax-based statistical translation model. In *Proc. ACL*, 2001.
- [4] Fei Xia and Michael McCord. Improving a statistical MT system with automatically learned rewrite patterns. In *Proc. COLING*, 2004.
- [5] Michael Collins, Philipp Koehn, and Ivona Kucerova. Clause restructuring for statistical machine translation. In *Proc. ACL*, 2005.
- [6] Hideki Isozaki, Katsuhito Sudoh, Hajime Tsukada, and Kevin Duh. Head finalization: A simple reordering rule for SOV languages. In *Proc. WMT and MetricsMATR*, 2010.
- [7] Haitao Mi and Liang Huang. Forest-based translation rule extraction. In *Proc. EMNLP*, pages 206–214, 2008.
- [8] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proc. ACL*, pages 311–318, Philadelphia, USA, 2002.
- [9] Graham Neubig, Taro Watanabe, and Shinsuke Mori. Inducing a discriminative parser to optimize machine translation reordering. In *Proc. EMNLP*, pages 843–853, Korea, July 2012.
- [10] Dekai Wu. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–403, 1997.
- [11] Dan Klein and Christopher D. Manning. Accurate unlexicalized parsing. In *Proc. ACL*, pages 423–430, 2003.
- [12] Graham Neubig, Yosuke Nakata, and Shinsuke Mori.

- Pointwise prediction for robust, adaptable Japanese morphological analysis. In *Proc. ACL*, pages 529–533, Portland, USA, June 2011.
- [13] Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. Learning accurate, compact, and interpretable tree annotation. In *Proc. ACL*, pages 433–440, 2006.
- [14] Miyao Yusuke and Tsujii Jun’ichi. Maximum entropy estimation for feature forests. In *Proceedings of the second international conference on Human Language Technology Research*, pages 292–297, 2002.
- [15] Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, 2003.
- [16] Franz Josef Och. Minimum error rate training in statistical machine translation. In *Proc. ACL*, 2003.
- [17] Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. Automatic evaluation of translation quality for distant language pairs. In *Proc. EMNLP*, pages 944–952, 2010.
- [18] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *Proc. ACL*, pages 177–180, Prague, Czech Republic, 2007.
- [19] Graham Neubig. Travatar: A forest-to-string machine translation engine based on tree transducers. In *Proc. ACL Demo Track*, Sofia, Bulgaria, August 2013.
- [20] Philipp Koehn. Statistical significance tests for machine translation evaluation. In *Proc. EMNLP*, 2004.
- [21] Jonathan K Kummerfeld, David Hall, James R Curran, and Dan Klein. Parser showdown at the wall street corral: an empirical investigation of error types in parser output. In *Proc. EMNLP*, pages 1048–1059, 2012.
- [22] Jason Riesa and Daniel Marcu. Hierarchical search for word alignment. In *Proc. ACL*, pages 157–166, 2010.
- [23] Daniel Flannery, Yusuke Miyao, Graham Neubig, and Shinsuke Mori. Training dependency parsers from partially annotated corpora. In *Proc. IJCNLP*, pages 776–784, Chiang Mai, Thailand, November 2011.
- [24] Isao Goto, Bin Lu, Ka Po Chow, Eiichiro Sumita, and Benjamin K. Tsou. Overview of the patent machine translation task at the ntcir-9 workshop. In *Proceedings of NTCIR*, volume 9, pages 559–578, 2011.