NAIST-IS-MT1351088

修士論文

統語ベース翻訳における統語的前処理

波多腰 優斗

2015 年 2 月 24 日

奈良先端科学技術大学院大学
情報科学研究科 情報科学専攻

本論文は奈良先端科学技術大学院大学情報科学研究科に
修士（工学）授与の要件として提出した修士論文である.

波多腰 優斗

審査委員：

中村 哲 教授	（主指導教員）
松本 裕治 教授	（副指導教官）
戸田 智基 准教授	（副指導教官）
Graham Neubig 助教	（副指導教官）
Sakriani Sakti 助教	（副指導教官）

統語ベース翻訳における統語的前処理*

波多腰 優斗

内容梗概

統計的機械翻訳 (statistical machine translation, SMT) では，ある言語で記述された原言語文を翻訳先の目的言語文へ統計モデルを用いて自動変換する．SMT の統計モデルにおいて，既存の学習方法では適切な翻訳規則を学習できないことが多く，翻訳精度が低下してしまう問題が指摘されてきた．これに対して，言語的な知見に基づく前処理が適用された学習データを用いて統計モデルを学習することで，翻訳精度の向上を図る手法が数多く提案されている．特にフレーズベース機械翻訳 (phrase-based machine translation, PBMT) において，統語情報を用いたルールに基づく前処理の効果が示されており，翻訳精度が改善されている．一方で，他の翻訳方式である統語ベース翻訳に対してはこのような前処理の適用例が少ない．そこで本研究では，PBMT において有効な英日翻訳のためのルールに基づく統語的前処理を，統語ベース翻訳に適用し，その効果を確かめる．また，人手によるルールやアノテーション済みのデータを利用せずに，対訳データのみを用いて言語的な知見をモデル化し，統語的前処理に適用する枠組みについても提案する．実験によって，ルールに基づく統語的前処理は，PBMT に適用した場合ほど改善幅を示さないものの，統語ベース翻訳に対しても十分な効果があることが確認された．

キーワード

機械翻訳, 統語ベース翻訳, tree-to-string 翻訳, 前処理

*奈良先端科学技術大学院大学 情報科学研究科 情報科学専攻 修士論文, NAIST-IS-MT1351088, 2015 年 2 月 24 日.

Syntactic Preprocessing for Syntax-based Machine Translation*

Yuto Hatakoshi

Abstract

Statistical machine translation (SMT) has been researched actively throughout the world as a way to translate one language to other languages using statistical models. However, the failure to obtain applicable translation patterns can cause reductions in accuracy. To solve this problem, many preprocessing methods which convert training data into a form appropriate for the training process have been developed. In particular, several preprocessing techniques using syntactic information and linguistically motivated rules have been proposed to improve the quality of phrase-based machine translation (PBMT) output. On the other hand, there has been little work on similar techniques in the context of other translation formalisms such as syntax-based SMT. In this research, we examine whether the sort of rule-based syntactic preprocessing approaches that have proved beneficial for PBMT can contribute to syntax-based SMT. In addition, we propose a new method using parallel corpus to incorporate syntactic information without hand-crafted rules and annotated corpus. Specifically, we tailor a highly successful preprocessing method for English-Japanese PBMT to syntax-based SMT, and find that while the gains achievable are smaller than those for PBMT, significant improvements in accuracy can be realized.

Keywords:

machine translation, syntax-based machine translation, tree-to-string translation, preprocessing

*Master's Thesis, Department of Information Science, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-MT1351088, February 24, 2015.

目次

図目次

表目次

第 1 章 諸言

1.1 背景

近年，国際化や通信手段の発達に伴い，異なる言語で記述された文面を目にする機会が増加している．しかし，言語の壁が障害となり，このような文面を理解するためには多大な労力を要する．そのため，原言語文を入力とし自動的に翻訳先の目的言語へ変換する機械翻訳 (Machine translation, MT) の翻訳精度を高めることが重要なタスクとなっている．MT の方式の中でも，統計的機械翻訳 (statistical machine translation, SMT) は，二言語の対訳データから機械翻訳システムを自動的に構築することが可能であり，大量のデータを活用することで多言語対や特定の分野への適応が短期間かつ低コストで実現できるため，盛んに研究されている．

SMT では，複数の単語からなるフレーズ間の翻訳確率を計算し，目的言語として適切な語順となるように並べ替えモデルによる局所的なフレーズの移動を行う，フレーズベース機械翻訳 (phrase-based machine translation, PBMT)[1] が広く用いられている．PBMT は翻訳モデルの学習が容易であり，多くの言語対で高い精度での翻訳が可能である一方，形態的・統語的な情報の扱いに乏しいという問題点もある．また，並べ替えモデルによる長距離の並べ替え確率の推定が困難であり，英語と日本語のように語順が大きく異なる言語対では翻訳精度が低下することが知られている．

このような統計モデルの欠点を補い，翻訳精度を向上させるために，学習データに対して前処理を適用する手法が提案されている．例えば，動詞の接頭辞を分割して目的言語との単語対応を取りやすくする前処理 [2] や，名詞の格の不一致・動詞の活用誤り減らすための前処理 [3] などにより PBMT の翻訳精度が向上しており，統語的前処理の有効性が示されている．特に注目されている前処理の一つとして事前並べ替え [4] が挙げられる．PBMT の並べ替えモデルは，長距離の並べ替え確率を正確に推定することが困難であることから，英語と日本語のように語順が大きく異なる言語対では翻訳精度が低下する問題もある．そのため，原言語文を目的言語に近い語順に並び替える事前並べ替えを適用してから PBMT による翻訳を行うことで，翻訳精度が大幅に改善されることが知られている．

Head Finalization [5] は，英日翻訳において有効な統語的前処理手法であり，二

言語間の統語的な構造に基づくシンプルなルールによって，PBMT の翻訳精度が飛躍的に向上することが知られている．Head Finalization の処理の中でも，日本語の主辞後置性を考慮し，英語の主辞を句の末尾に移動させる事前並べ替え処理が特に有効であるとされている．しかし，英語文を日本語に近い語順に並べ替える処理以外にも，より日本語に近い文とするために冠詞の削除，助詞の挿入といった語順の操作とは関係のない，単語に関する前処理も行っている．このような並べ替え以外の処理も重要な要素となっている場合があり，これらの統語的前処理による効果は PBMT に限ったものではないと考えられる．

PBMT 以外の SMT の翻訳手法としては，文の構文解析結果に基づいて翻訳を行う統語ベース翻訳 [6] がある．統語ベース翻訳は翻訳パターンに構文木の部分木の構造を用いており，文法構造が大きく異なる言語対において PBMT よりも正確な翻訳を実現することが多い．統語ベース翻訳に対する統語的前処理としては，構文木の変換によって文の構成要素と単語アライメントの対応を改善するための前処理 [7] が提案され，翻訳精度を向上させている．統語ベース翻訳に対しても統語的前処理の有効性は示唆されており，前述したルールに基づく並べ替え処理や，単語に関する前処理の適用についても検証する余地がある．

1.2 研究目的

本研究では，統語ベース翻訳に対する 2 つの統語的前処理手法を提案し，翻訳精度を向上させることを目的とする．

1 つ目の手法として，ルールに基づく統語的前処理を提案する．具体的には，Head Finalization の各処理がどの程度 PBMT の翻訳精度の向上に貢献しているかを検証するとともに，統語ベース機械翻訳に対しても同様の処理が有効であるかを確かめる．さらに，デコード時に用いる素性として追加するなど，ソフトな制約としてルールを取り入れた場合の翻訳精度についても調査する．

2 つ目の手法として，SMT の学習に用いる対訳データから，言語的な知見に基づくルールを抽出し，統語的前処理に取り入れる方法を提案する．Head Finalization のような目的言語側の文法知識を用いた統語的前処理は，単純な処理によって高い効果が得られる一方，特定の言語対に特化した処理となり多言語対への応用は困難

である．本研究ではこの点も踏まえ，対訳データを用いて目的言語側の情報を原言語側に自動的に付与し，それをモデル化することで，様々な言語対の統語的前処理を可能とする手法についても提案する．

1.3 論文構成

本論文では，2 章で様々な統計的機械翻訳の要素技術について説明し，PBMT と統語ベース翻訳の違いを示す．3 章では PBMT および統語ベース翻訳に対する統語的前処理の先行研究について説明し，問題点を明らかにする．4 章では，統語ベース翻訳の翻訳精度を向上させることを目的とした統語的前処理について提案する．5 で提案法の実験的評価を行い，その結果を示した上で考察する．最後に 6 章で本論文のまとめと今後の課題について述べる．

第 2 章 機械翻訳

機械翻訳とは，ある言語で表現された自然言語を別の言語へ自動的に変換する技術である．機械翻訳の手法として，ルールベース機械翻訳 [8]，用例ベース機械翻訳 [9]，統計的機械翻訳 [10] が存在する．

ルールベース機械翻訳は人手で作成した翻訳ルールを用いる手法であり，1980 年代までは積極的に研究が行われた．ルールに当てはまる入力文に対しては高精度な翻訳結果が得られるが，ルールにない例外的な文に対応できない問題点がある．また，両言語に精通した言語学者によって翻訳ルールが作成されるため，コストが高く，多言語化が困難である．

用例ベース機械翻訳は，大量の対訳データ (2 言語で同じ内容を記述したテキストの集まり) を模倣して新たな文を翻訳する手法であり，1980 年代に提案された．翻訳能力は対訳データの大きさに依存し，入力文と似た文が対訳データに含まれている場合に良い翻訳結果が得られる．一方で，用例の選択やスコア化の過程がヒューリスティックである問題点がある．

統計的機械翻訳とは，大量の対訳データから自動的に対応する単語やフレーズなどを学習，翻訳する手法であり，1980 年代後半からは盛んに研究がされている．対訳データがあればシステムを構築できるため，多言語化が容易である．大量の対訳データを獲得することは困難であったが，近年は Web の発達などに伴う言語資源の充実化に伴い大規模な対訳データが利用可能になっており，研究・開発に拍車がかかっている．

次節以降では，統計的機械翻訳の要素技術について説明し (2.1 節)，統計的機械翻訳の代表的な翻訳方式であるフレーズベース機械翻訳 (2.2 節) と統語ベース機械翻訳 (2.3 節)，機械翻訳の自動評価尺度 (2.4 節) について述べる．

2.1 統計的機械翻訳

統計的機械翻訳は，雑音のある通信路モデル [11] に基づいている．ある原言語文 $\boldsymbol{f}$ に対して，目的言語の訳出候補集合を $\mathcal{E}(\boldsymbol{f})$ とし，$\boldsymbol{f}$ が目的言語文 $\boldsymbol{e} \in \mathcal{E}(\boldsymbol{f})$ に翻訳される確率 $Pr(\boldsymbol{e}|\boldsymbol{f})$ を全ての $\boldsymbol{e}$ について計算可能とする．統計的機械翻訳では，

以下のように $Pr(\boldsymbol{e}|\boldsymbol{f})$ を最大化する $\hat{\boldsymbol{e}} \in \mathcal{E}(\boldsymbol{f})$ を求めることにより目的言語文を生成する．

$$\hat{\boldsymbol{e}} = \underset{\boldsymbol{e} \in \mathcal{E}(\boldsymbol{f})}{\arg\max}\, Pr(\boldsymbol{e}|\boldsymbol{f}) \tag{2.11}$$

式 (2.11) は，ベイズの定理を用いて以下のように書き換えられる．

$$\hat{\boldsymbol{e}} = \underset{\boldsymbol{e} \in \mathcal{E}(\boldsymbol{f})}{\arg\max}\, \frac{Pr(\boldsymbol{f}|\boldsymbol{e})Pr(\boldsymbol{e})}{Pr(\boldsymbol{f})} \tag{2.12}$$

$$= \underset{\boldsymbol{e} \in \mathcal{E}(\boldsymbol{f})}{\arg\max}\, Pr(\boldsymbol{f}|\boldsymbol{e})Pr(\boldsymbol{e}) \tag{2.13}$$

式 (2.13) において，目的言語文 $\boldsymbol{e}$ が与えられた時の原言語文 $\boldsymbol{f}$ の条件付き確率 $Pr(\boldsymbol{f}|\boldsymbol{e})$ は翻訳モデル (translation model, TM) と呼ばれる．$Pr(\boldsymbol{e})$ は言語モデル (language model, LM) と呼ばれ，訳出する目的言語文の流暢性の向上に寄与する．統計的機械翻訳は，$Pr(\boldsymbol{f}|\boldsymbol{e})Pr(\boldsymbol{e})$ を最大化する問題として表現され，原言語が目的言語へ符号化された過程を逆にたどることから復号化と呼ばれる．

2.1.1 翻訳モデル

翻訳モデルとは，翻訳の確からしさを示すモデルである．上述した統計的機械翻訳の基本的な枠組みに基づいて，1990 年代初頭にモデル 1 からモデル 5 へと徐々に複雑になっていく 5 つの翻訳モデル「IBM モデル」が提案された [12]．IBM モデルでは，対訳文の単語単位の対応を表現した単語アライメント $\boldsymbol{\alpha}$ を導入し，翻訳モデル $Pr(\boldsymbol{f}|\boldsymbol{e})$ は，条件付き確率 $Pr(\boldsymbol{f},\boldsymbol{\alpha}|\boldsymbol{e})$ を全ての可能な $\boldsymbol{\alpha}$ により周辺化したものと考える．

$$Pr(\boldsymbol{f}|\boldsymbol{e}) = \sum_{\alpha} Pr(\boldsymbol{f},\boldsymbol{\alpha}|\boldsymbol{e}) \tag{2.14}$$

このような単語単位での翻訳は，英語やフランス語など近い言語対の翻訳に有効であった．しかし，慣用句などの表現をうまく翻訳できない問題点があり，フレー

ズを最小単位とするフレーズベース機械翻訳 [1] が提案された．その後，文法構造が大きく異なる言語対の翻訳精度を改善するため，文の構文情報を用いて翻訳を行う統語ベース翻訳 [6] も提案された．これらの翻訳モデルについては 2.2 節，2.3 節で詳細に述べる．

2.1.2 言語モデル

言語モデルは，与えられた単語の並びがどの程度発生するかを確率的に表したモデルであり，機械翻訳システムの訳出の流暢性を保証するのに重要な役割を果たす．ある文 $\boldsymbol{e}$ の長さを I とすると，$\boldsymbol{e} = e_1 \cdots e_I$(以下，$\boldsymbol{e} = e_1^I$) であり，言語モデルは以下の式で表される．

$$P(e_1^I) = \prod_{i=1}^{I+1} P(e_i|e_1^{i-1}) \tag{2.15}$$

各単語の生起確率を式 (2.15) のまま求めるのは現実的に困難なため，広く用いられる N-gram モデルでは，e_i の生起確率が直前の $N-1$ 単語にのみ依存するという制約を設け，以下の様な近似を行う．

$$P(e_1^I) \simeq \prod_{i=1}^{I+1} P(e_i|e_{i-N+1}^{i-1}) \tag{2.16}$$

式 (2.16) における条件付き確率は以下の様な最尤推定により求めることができる．

$$P(e_i|e_{i-N+1}^{i-1}) = \frac{C_{train}(e_{i-N+1}^{i})}{C_{train}(e_{i-N+1}^{i-1})} \tag{2.17}$$

ここで，$C_{train}(\cdot)$ は学習データにおける単語列の出現頻度を表す．このような最尤推定による条件付き確率の求め方において，学習データに含まれない N-gram の確率が 0 になることが問題となる．この問題を解決するために，平滑化という手法が存在する．平滑化の基本的な考え方は，最尤推定により求められる N-gram の確率

$P(e_i|e_{i-N+1}^{i-1})$ と (N − 1)-gram の確率 $P(e_i|e_{i-N+2}^{i-1})$ を組み合わせることである．代表的な平滑化の手法として，線形補間や Kneser-Ney 法などがある [13].

2.1.3 対数線形モデル

式 (2.13) に示したベイズの定理による定式化には，翻訳モデルと言語モデルのスコアを翻訳精度が向上するように適切に重み付けできない問題点がある．各モデルのパラメータをドメインに合わせて設定することにより翻訳精度は向上する．そこで，式 (2.11) の生成モデルをより一般化するため，事後確率 $Pr(\boldsymbol{e}|\boldsymbol{f})$ を対数線形モデルにより直接表現する．

$$\begin{aligned}\hat{\boldsymbol{e}} &= \underset{\boldsymbol{e}\in\mathcal{E}(\boldsymbol{f})}{\arg\max}\, Pr(\boldsymbol{e}|\boldsymbol{f}) \\ &= \underset{\boldsymbol{e}\in\mathcal{E}(\boldsymbol{f})}{\arg\max}\, \frac{\exp\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{h}(\boldsymbol{f},\boldsymbol{e})\right)}{\sum_{\boldsymbol{e}'}\exp\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{h}(\boldsymbol{f},\boldsymbol{e}')\right)} \qquad (2.18) \\ &= \underset{\boldsymbol{e}\in\mathcal{E}(\boldsymbol{f})}{\arg\max}\, \boldsymbol{w}^{\mathrm{T}}\boldsymbol{h}(\boldsymbol{f},\boldsymbol{e}) \qquad (2.19)\end{aligned}$$

ここで，$\boldsymbol{h}(\cdot)$ は $\boldsymbol{w}$ により重み付けされる M 次元の素性ベクトルである．$\boldsymbol{h}(\cdot)$ は翻訳モデルや言語モデルだけでなく，並び替えの正しさなど任意の情報を素性として用いることができ，より柔軟なモデルの設計が可能となる．

2.1.4 最適化

式 (2.19) における重み $\boldsymbol{w}$ は，エラー最小化学習 (minimum error rate training, MERT)[14] などの最適化手法により，実際のテストデータに近い開発データ (development data) に適合するように学習される．N 文から構成される開発データ $\{\boldsymbol{f}^{(i)},\boldsymbol{e}^{(i)}\}_{i=1}^{N}$ が存在すると仮定すると，MERT により最適化される重み $\hat{\boldsymbol{w}}$ は以下のように表せる．

$$\hat{\boldsymbol{w}} = \arg\max_{\boldsymbol{w}} \sum_{i=1}^{N} l(\arg\max_{\boldsymbol{e}\in\mathcal{E}(\boldsymbol{f})} \boldsymbol{w}^{\mathrm{T}}\boldsymbol{h}(\boldsymbol{f}^{(i)},\boldsymbol{e}),\boldsymbol{e}^{(i)}) \tag{2.110}$$

ここで，$l(\cdot)$ は翻訳精度を評価する尺度であり (2.4 節で後述)，MERT はそれを直接最適化するアルゴリズムである．式 (2.110) は勾配を求めることができないため，Powell 法や Downhill-Simplex 法などの勾配を必要としない最適化法 [15] が用いられる．MERT は翻訳精度を評価する尺度 $l(\cdot)$ を直接損失関数として利用し，それを最適化できるが，最適解を得るために要する時間が用いる素性の数に比例して増加するというデメリットがある．

2.2 フレーズベース機械翻訳

統計的機械翻訳で最も代表的な手法として，フレーズ単位で翻訳を行うフレーズベース機械翻訳 (phrase-based machine translation, PBMT)[1] がある．

図 2.1 のように英語の入力文 “John hit a ball” が与えられた場合，最初に入力文を翻訳可能なフレーズに分割する．ここで，フレーズとは言語学的な境界により決定される単位ではなく，1 単語以上から成る単なる単語の連続的な系列を意味する．原言語側の各フレーズは，翻訳モデルにより目的言語のフレーズへ変換される．さらに並び替えモデルによりフレーズの並び替えを行い，翻訳結果を生成する．

以上のような PBMT の翻訳過程は，式 (2.13) に対して隠れ変数 $\boldsymbol{\phi}$，$\boldsymbol{\alpha}$ を導入して変形を施すことで，以下の様な生成モデルとして表現できる．

$$\hat{\boldsymbol{e}} = \arg\max_{\boldsymbol{e}} \sum_{\boldsymbol{\phi},\boldsymbol{\alpha}} Pr(\boldsymbol{f},\boldsymbol{\phi},\boldsymbol{\alpha}|\boldsymbol{e})Pr(\boldsymbol{e}) \tag{2.21}$$

$$\approx \arg\max_{\boldsymbol{e}} \sum_{\boldsymbol{\phi},\boldsymbol{\alpha}} Pd(\boldsymbol{f},\boldsymbol{\alpha}|\boldsymbol{\phi})P_{\phi}(\boldsymbol{\phi}|\boldsymbol{e})P_{lm}(\boldsymbol{e}) \tag{2.22}$$

式 (2.22) において，原言語文 $\boldsymbol{f}$ と目的言語文 $\boldsymbol{e}$ は L 個のフレーズからなる文 $\bar{\boldsymbol{f}} = \bar{f}_1,\cdots,\bar{f}_L$，$\bar{\boldsymbol{e}} = \bar{e}_1,\cdots,\bar{e}_L$ にそれぞれ分割されるとする．$\boldsymbol{\alpha} = \{\alpha_1,\cdots,\alpha_L\}$ はフレーズ単位のアライメントを表すベクトルであり，$\bar{e}_k$ が対応する $\bar{\boldsymbol{f}}$ の位置を α_k で示している．$\boldsymbol{\phi}$ は L 個のフレーズペア (対応するフレーズの対) を示し，$\boldsymbol{\alpha}$ の対

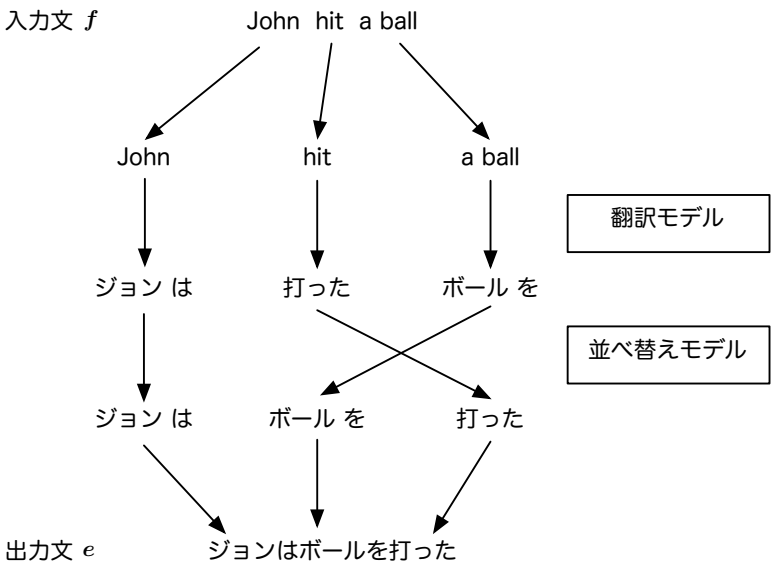

図 2.1　フレーズベース機械翻訳

応情報を基に目的言語側の順に並べられる．式 (2.22) において，$Pd(\boldsymbol{f},\boldsymbol{\alpha}|\boldsymbol{\phi})$ を並べ替えモデル (reordering model)，$P_{\phi}(\boldsymbol{\phi}|\boldsymbol{e})$ をフレーズ翻訳モデル (phrase translation model) と呼ぶ．$P_{lm}(\boldsymbol{e})$ は言語モデル (language model) である．

2.2.1　フレーズ翻訳モデル

本節では，PBMT の翻訳モデルの学習方法について述べる．翻訳モデルの学習のために原言語文と目的言語文の単語アライメントを取る必要があるが，ここでは単語アライメントがすでに得られているものとして説明する．単語アライメントを獲得するための手法としては，IBM モデルに基づく手法 [12] などが提案されている．

単語アライメントがすでに付与された対訳データ $\langle \boldsymbol{f},\boldsymbol{e},\boldsymbol{a} \rangle$ がすでに存在する場合，単語アライメントが内部で閉じているフレースペアを列挙する．図 2.2 の例では，以下のような対応を列挙できる．

$$\{ \langle (ジョン),(John) \rangle , \langle (ジョン,は),(John) \rangle , \langle (ボール),(a,ball) \rangle , \langle (打,っ,た),(hit) \rangle , \langle (ボール,を,打,っ,た),(hit,a,ball) \rangle , \cdots \}$$

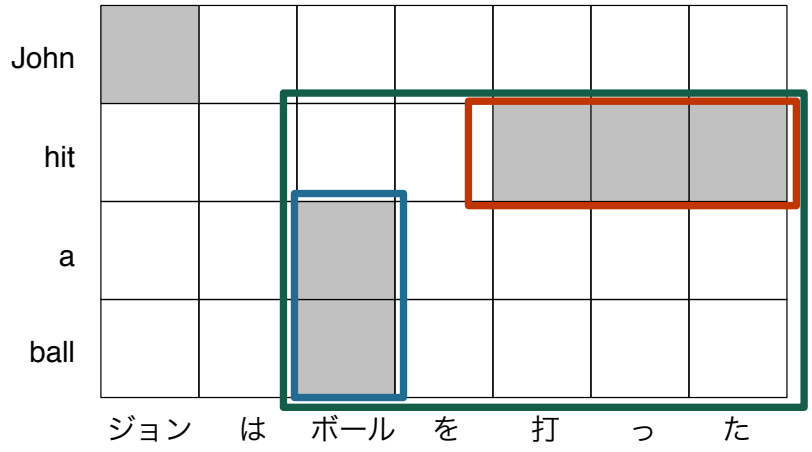

図 2.2　単語アライメント

$\langle$(ジョン, は, ボール), ($John, a, ball$)$\rangle$ などのペアは単語アライメントが内部で閉じていないため獲得できない．対訳データ $\langle \boldsymbol{f}, \boldsymbol{e}, \boldsymbol{a} \rangle$ から獲得できるフレーズペアの和集合 Φ は以下のように求められる．

$$\Phi = \bigcup_{\langle \boldsymbol{f}, \boldsymbol{e}, \boldsymbol{a} \rangle \in \langle F, E, A \rangle} \Phi_{\langle \boldsymbol{f}, \boldsymbol{e}, \boldsymbol{a} \rangle} \tag{2.23}$$

$\Phi_{\langle \boldsymbol{f}, \boldsymbol{e}, \boldsymbol{a} \rangle}$ は $\boldsymbol{f}$，$\boldsymbol{e}$ に対してアライメント $\boldsymbol{a}$ が内部で閉じているすべてのフレーズペアを表す．抽出されたフレーズペア $\langle \boldsymbol{f}, \boldsymbol{e} \rangle \in \Phi$ の頻度を $\mathrm{count}(\bar{\boldsymbol{f}}, \bar{\boldsymbol{e}})$ とすると，$P_\phi(\bar{\boldsymbol{f}} | \bar{\boldsymbol{e}})$ を最尤推定により求めることができる．

$$P_\phi(\bar{\boldsymbol{f}} | \bar{\boldsymbol{e}}) = \frac{\mathrm{count}(\bar{\boldsymbol{f}}, \bar{\boldsymbol{e}})}{\sum_{\langle \bar{\boldsymbol{f}}', \bar{\boldsymbol{e}} \rangle \in \Phi} \mathrm{count}(\bar{\boldsymbol{f}}', \bar{\boldsymbol{e}})} \tag{2.24}$$

2.2.2　並べ替えモデル

フレーズベース機械翻訳における並べ替えモデルは，原言語側の距離を以下の 3 つの並べ替えの方向 $O \in \{m, s, d\}$ により抽象化している [16].

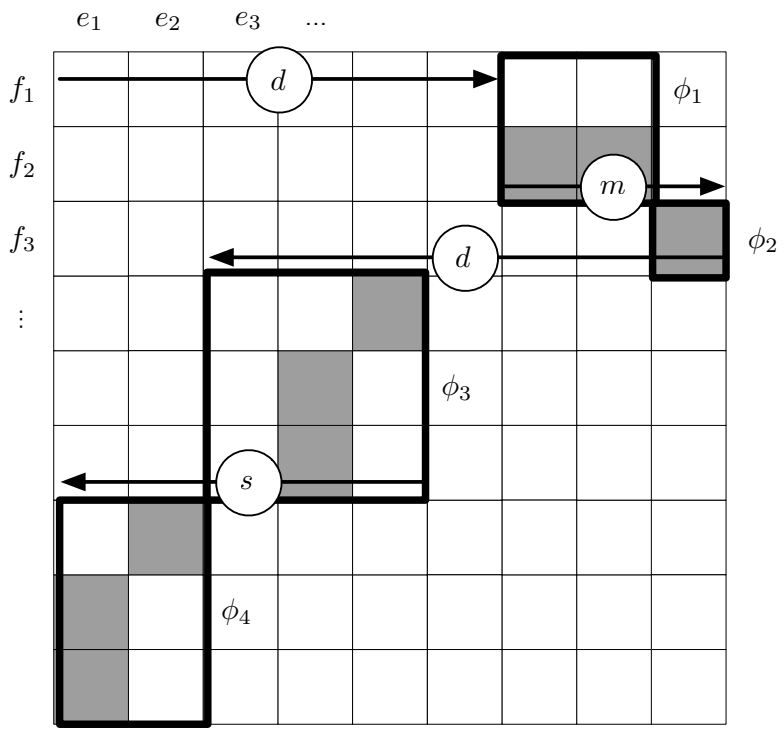

図 2.3　並べ替えモデル

単調 ($O = m$):　原言語と目的言語で二つのフレーズが連接し，順番に並ぶ
交換 ($O = s$):　原言語と目的言語で二つのフレーズが連接し，逆順に並ぶ
不連続 ($O = d$):　単調および交換でない

図 2.3 にフレーズペア ϕ_k と ϕ_{k+1} において単調，交換，不連続の例を示した．例えば，ϕ_1 と ϕ_2 は原言語側で連接しており，順番に並んでいるため「単調」な関係にある．学習データから抽出されたフレーズペアの並べ替えモデルのスコアは，各方向ごとの出現頻度 count(O,ϕ) に基づいて，最尤推定を用いて求められる．例えば単調方向 ($O = m$) のスコアは以下のように求められる．

$$p_{O=m}(m|\phi) = \frac{\text{count}(m,\phi)}{\sum_{o' \in \{m,s,d\}} \text{count}(o',\phi)} \tag{2.25}$$

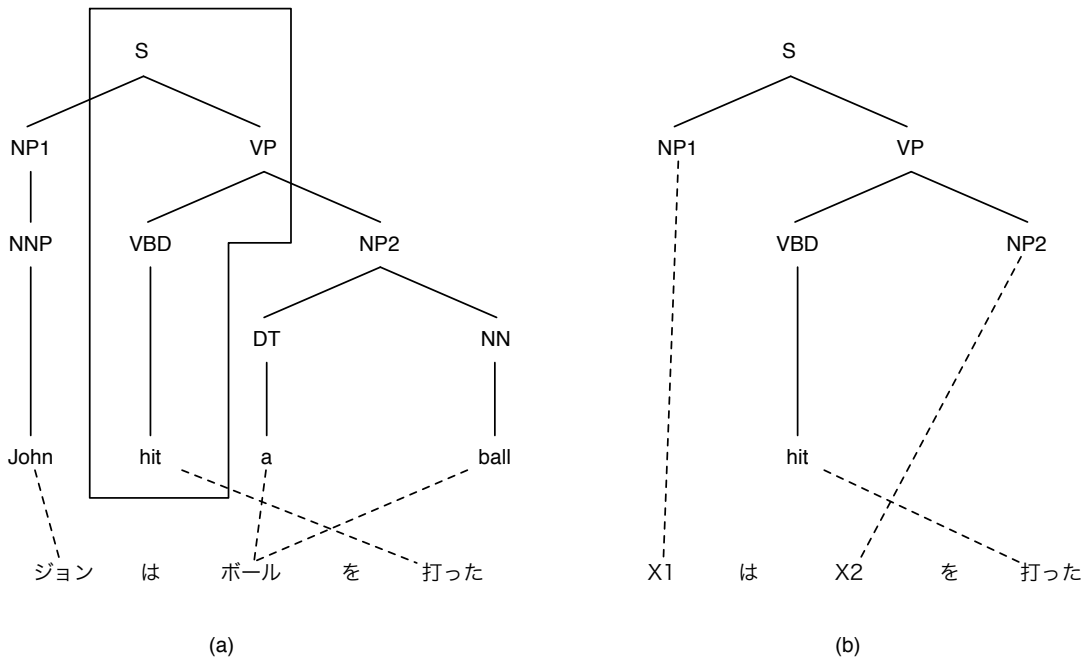

図 2.4　T2S の翻訳パターン

2.3　統語ベース機械翻訳

2.2 節で述べた PBMT は統語情報を利用しない手法であり，フレーズ単位の対応を求めることにより，どのようなペアの言語に対しても機械翻訳を実現可能であった．それに対して統語ベース機械翻訳 (syntax-based machine translation)[6] は，文の構文情報を用いて翻訳を行う方式である．統語ベース翻訳では，翻訳パターンを構文木の部分木の構造を用いて与えるため，PBMT よりも並び替えを正確に行うことができる．本節では，統語ベース翻訳の中でも，原言語側の構文情報として木構造を利用した，同期木置換文法 (STSG)[17] に基づく Tree-to-String 翻訳 (T2S)[18] について説明し，PBMT との比較について述べる．

2.3.1　Tree-to-String 翻訳

T2S は原言語文の構文解析結果を利用することで，二言語間の関係を統語的な構造により捉えることができ，正確な翻訳が可能となる．翻訳パターンは PBMT のよ

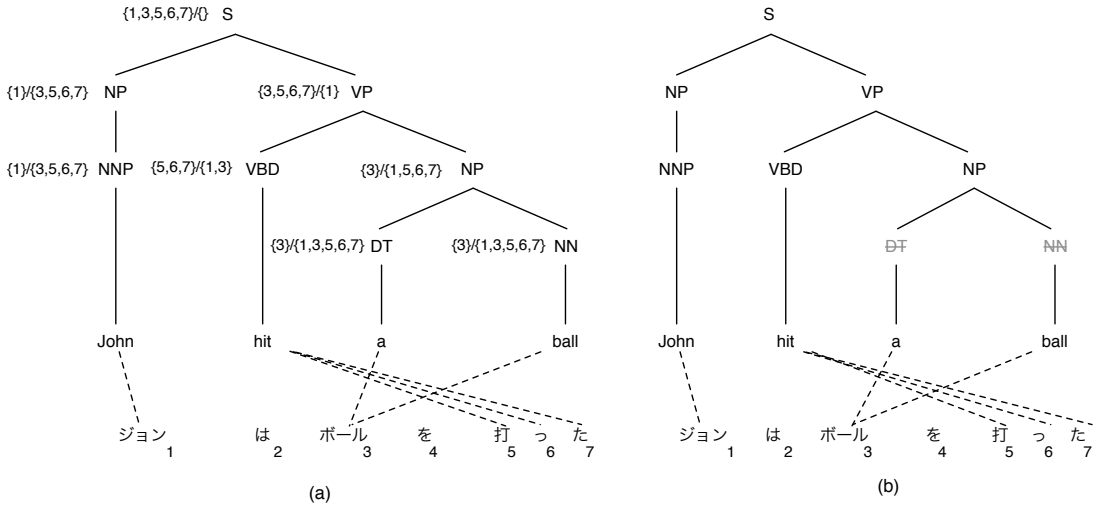

図 2.5　アライメントスパン (a) と許容可能な頂点 (b)

うに単語列ではなく，変数 x を含むルールも利用して原言語文の部分木の構造として表現される．例えば，図 2.4(a) の実線で囲まれた部分木に着目すると，図 2.4(b) の部分木を抽出できる．この部分木は動詞 “hit” と二つの名詞句 “NP1”，“NP2” から構成されており，以下の翻訳パターンを得られる．以下の例は，置き換え可能な２つの NP を翻訳パターンに直接含んでおり，部分木の構造を保ちながら X1，X2 に当てはまる候補の確率と翻訳パターン自体の確率を考慮して訳文を生成する．

S((X1:NP1)(VP(VBD hit)(X2:NP2))) → X1 は X2 を打った

T2S 翻訳は，木構造に対して二言語間で書き換えを行う文法規則である同期木置換文法 (synchronous tree substitution grammar, STSG) に基いて翻訳パターンの学習を行う．STSG の各ルールは，PBMT の翻訳モデルの学習時と同様に，対訳データ $\langle F, E, A \rangle$ からヒューリスティックを用いて自動的に獲得される．

対訳文 $\langle \boldsymbol{f}, \boldsymbol{e}, \boldsymbol{a} \rangle \in \langle F, E, A \rangle$ に対して，原言語側の句構造解析木を T_f とし，同期木置換ルールを抽出する方法について具体例を用いて説明する．原言語側の構文木の各頂点 $v \in T_f$ が被覆する終端記号が目的言語のある単語列に対応するとする．このとき，対応する各単語のアライメント集合 $\pi(v)$ をアライメントスパン (alignment span) と呼ぶ．さらに，$\pi(v)$ の補集合 $\bar{\pi}(v)$ は補完アライメントスパン (complement

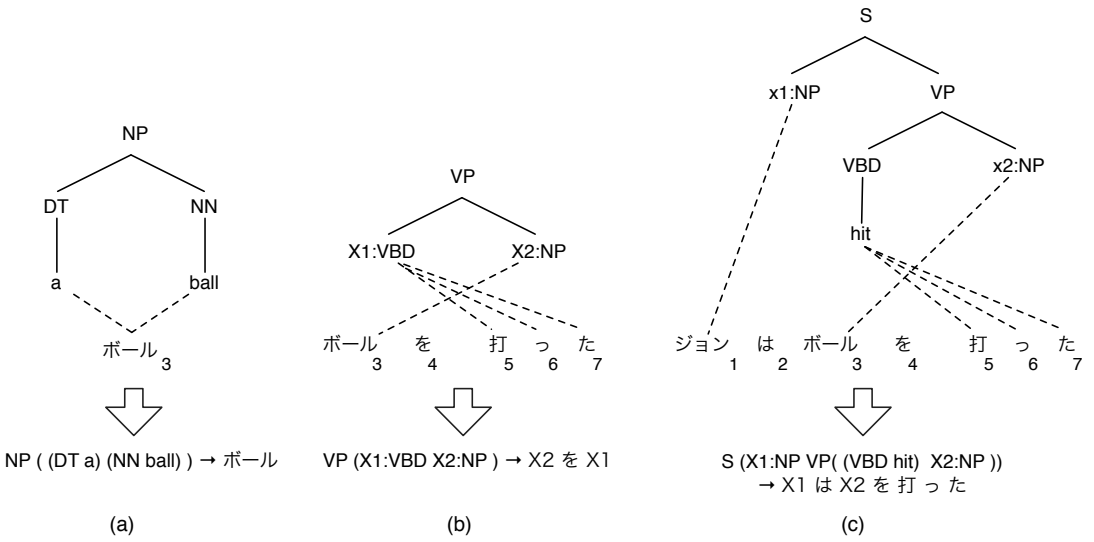

図 2.6　同期ルールの例

alignment span) と呼び，v および v が被覆する子孫の頂点以外の頂点のアライメントスパンの和集合を表す．図 2.5(a) はアライメントスパンと補完アライメントスパンの例である．構文木の各非終端ノードに “アライメントスパン/補完アライメントスパン” を示している．例えば，頂点 VP の子ノードの NP に着目すると，終端記号の “ball” が目的言語側の「ボール」に対応しているため，アライメントスパンは{3}であり，補完アライメントスパンは{1,5,6,7}となる．このようなアライメントスパンと補完アライメントスパンは，Inside-Outside アルゴリズム [19] を用いて求めることができる．

これらのスパンを利用して，単語アライメントが閉じている原言語の終端記号と目的言語側の単語の対応を求める．あるアライメントスパン $\pi(v)$ の上位集合で，かつ連続したスパンで最も短いものをアライメントスパンの閉包と呼ぶ．また，アライメントスパンの閉包と補完アライメントスパンの積集合が空集合で，かつアライメントスパンが空集合でない頂点を許容可能な頂点と呼ぶ．図 2.5(a) のアライメントスパンに基づいて許容可能な頂点を求めると図 2.5(b) のようになる．VP の子ノードの NP は許容可能であるが，その子ノードである DT や NN はアライメントスパンの閉包{3}と補完アライメントスパン{1,3,5,6,7}の積集合が空集合にならないことから，許容可能な頂点にならない．

図 2.5 の例から抽出可能な同期木置換ルールを図 2.6 に示す．図 2.6(a),(b) のように，原言語側に許容可能な頂点を内部ノードとして持たないルールを最小ルールと呼ぶ．さらに，図 2.6(c) 得られた最小ルールを組み合わせて獲得したルールを組み合わせルールと呼ぶ．最小ルールを抽出し，それらを組み合わせて多様な同期木置換ルールを得る手法としては GHKM アルゴリズム [20, 21] がある．

2.3.2 フレーズベース翻訳との比較

T2S を PBMT と比較した場合，特徴として以下のようなものが挙げられる．

1. 高い翻訳精度
2. 高速な訳出
3. 翻訳モデルのスパース性

特徴 1 に関して，日英間など語順が大きく異なる言語間では，並べ替えモデルに制限のある PBMT よりも翻訳精度が高いことが報告されている [22]．T2S は「語彙選択」と「並べ替え」を同時に行う翻訳パターンを学習することで文法構造が大きく異なる言語間であっても高い翻訳精度を実現できる．一方で，構文解析の結果を基に翻訳パターンを生成するため，翻訳精度が構文解析器の精度に依存してしまうというデメリットもある．

特徴 2 に関して，原言語文の部分木を用いて翻訳を行うため，訳出候補が少なくなるメリットがあり，探索空間が小さくなることで翻訳に必要な時間も短縮される．

最後に特徴 3 に挙げた翻訳モデルのスパース性について述べる．表 2.1 は英語の特許文を，PBMT と T2S により日本語へ実際に翻訳した例である．表中の “relinquish” に着目すると，PBMT は「放棄」と正しく翻訳できているのに対して，T2S はそのまま未知語として訳出している．この原因として，学習データにおいて出現した “relinquish” に付与された品詞と，入力文の “relinguish” に付与された品詞が異なっており，一致する翻訳パターンが存在しないと判断されたためと考えられる．このように，T2S は構文情報を用いるため，PBMT よりも翻訳パターンがスパースになる．十分な量の学習データが利用できない場合，翻訳モデルのスパース性は大きな問題になる．

表 2.1　PBMT と T2S の訳出比較

Source	the node 2 is in this master right relinquish request packet receiving status
Reference	ノード 2 が マスタ 権利 放棄 要求 パケット受信 ステータス で あ る こと
PBMT	ノード 2 は この マスター 右 放棄 要求 パケット受信 状態
T2S	ノード 2 は この マスター 権 relinquish 要求 パケット 受信 状態

2.4　機械翻訳の自動評価尺度

機械翻訳システムの訳出がどの程度良いか判断する評価方法として，主観評価と自動評価がある．主観評価は翻訳の正しさを人間が評価する方法てあり，原言語文またはその正しい参照訳に対して翻訳結果がどの程度意味を保持しているかを表す「忠実性」と，翻訳結果がどの程度流暢かを表す「流暢性」が評価項目として用いられる．しかし，主観評価は多くの時間とコストを要するため，様々な自動評価尺度が提案されている．自動評価の基本的な考え方は，機械翻訳の出力と人間が作成した参照訳を比較し，どれだけ近いか計算するというものである．本節では，本研究で使用した自動評価尺度である BLEU[23] と RIBES[24] について説明する．

2.4.1　BLEU

機械翻訳の評価において, 最も広く用いられる評価尺度が BLEU(bilingual evaluation understudy) である．BLEU は機械翻訳の出力における n-gram が, 参照訳とマッチする割合に基づいて計算される．N 文から成る機械翻訳 $E = \{\boldsymbol{e}^{(1)}, \cdots, \boldsymbol{e}^{(N)}\}$ と，それぞれの文に対応する M 文の参照訳 $R = \left\{\{\boldsymbol{r}_1^{(1)}, \cdots, \boldsymbol{r}_M^{(1)}\}, \cdots, \{\boldsymbol{r}_1^{(N)}, \cdots, \boldsymbol{r}_M^{(N)}\}\right\}$ が与えられたとき, $\boldsymbol{e}$ の n-gram の数を $C_n(\boldsymbol{e})$, そのうち $\boldsymbol{r}$ とマッチした数を $m_n(\boldsymbol{r}, \boldsymbol{e})$ とすると，BLEU 値は以下の式 (2.41) のように計算できる．

$$\mathrm{BLEU}(R,E) = \prod_{n=1}^{4} \left\{ \frac{\sum_{i=1}^{N} m_n(\{\boldsymbol{r}_1^{(i)}, \cdots, \boldsymbol{r}_M^{(i)}\}, \boldsymbol{e}^{(i)})}{\sum c_n(\boldsymbol{e}^{(i)})} \right\}^{1/4} \mathrm{BP}(R,E) \tag{2.41}$$

BLEU は一般的に 4-gram までの適合率を用い，その幾何平均を計算する．さらに，システムが参照訳に対して短い文を出力した場合にスコアが不当に高くなることを避けるため，再現率の役割を果たす簡潔ペナルティ (brevity penalty) と呼ばれる値 BP を用いる．

$$\mathrm{BP}(R,E) = \min\left\{1, \exp\left(1 - \frac{\sum_{i=1}^{N} |\tilde{\boldsymbol{r}}^{(i)}|}{\sum_{i=1}^{N} |\boldsymbol{e}^{(i)}|}\right)\right\} \tag{2.42}$$

ここで，$\tilde{\boldsymbol{r}}^{(i)}$ は，M 文の参照訳の中で $\boldsymbol{e}^{(i)}$ と最も長さが近く，かつ短い参照訳を選択する．

2.4.2 RIBES

RIBES(rank-based intuitive bilingual evaluation score) は，並べ替えに着目した評価尺度であり，日英など語順が大きく異なる言語対の評価を行うために提案されたものである．RIBES は，機械翻訳 $\boldsymbol{e}$ と参照訳 $\boldsymbol{r}$ の単語アライメントを用いて，対応する単語の並べ替えを順位として捉え，順位相関係数 (rank correlation coefficient) を求めることで計算される．$\boldsymbol{r}$ の単語列が (w_1, w_2, w_3, w_4, w_5)，$\boldsymbol{e}$ の単語列が (w_1, w_5, w_2, w_3) である場合，e が r の各単語に対応付けられる位置は (1,5,2,3) であるため，その順位は (1,4,2,3) となる．$\boldsymbol{h}$ を順位ベクトルとし，ケンドールの τ を順位相関係数として用いる場合，

$$\tau(\boldsymbol{h}) = 2\frac{\sum_{k=1}^{|\boldsymbol{h}|-1} \sum_{k'=k+1}^{|\boldsymbol{h}|} \delta(h_k < h_{k'})}{\binom{|\boldsymbol{h}|}{2}} - 1 \tag{2.43}$$

のように，すべての順序ペアのうち参照訳と一致する順序の数に基づいてペナルティを計算する．ここで $\delta(\cdot)$ は，$(\cdot)$ の条件が成立する場合に 1 を返す関数である．

RIBES の値は，対応付けられた単語の 1-gram 適合率により重み付けされた順位相関係数として表現され，以下の式 (2.44) で計算される．

$$\mathrm{RIBES}(\boldsymbol{r},\boldsymbol{e}) = \mathrm{KT}(\boldsymbol{r},\boldsymbol{e}) \cdot \left(\frac{|\boldsymbol{h}(\boldsymbol{r},\boldsymbol{e})|}{|\boldsymbol{e}|}\right)^{\alpha} \cdot \mathrm{BP}(\boldsymbol{r},\boldsymbol{e})^{\beta} \tag{2.44}$$

ここで KT(·) は式 (2.45) のような順位相関係数が [0,1] の値をとるように正規化するための関数，BP(·) は簡潔ペナルティである．また，α, β はパラメータであり，一般的に $\alpha = 0.25, \beta = 0.1$ を使用する．

$$\mathrm{KT}(\boldsymbol{r},\boldsymbol{e}) = \frac{\tau(\boldsymbol{h}(\boldsymbol{r},\boldsymbol{e})) + 1}{2} \tag{2.45}$$

2.5 機械翻訳のまとめ

2 章では，統計的機械翻訳で用いられる統計モデルについて述べ (2.1 節)，広く用いられているフレーズベース機械翻訳と構文情報を用いて翻訳を行う統語ベース翻訳について説明した (2.2 節，2.3 節)．さらに，機械翻訳の性能の確かめる上で重要となる自動評価尺度についても述べた (2.4 節)．

第 3 章 統語的前処理

SMT は統計モデルに基づいて二言語の対訳データから機械翻訳システムを自動的に構築することができ，Web の発達により大量の対訳データが利用できるようになった現在，盛んに研究されるようになった．しかし，2 章で述べた統計モデルの学習方法では適切な翻訳パターンを獲得できない場面も多く，それを補うための統語的前処理が提案されてきた．本章では，PBMT および統語ベース翻訳に対する統語的前処理の先行研究について説明し，その効果や問題点について述べる．

3.1 PBMT における統語的前処理

2.2 節で述べたように，PBMT の並べ替えモデルは長距離の並べ替え確率を正確に推定することが困難であり，英語と日本語のように語順が大きく異なる言語対では翻訳精度が低下してしまう．PBMT における統語的前処理としては，このような並べ替えモデルの貧弱さを補うための処理である事前並べ替え [4] が，盛んに研究されている．

事前並べ替えでは，構文情報を利用して原言語文を目的言語に近い語順に並べ替えてから PBMT による翻訳を行う．例えば，英日翻訳において “John hit a ball” が入力として与えられたとする．この英語文を翻訳する場合，“ジョンはボールを打った．” が正しい翻訳例として考えられる．事前並べ替えの手法では，並べ替えモデルによる長距離の語の移動の推定を避けるため，“John a ball hit” のように日本語の語順に近い英語文を結果として出力する．事前並べ替えの手法は，人手により作成された並べ替え規則を用いる手法 [25, 26] や，並べ替え規則をコーパスから自動的に学習する手法 [27, 28] がある．原言語文を目的言語の語順に並べ替えた対訳データを用いて PBMT を構築し，事前並べ替えを適用した入力文を目的言語に翻訳することで，精度が向上する．

並べ替え以外にも，翻訳パターンの獲得方法に起因した問題点も存在する．SMT の手法における翻訳パターンは，IBM モデルや HMM などを利用して得られる 2 言語間の単語アライメントに基づいてヒューリスティックにより獲得される．そのため，2 言語間の形態的・統語的な乖離が存在し，単語対応が存在しない場合は良い

翻訳パターンを獲得することが困難になる．例えば，目的言語側で原言語側に存在しない機能語が用いられることにより，単語対応の精度が低下し，その単語に関する翻訳性能も低下していまう．これに対し，ルールを用いた統語的前処理により，原言語側に対応する擬似的な文字列を挿入し，翻訳モデルを学習することによる改善が提案されている [29]．また同様に，主語の省略など，空範疇 (Empty Category, EC) の生起が見られる場合も，2 言語間で対応する単語が存在しないため上手く訳出を行うことができない [30]．この問題に対して Kudo らは，アノテーション済みのコーパスから日英機械翻訳における日本語側の主語の省略を推定可能なモデルを構築し，主語に相当する擬似文字列を日本語文に挿入して翻訳モデルの学習データを行うことで PBMT の翻訳精度を向上させている [31].

Head Finalization [5] は英日翻訳におけるこれら 2 つの問題点を緩和するための処理を含む統語的前処理であり，PBMT の精度を大幅に改善させることが知られている．Head Finalization は二言語間の統語的な構造に基づくシンプルな並べ替え処理と，より日本語に近い文とするための文字列の挿入，削除といった語順の操作とは関係のない単語に関する処理からなる．以下，この 2 つの処理について説明する．

3.1.1 Head Finalization における並べ替え処理

英語文に対する Head Finalization の適用例を図 3.1 に示す．Head Finalization の並べ替え処理は，日本語の主辞が，主辞を修飾する語の後方に置かれるという統語的な特徴を利用したものであり，英語の構文木の各非終端ノードにおいて，その子ノードの中の主辞を末尾に移動させるというものである．図 3.1 の構文木では，主辞となっている要素への枝を太線で示しており，この要素が子ノードの先頭であるときに入れ替えを行う．例えば，動詞句 VP(黒色のノード) の主辞である VBD を末尾に移動させることで，“John hit a ball”(Original English) が “John a ball hit” のように日本語語順の英語 (Head Final English) に変換される．

3.1.2 Head Finalization における単語に関する処理

Head Finalization では，語の並べ替えと関係のない以下の 3 つの処理を行う．

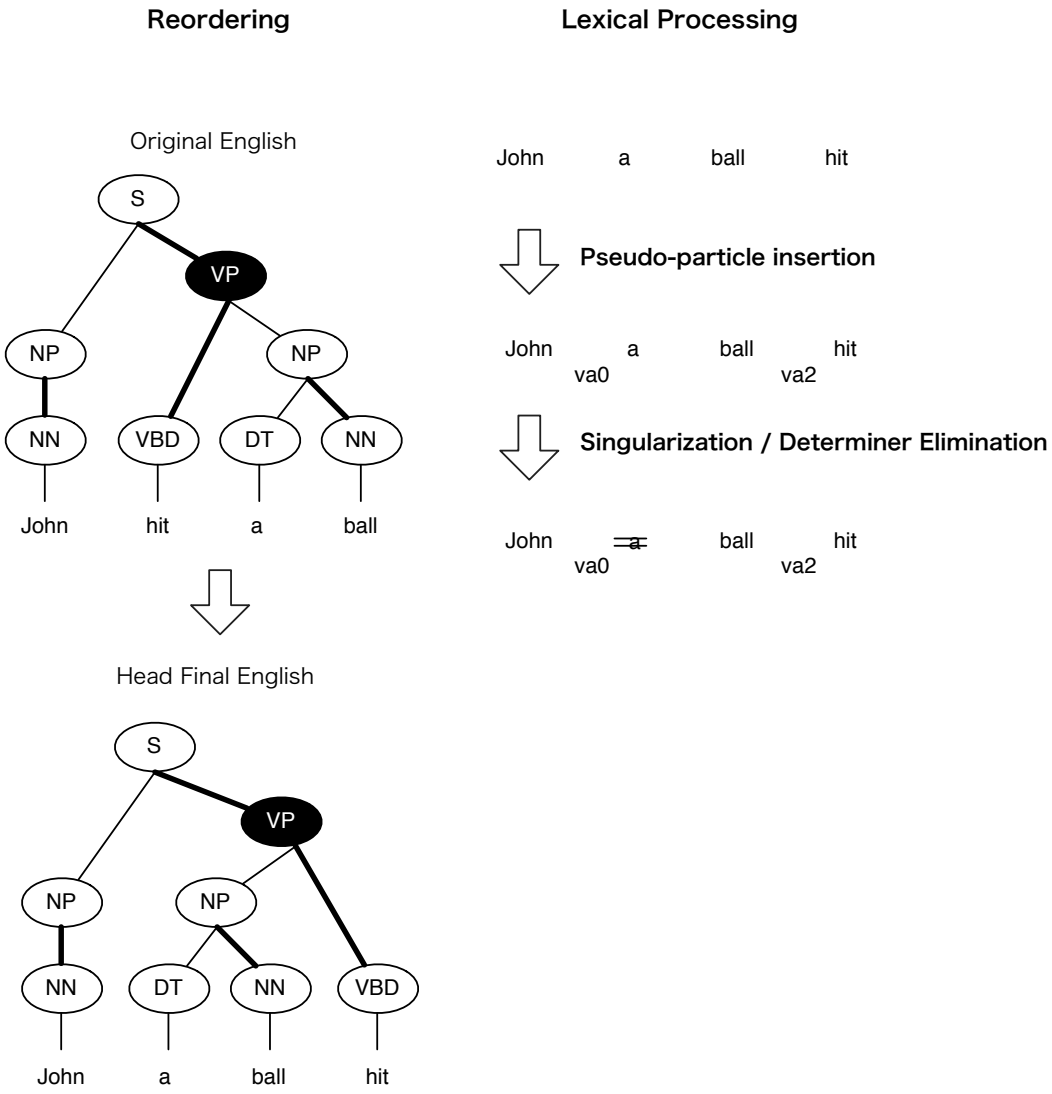

図 3.1　Head Finalization の適用例

1. 擬似助詞の挿入
2. 冠詞 “a”，“an”，“the” の削除
3. 単数化

1 の擬似助詞の挿入は，英語文にない助詞を補うことで，翻訳時の助詞の欠如や誤訳を防ぐことを目的としている．具体的には，英語文の述語項構造を利用し，動

詞の意味的主語の後ろに日本語の格助詞「が」「は」，目的語の後ろに日本語の格助詞「を」に相当する以下の 3 つの擬似助詞を挿入する．

- va0: 文の主辞動詞の主格助詞
- va1: その他の動詞の主格助詞
- va2: 動詞の目的格助詞

図 3.1 の例では，動詞 “hit” の主語となる “John” の後ろに主格を付与する va0，目的語となる “ball” の後ろに目的格を付与する va2 を挿入する．

さらに，日本語には冠詞が存在せず名詞の語尾変化による単数と複数の区別がないため，冠詞の削除，単数化の処理を行い，より日本語らしい文を生成することで，単語対応を取りやすくする．これらの処理をすべて適用することで，“John va0(wa) ball va2(wo) hit” という日本語に近い文が得られる．

3.2 統語ベース翻訳における統語的前処理

統語ベース翻訳に対しては，翻訳パターンの獲得方法に起因した問題を解決するための統語的前処理手法が多く提案されている．Burkett らは，構文木の変換によって 2 言語間の文の構成要素の対応を改善し，翻訳パターン生成時に用いる最小ルールを多く獲得することで中英翻訳における String-to-Tree 翻訳の精度を改善している [7]．また，EC の生起によって学習データにおける単語対応の精度が低下する問題も PBMT と同様に指摘されている．これに対して，Xiang らは，人手によって EC が補完された構文木を用いてそれを予測するモデルを構築し，学習データの構文木に文字列として挿入することで，中英翻訳における T2S の精度を向上させている [32].

3.3 統語的前処理に関するまとめ

本章では，PBMT に対する統語的前処理として，Head Finalization の構成要素である並べ替え処理と単語に関する処理を中心に説明した．これらはそれぞれ，PBMT の並べ替えモデルにおける問題，および 2 言語間の形態的・統語的な乖離が

存在することによる翻訳性能の低下を緩和するための処理である．

統語ベース翻訳においても PBMT と同様に統語的前処理の有効性は示唆されている．しかし，PBMT におけるルールに基づく並べ替え処理や，単語に関する前処理の適用については適用した事例がなく，検証する余地がある．また，先行研究において言語的な知見をモデル化して前処理に取り入れる際には，人手によりアノテーションされたデータを必要としている．統語的前処理に言語的な知見を用いることは有効であるが，学習データの作成に多大な労力を必要とする点は改善すべきである．

第 4 章 Tree-to-String 翻訳における統語的前処理の提案

本章では，提案法である統語ベース翻訳における統語的前処理について説明する．本研究では，統語ベース翻訳の中でも，原言語側の構文情報を用いる T2S に対して統語的前処理を適用する．4.1 節でルールに基づく統語的前処理について説明し，4.2 節でルールや人手によるアノテーション済みのデータを必要としない対訳データを用いた統語的前処理について述べる．

4.1 ルールに基づく統語的前処理の適用

3.1 節で述べたように，Head Finalization では英語文の並べ替え以外にも，擬似助詞の挿入，冠詞の削除，単数化といった翻訳精度向上のための処理も行っている．これらの処理は，挿入あるいは削除した語に対応する語の翻訳性能を向上させるものであり，PBMT 以外の翻訳手法においてもその効果が期待できる．

本論文では，Head Finalization を T2S の前処理として適用した場合，翻訳パターンの素性として追加した場合の 2 つの手法の効果を確かめる．PBMT に対して適用する場合と同様に，表 4.1 に示す 2 つの処理を T2S の前処理として適用する．以下，これらの処理の適用方法と期待される効果について詳細に述べる．

表 4.1 T2S に適用する統語的前処理

処理名	処理内容
Reordering	日本語の主辞後置性に基づく並べ替え
Lexical Processing	擬似助詞の挿入，冠詞の削除，単数化

4.1.1 T2S における並べ替え処理

T2S における並べ替え処理 (Reordering) は図 3.1 に示したように，日本語の主辞後置性に基づいて，英語文を日本語の語順に変換する処理である．PBMT における前処理は，原言語文を目的言語の語順に近づけることで，並べ替えモデルにおける

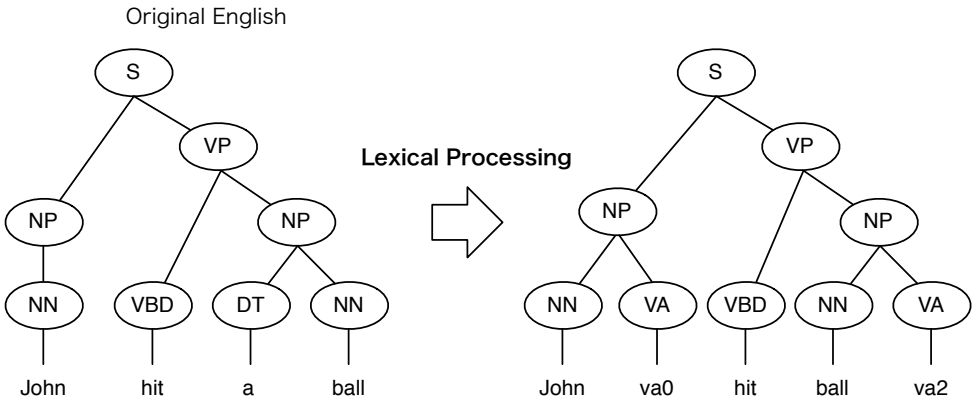

図 4.1　Lexical Processing の適用例

長距離の語順移動問題を緩和する目的で用いられる．一方で，T2S における翻訳パターンは原言語文の部分木の構造を用いて表現されるため，両言語の語順が大きく異なる場合でも並べ替えの問題は比較的少なく，PBMT の場合ほど事前並べ替えの効果は期待できない．

しかし，事前並べ替えは翻訳時の並べ替え問題を緩和する以外にも，翻訳モデル学習時の精度向上に貢献することが考えられる．例えば，IBM Model や HMM などの単語対応を獲得する手法 [33] は語順に強く影響されているため，二言語間の語順が近づくことによるアライメント精度の改善が期待できる．この要素も翻訳精度の向上に寄与するため，前処理として適用する．

4.1.2　T2S における単語の処理

T2S における単語の処理 (Lexical Processing) は，Head Finalization における擬似助詞の挿入，冠詞の削除，単数化の処理を行う．Lexical Processing の適用例を図 4.1 に示す．

擬似助詞の挿入，冠詞の削除では，簡単な構文木の変換処理を行っている．擬似助詞の挿入では，3.1.2 節に示した 3 種類の擬似助詞 “va0”，“va1”，“va2” を終端ノードとして，非終端ノード VA とともに追加する．図 4.1 の例では，動詞 “hit” の主語となる NP ノードの末尾に主格を付与する “va0”，目的語となる NP ノード

の末尾に目的格を付与する “va2” を挿入する．冠詞の削除では，終端ノード “a”, “an”, “the” とともに DT ノードを削除する．例の場合，“a” を DT ノードとともに削除する．

Lexical Processing は PBMT に適用する場合と同様に，英語文に無い助詞を補う処理や日本語に無い冠詞を削除する処理を行うことで，これらの単語に関する翻訳性能を上げることを目的としている．

4.1.3 並べ替え素性の追加

2.3 節で述べたように，T2S では語順が大きく異なる言語対を用いた場合でも，並べ替えにおける問題の影響が少ない．そのため，二言語間の語順を近づけるための前処理として Reordering を適用しても，並べ替えの精度が大幅に向上することは期待できない．そこで本研究では，並べ替えルールに従う翻訳パターンに対して素性を追加する．並べ替えに用いた統語的な特徴は，T2S の翻訳モデルにおいて，精度の向上に役立つ翻訳パターンを判別する指標として利用できる可能性がある．

本研究では，式 (2.19) の対数線形モデルを用いて Head Finalization の並べ替えルールに従う翻訳パターンに対して新たなバイナリ素性 HF-feature を追加する．Head Finalization の並べ替えルールは，日本語の主辞後置性に基づいて英語文を日本語の語順に近づけるため，翻訳精度向上のための指標になり得る．この並べ替えルールに従う翻訳パターンに対して素性を与える事により，学習された T2S の翻訳パターンの中で，二言語の統語的な特徴を考慮したものとそうでないものを区別することが可能になる．

図 4.2 に HF-feature の追加方法を示す．HF-feature を追加するために，目的言語側の単語列と原言語側の終端ノードの単語アライメントが取れた翻訳パターンを利用する．まず，原言語側の部分木に対して Head Finalization の並べ替え処理を適用する．次に，単語アライメントの交差を確認することで，並べ替え処理が適用された原言語側の単語の並びが目的言語側と一致するかどうか調べる．単語アライメントが交差しない場合は目的言語側の単語列が head final であることを表すため，翻訳パターンに対してバイナリ素性 $h_{hf}(\boldsymbol{f},\boldsymbol{e}) = 1$ を追加する．単語アライメントが交差する場合は $h_{hf}(\boldsymbol{f},\boldsymbol{e}) = 0$ とする．なお，HF-feature は目的言語側が 2 単語以

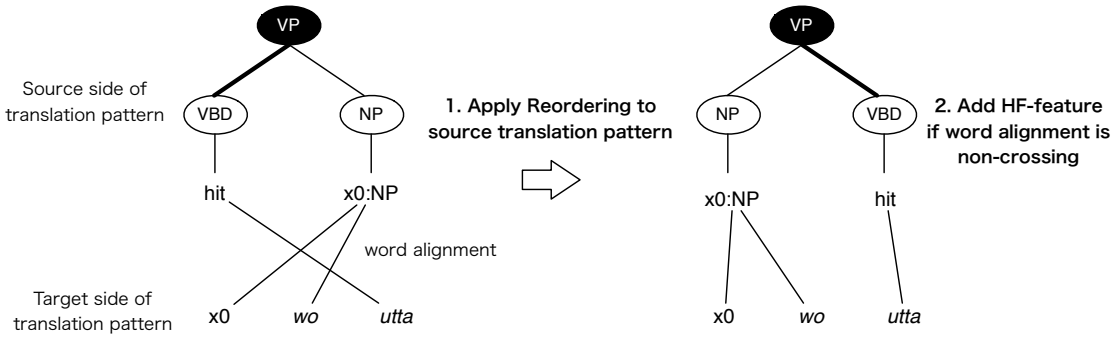

図 4.2　HF-feature の追加方法

上の翻訳パターンにのみ適用する．

4.2　対訳データを利用した統語的前処理

3.1 節や 3.2 節で述べたように，PBMT や統語ベース翻訳の学習に用いる対訳データにおいて，言語的な知見を用いて 2 言語間の形態的・統語的な乖離を埋める統語的前処理が翻訳精度を向上させる上で効果的であることが報告されている．しかし，このような知見を用いる場合は，多言語対への対応が困難となり，処理をモデル化する場合も人手によるアノテーション済みのデータが必要となる．そこで本研究では，SMT の学習に用いる対訳データとその単語アライメントに対して，目的言語側の情報を原言語側にアノテーションし，それをモデル化する手法を提案する．対訳データを利用した統語的前処理の流れは次の通りである．

1. 原言語側の構文木 T_f，目的言語側の構文木 T_e および単語アライメント $\boldsymbol{a}$ を用いて目的言語情報が付与された原言語側の構文木 $T_{f \leftarrow e}$ を生成する．
2. $T_{f \leftarrow e}$ を用いて未知のデータに対してアノテーションを行うためのモデル M を学習する．
3. M を用いて原言語文 $\boldsymbol{f}$ に対するアノテーション済みの構文木 $\hat{T}_{f \leftarrow e}$ を生成する．
4. $\hat{T}_{f \leftarrow e}$ および目的言語文 $\boldsymbol{e}$ を用いて T2S 翻訳システムを構築する．

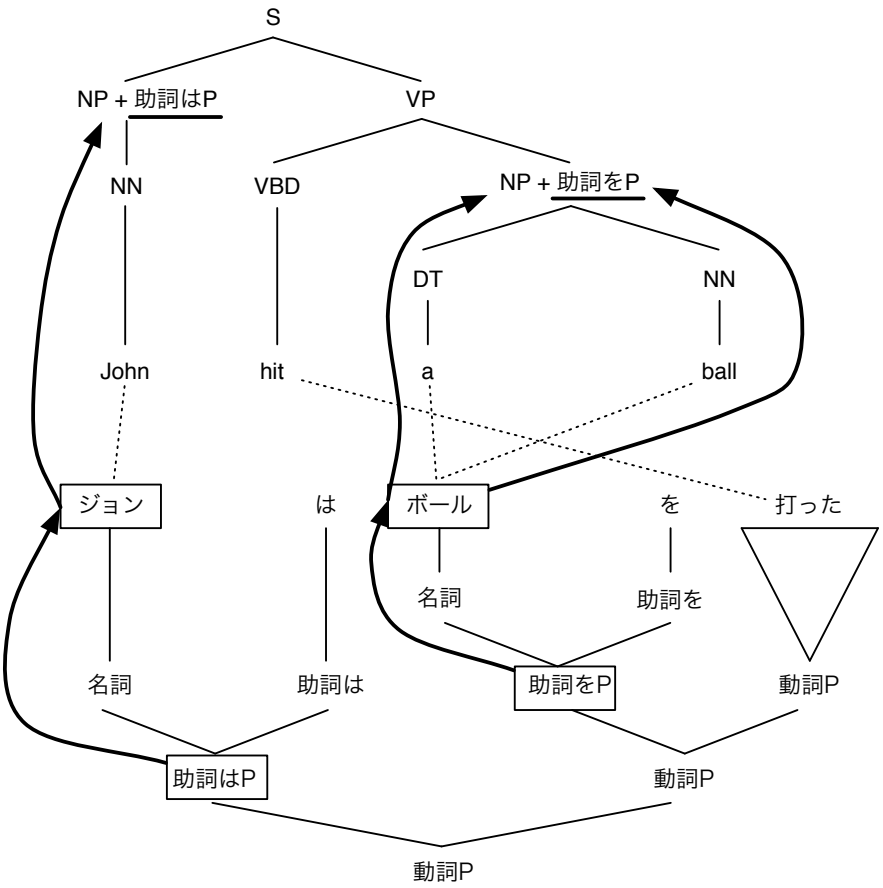

図 4.3　対訳データを用いたアノテーション

本研究では，英日翻訳を対象とし，目的言語側の情報として名詞の表層格の情報を原言語側に付与する．以下，原言語側の構文木に対する目的言語情報のアノテーション方法，およびモデルの学習方法について述べる．

4.2.1　原言語側の構文木に対する目的言語情報のアノテーション

本説では，アノテーションの手順について図 4.3 を用いて説明する．以下の手順により，英語側の構文木 T_f 対して日本語の構文木 T_e における表層格の情報を付与することで，$T_{f \leftarrow e}$ を生成する．

1. 日本語文の構文木において，名詞の親ノードをたどり，表層格の情報である助詞ラベルを抽出する．
2. 単語アライメントを介して助詞ラベルを英語側の構文木に付与する．図 4.3 に示すように，終端ノードから親ノードを順に辿り，最初の NP ノードを (NP+ 助詞ラベル) の形式に変更する．
3. 変更された NP ノードの親ノードのラベルが NP である場合，その親ノードに対しても同様の変更を行う．

これらの 1~3 の処理により，図 4.3 上部に示した日本語側の表層格の情報が付与された英語の構文木が生成される．4.1 節の Lexical Processing においても，同様の構文木を生成しているが，助詞を挿入する位置を日本語の文法知識に基いて決定しており，他の言語対に対応することが困難である．一方で，対訳データを用いた手法においては上述した 2 の処理により，日本語側の表層格の情報を文法知識を利用せずに単語アライメントを介して英語側の名詞句に付与している．このような処理により，二言語の対訳データから自動的に目的言語側の情報を原言語側に付与することが可能となり，アノテーション済みの構文木を生成するためのモデルを学習することで，より多くの言語対で用いることが可能である．

4.2.2 アノテーション済み構文木の生成

本節では 4.2.1 節で説明したアノテーションの方法により生成された $T_{f\leftarrow e}$ を用いてモデルを学習し，それを未知の原言語文 $\boldsymbol{f}$ に対して適用する方法について述べる．本研究では，英語文からアノテーション済みの構文木を生成するため，アノテーション済みのラベルを多値分類器により判別する手法，および $T_{f\leftarrow e}$ により構文解析器の再学習する手法の 2 種類を提案する．

多値分類器によるアノテーション済みラベルの生成

本研究において $\hat{T}_{f\leftarrow e}$ を生成するタスクは，原言語側の構文木 T_f の NP ラベルを $T_{f\leftarrow e}$ におけるアノテーション済みの NP ラベルに多値分類する問題と捉えることができる．$T_{f\leftarrow e}$ における NP ラベル t_{np} は以下の式 (4.21) のように選択する．

表 4.2　分類器の作成に用いた素性

No.	Tree Label Features
1	親ノードのラベル
2	2 つ上の親ノードのラベル
3	左の兄弟ノードのラベル
4	右の兄弟ノードのラベル
5	子ノードのラベル
6	子ノードの数
	Lexical Features
7	分類ラベルの部分木区間の先頭の単語
8	分類ラベルの部分木区間の末尾の単語
9	分類ラベルの部分木区間の直前の単語
10	分類ラベルの部分木区間の直後の単語

$$\hat{t}_{np} \simeq \underset{t_{np} \in T_{f \leftarrow e}}{\arg\max}\, \boldsymbol{w}^T \phi(t_{np}, T_f) \tag{4.21}$$

式 (4.21) において $\phi(\cdot)$ は素性関数であり，$\boldsymbol{w}$ はその重みベクトルである．$\hat{t}_{np}$ を適当な学習器によって学習し，未知のデータに対する予測を行うモデル M を学習する．本研究では，線型 SVM [34] により学習を行い，多クラス分類器への拡張は One-Versus-Rest 法を用いた．また，$\phi(\cdot)$ における素性として表 4.2 に示す素性を用いた．これらは，先行研究で EC の予測に用いられる構文木のラベルの素性 [32] や，構文解析に有効な単語素性 [35] を参考に設計した．

構文解析器の再学習

その他の方法として，$T_{f \leftarrow e}$ を用いて式 (4.22) のような構文解析器のモデルを再学習し，未知の原言語文 $\boldsymbol{f}$ を構文解析することで $\hat{T}_{f \leftarrow e}$ を生成することが考えられる．

$$\hat{T}_{f\leftarrow e} \simeq \arg\max_{T_{f\leftarrow e}} P(T_{f\leftarrow e}|\boldsymbol{f}) \tag{4.22}$$

多値分類器による手法が，原言語側の構文木 T_f を基にアノテーション済みのラベルのみを推定するのに対し，構文解析器の再学習による手法では，原言語文 $\boldsymbol{f}$ を基に $T_{f\leftarrow e}$ の木構造自体も推定する．このように，他の言語の情報を含む構文木により構文解析器の再学習を行った事例としては，Goto らの研究 [36] がある．この研究では，擬似助詞を含む英語の構文木を用いて構文解析器のモデルを再学習し，日英翻訳における事後並べ替え [37] の処理に用いている．再学習された構文解析器により，他の言語の情報を含む構文木を生成できることが示唆されており，$\hat{T}_{f\leftarrow e}$ を生成するタスクに用いることも十分可能と考えられる．本研究では，PCFG-LA モデルを用いた Berkeley Parser [38] を用いて，アノテーション済みの句構造を解析するモデルを再学習した．

4.3 Tree-to-String 翻訳における統語的前処理のまとめ

本章では，Tree-to-String 翻訳における統語的前処理を提案した．4.1 節でルールに基づく統語的前処理について記述した．これにより，PBMT において有効な 2 つの前処理が T2S に対しても有効であるか調査を行う．また，ルールを翻訳モデルの素性として導入し，ソフトな制約として用いる手法についても説明した．4.3 では対訳データを用いた統語的前処理について記述した．この手法は言語的な知見に基づくルールをデータから抽出し，統語的前処理に取り入れる手法である．

第 5 章 実験的評価

本章では提案法の有効性を実験によって評価する．5.1 節でルールに基づく統語的前処理，5.2 節で対訳データを用いた統語的前処理の実験的評価を行う．

5.1 ルールに基づく統語的前処理の実験的評価

ルールに基づく統語的前処理の実験的評価では，Reordering と Lexical Processing の 2 つの前処理の組み合わせにより，PBMT の翻訳精度がどの程度向上するかを検証する．T2S に対しては 2 つの前処理と HF-feature の追加の組み合わせによる翻訳精度を調べる．

5.1.1 実験条件

実験データには NTCIR-7 特許機械翻訳テストコレクション [39] の英日翻訳データを用いた．実験データに関して，学習データ (train)，開発データ (dev)，テストデータ (test) の詳細を表 5.1 に示す．

単語アライメントを取るツールとして GIZA++ [33] [*] を用い，目的言語である日本語の言語モデルは SRILM [40] を用いて 5-gram で学習した．英語側の文に対する構文解析は Enju [41] [†] を，日本語側の単語分割には KyTea [42] [‡] を用いた．実験における翻訳精度は，BLEU [23] と RIBES [24] の 2 つの自動評価尺度を用いて測った．各素性の重みは BLEU が最大となるように MERT [14] を用いて最適化した．最適化の失敗による翻訳精度の低下を防ぐため，MERT による最適化は 3 回行い，その平均スコアを最終的な評価とした [43]．PBMT は Moses [44]，T2S は Travatar [45] [§] に実装されているものをデフォルトの設定で用いた．

[*] https://code.google.com/p/giza-pp/
[†] http://www.nactem.ac.uk/enju/index.ja.html
[‡] http://www.phontron.com/kytea/index-ja.html
[§] http://www.phontron.com/travatar/

表 5.1　NTCIR7 のデータ内訳

NTCIR	Words (En)	Words (Ja)	Sentences
train	99.0M	117M	3.08M
dev	28.6k	33.5k	0.82k
test	44.3k	52.4k	1.38k

表 5.2　3 つの処理の組み合わせによる各翻訳手法の精度

ID	HF-feature	Reordering	Lexical Processing	PBMT		T2S	
				BLEU	RIBES	BLEU	RIBES
1	-	-	-	32.11	69.06	38.94	78.48
2	-	-	+	33.16	70.19	**39.51**	**79.47**
3	-	+	-	37.62	77.56	38.44	78.48
4	-	+	+	37.77	77.71	**39.60**	**79.26**
5	+	-	-	—	—	38.74	78.33
6	+	-	+	—	—	**39.29**	**79.23**
7	+	+	-	—	—	38.48	78.44
8	+	+	+	—	—	**39.38**	**79.21**

5.1.2　翻訳精度の比較

実験結果を表 5.2 に示す．表中の太字は，危険率 5% の下でブートストラップ・リサンプリング法 [46] を用いて，最も精度の高い条件と比較した結果，統計的有意性がない数値を示している．また，“+” は各処理を行った場合，“-” は行っていない場合を表している．

PBMT の翻訳精度は，Reordering を適用した 2 つの条件下で BLEU と RIBES が最も高くなった (ID 3,4)．Reordering 適用下では，Lexical Processing の効果が確認できなかったが，Lexical Processing のみを適用した場合に BLEU と RIBES が高くなった (ID 1 vs ID 2)．

T2S の翻訳精度は前処理を適用しない状態で，PBMT の最も良い条件下よりも高い精度となった (T2S:ID 1 vs PBMT:ID 4)．BLEU，RIBES ともに Lexical Processing を適用した条件下で最も高いスコアとなった (ID 2,4,6,8)．

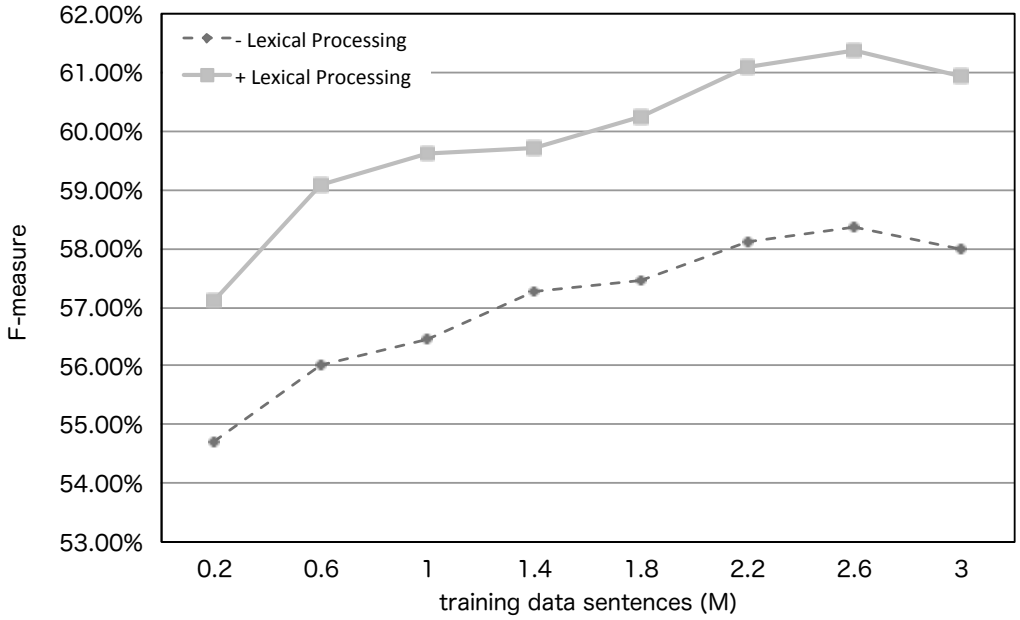

図 5.1　学習データサイズ毎の Lexical Processing による助詞の翻訳性能の改善

5.1.3　考察

事前並べ替えの先行研究で述べられているように，Reordering の処理により原言語文を目的言語の語順に近づけた場合に，PBMT の翻訳精度が大きく上昇することが確認された．また，Lexical Processing のみを追加した場合，追加していない場合と比較して BLEU スコアが向上した．なお，Lexical Processing における 3 つの処理 (助詞の挿入，冠詞の削除，単数化) について，個別に翻訳精度への影響を調べたが，冠詞の削除，単数化の処理の有無による翻訳精度への影響は確認できなかった．

Reordering の適用によって PBMT の翻訳精度は大きく改善されたが，それとは対照的に，T2S に対しては効果が見られなかった．この理由としては，T2S が並べ替えに関して比較的頑健な手法であることと，二言語間の語順が近づくことによるアライメント精度の改善に関して，期待するほどの効果が得られなかったことが挙げられる．

一方，Lexical Processing は T2S に対しても有効に機能していることが分かった．Lexical Processing を適用したすべての条件下で最も BLEU と RIBES が高くなり，

表 5.3　Lexical Processing の適用による T2S の翻訳結果の改善例

Source	another connector 96 , which is matable with this cable connector 90 , comprises a plurality of male contacts 98 aligned in a row in an electrically insulative housing 97 as shown in the figure .
Reference	この ケーブル コネクタ ９０ と 嵌合 接続され る 相手 コネクタ ９６ は 、 図示 の よう に 、 絶縁 ハウジング ９７ 内 に雄 コンタクト ９８ を 整列 保持 し て 構成 され る 。
- Lexical Processing	この ケーブル コネクタ ９０ は 相手 コネクタ ９６ は 、 図 に 示 す よう に 、 電気 絶縁 性 の ハウジング ９７に 一 列 に 並 ぶ 複数 の 雄型 コンタクト ９８ と から 構成 され て い る 。
+ Lexical Processing	この ケーブル コネクタ ９０ と 相手 コネクタ ９６ は 、 図 に 示 す よう に 、 電気 絶縁 性 の ハウジング ９７に 一 列 に 並 ぶ 複数 の 雄型 コンタクト ９８ を 有 し て 構成 され る 。

表 5.3 の訳出例のように助詞の翻訳性能の向上が確認できた．

統語的前処理による改善の効果が学習データ量を増やすことで得られる可能性を考慮し，学習データのサイズ毎に Lexical Processing の効果に差があるかどうか調査を行った．図 5.1 は学習データサイズ毎の Lexical Processing による助詞の翻訳性能の改善を示すグラフである．助詞の翻訳性能は，テスト文に含まれる “は，が，を，に” の 4 つの助詞に対する一致を F-measure により評価している．これを見ると，Lexical Processing を適用していない場合においても学習データの量を増やすことで助詞の翻訳性能が向上していることがわかる．しかし，Lexical Processing による改善幅はほぼ一定であり，この前処理による翻訳性能の向上効果は学習データを増やした場合にも得られないことがわかる．

HF-feature の追加による翻訳精度の向上は見られなかった (表 5.2:ID 1-4 vs ID

表 5.4　各条件における最適化された HF-feature の重み

HF-feature	Reordering	Lexical Processing	Weight of HF-feature
+	-	-	-0.00707078
+	-	+	0.00524676
+	+	-	0.156724
+	+	+	-0.121326

5-8)．この原因として，T2S による並べ替えの精度はすでに高く，素性の追加による改善の余地が無かったことが考えられる．表 5.4 に各条件における最適化された HF-feature の重みを示す．表 5.4 から，HF-feature の重みは 2 つの条件下で正の重み，他の 2 つの条件下で負の重みが学習されていることが分かる．head final な並べ替えを行う翻訳パターンに対して一貫性のある重みが学習されなかったため，T2S の翻訳精度に影響を及ぼす要素でなかったことが示唆される．

5.2　対訳データを用いた統語的前処理の実験的評価

対訳データを用いた統語的前処理の実験的評価では，提案法である多値分類器および再学習した構文解析器によって生成される目的言語情報付きの構文木によって T2S の翻訳精度が向上するか調査を行う．

5.2.1　実験条件

実験データには 5.1 節と同様に，NTCIR-7 特許機械翻訳テストコレクション [39] の英日翻訳データを用いた．

単語アライメントを取るツールとして Nile [47] [¶] を用い，目的言語である日本語の言語モデルは SRILM [40] を用いて 5-gram で学習した．構文解析は Ckylark [48] [‖] により行い，日本語側の単語分割には KyTea [42] [**] を用いた．実験における翻訳

[¶] https://code.google.com/p/nile/
[‖] http://odaemon.com/?page=tools_ckylark
[**] http://www.phontron.com/kytea/index-ja.html

表 5.5　テストデータにおける翻訳精度

	translation quality		particle translation quality		
	BLEU	RIBES	Precision(%)	Recall(%)	F-measure(%)
Baseline	39.73	79.16	59.11	61.90	60.47
+Annotation-SVM	39.32	79.02	58.98	62.06	60.48
+Annotation-PCFGLA	**41.05**	**80.26**	60.27	63.42	61.81
+Annotation-Oracle	**41.50**	**80.12**	**65.81**	**67.78**	**66.78**
+Self-training-PCFGLA	**41.54**	**80.58**	59.82	60.60	60.21

精度は，BLEU [23] と RIBES [24] の 2 つの自動評価尺度を用いて測った．また，助詞の翻訳性能をテスト文に含まれる “は”，“が”，“を”，“に” の 4 つの助詞に対する適合率，再現率，F-measure により評価した．各素性の重みは BLEU が最大となるように MERT [14] を用いて最適化した．T2S は Travatar [45] [††] に実装されているものをデフォルトの設定で用いた．

アノテーション済みの構文木を生成するためのモデルの学習およびテストには，それぞれ train の 9 万文，1 万文を用いた．分類器の学習には線型 SVM [34] を用い，One-Versus-Rest 法によって多値分類器への拡張を行った．構文解析器の再学習は，PCFG-LA モデルを用いた Berkeley Parser [38] により行った．

5.2.2　翻訳精度の比較

テストデータにおける翻訳精度を表 5.5 に示す．表中の太字は，危険率 5% の下でブートストラップ・リサンプリング法 [46] を用いて，ベースラインと比較した結果，統計的有意性が認められた数値を示している ($p < 0.05$)．

Annotation-Oracle はオラクルを示しており，train，dev，test の全てにおいて 4.3 節の対訳データを利用したアノテーションによる構文木を用いたものである．本タスクはアノテーションの結果を模倣することで，ベースラインに対して BLEU が 1.77 ポイント上昇し得る問題である．オラクルにおいては，助詞の翻訳性能に関して，適合率，再現率，F-measure のそれぞれの値が向上しており，アノテーションにより英語の構文木に日本語側の表層格の情報を付与する処理の効果が見られる．

[††] http://www.phontron.com/travatar/

表 5.6　オラクルに対する各手法の構文木の精度

	Precision(%)	Recall(%)	F-measure(%)
Annotation-SVM	74.93	74.93	74.93
Annotation-PCFGLA	53.66	54.49	54.07

本研究の提案手法である多値分類器によって構文木を生成した場合は，翻訳精度の向上が見られなかった (Annotation-SVM)．実験において NP ラベルの分類に用いた SVM の精度が低く (次節を参照)，アノテーション済みの構文木を正確に生成できなかったと考えられる．

構文解析器の再学習によって構文木を生成した場合は，BLEU および RIBES の値が上昇した (Annotation-PCFGLA)．しかし，助詞の翻訳性能に関して，適合率，再現率，F-measure の値が向上せず，オラクルにおける翻訳精度の改善とは異なった傾向が見られた．この原因として，構文解析器の自己学習によって T2S の翻訳精度が向上したことが考えられる [49]．ベースラインの T2S の学習に用いた英語の構文木のうち，9 万文を用いて構文解析器を再学習したところ，BLEU および RIBES 値のみが向上した (Self-training-PCFGLA)．これらの事から，構文解析器の再学習においてもアノテーション済みの構文木が正確に生成できず，自己学習による構文解析器の精度向上が翻訳精度の向上に寄与したと言える．

5.2.3　考察

実験の結果，オラクルにおける T2S の精度は大きく向上しており，4.3 節で述べた手法により，英語の構文木に日本語側の表層格の情報を付与することは有効であると考えられる．しかし，提案手法である多値分類器および再学習された構文解析器によって助詞の翻訳性能の向上は見られず，アノテーション済みの構文木を正確に生成できなかったと言える．

表 5.6 はオラクルに対する各手法の構文木の精度を示しており，以下の式 (5.21), (5.22) により適合率，再現率を計算した (テスト文: 表 5.1 の train のうち 1 万文，

表 5.7　テスト文における SVM による各ラベルの分類精度

class	Precision(%)	Recall(%)	F-measure(%)
NP+ 助詞が P	32.38 (3184/9833)	26.54 (3184/11996)	29.17
NP+ 助詞ばかり P	0.00 (0/1)	0.00 (0/3)	—
NP+ 助詞を P	41.54 (8971/21598)	45.42 (8971/19752)	43.39
NP+ 助詞も P	34.04 (16/47)	2.74 (16/583)	5.08
NP+ 助詞の P	45.08 (11831/26247)	50.45 (11831/23449)	47.61
NP+ 助詞て P	0.00 (0/1)	0.00 (0/0)	—
NP+ 助詞で P	32.67 (262/802)	6.28 (262/4174)	10.53
NP+ 助詞 P	80.00 (4/5)	1.14 (4/352)	2.24
NP+ 助詞ぐらい P	0.00 (0/29)	0.00 (0/0)	—
NP+ 助詞へ P	9.71 (80/824)	11.53 (80/694)	10.54
NP	98.37 (181/184)	79.74 (181/227)	88.08
NP+ 助詞だけ P	4.90 (39/796)	27.08 (39/144)	8.30
NP+ 助詞は P	43.48 (4739/10899)	44.65 (4739/10613)	44.06
NP+ 助詞など P	6.27 (221/3524)	34.75 (221/636)	10.63
NP+ 助詞まで P	6.14 (54/880)	19.35 (54/279)	9.32
NP+ 助詞くらい P	0.00 (0/114)	0.00 (0/0)	—
NP+ 助詞や P	11.25 (27/240)	5.37 (27/503)	7.27
NP+ 助詞か P	0.00 (0/4)	0.00 (0/116)	—
NP+ 助詞より P	45.09 (193/428)	36.07 (193/535)	40.08
NP+ 助詞と P	44.28 (1634/3690)	15.52 (1634/10530)	22.98
NP+ 助詞ほど P	0.00 (0/318)	0.00 (0/10)	—
NP+ 助詞から P	51.23 (1123/2192)	39.81 (1123/2821)	44.80
NP+ 助詞に P	44.30 (12445/28095)	53.33 (12445/23334)	48.40

Evalb[‡‡]により算出).

[‡‡] http://nlp.cs.nyu.edu/evalb/

$$\text{Precision} = \frac{\text{出力した構文木における正しい構成素の数}}{\text{出力した構文木における構成素の数}} \tag{5.21}$$

$$\text{Recall} = \frac{\text{出力した構文木における正しい構成素の数}}{\text{正解木における構成素の数}} \tag{5.22}$$

これを見ると，多値分類器による構文木の F 値が 74.93(Annotation-SVM)，構文解析器の再学習による構文木の F 値が 54.07(Annotation-PCFGLA) と，ともに低い精度になっており，オラクルにおける正解木を模倣できていないことが分かる．

ラベルの分類に用いた SVM の正解率は 40.64%(45004/110751) であった．表 5.7 は，SVM による各ラベルの分類精度を示す．“NP+ 助詞も P” など，事例数の小さいラベルの分類精度が低くなっており，“NP+ 助詞は P”, “NP+ 助詞の P”, “NP+ 助詞を P” など，事例数の大きいラベルにおいても F-measure は 40~50% 程度であった．不均衡な学習データによる分類精度の低下を避けるために，数の少ない事例を学習データから取り除いた後，SVM の学習を行ったが分類精度は同程度であった．今回の実験では，英語の構文木のラベルや単語を素性として用いているがこれらの特徴量のみでは，“NP+ 助詞は P” と “NP+ 助詞が P” の差異など，日本語の表層格の情報が付与されたラベルの僅かな違いを分別できなかったと考えられる．

対訳データを用いた統語的前処理において，本研究で用いた手法では，原言語側に付与される多くの目的言語側の情報をモデル化する事が困難であった．一方，4.1 節で述べたルールベースの Lexical Processing では，原言語側に付与する情報を 3 種類の擬似助詞に限定し，翻訳モデルの学習を行うことで，それぞれの擬似文字列に対応する複数の助詞の翻訳性能を改善している．このことを踏まえ，原言語側に付与する目的言語側の情報をそれぞれの特徴に応じてクラスタリングを行い，擬似的なラベルに置き換えることで，モデル化における問題点が改善される可能性がある．

5.3 実験的評価のまとめ

本章では，まずルールに基づく統語的前処理の効果について示した．T2S に対しては特に単語に関する処理が有効であり，ベースラインと比較した結果高い翻訳精度を示した．一方，T2S は並べ替えの精度が高く，原言語文を目的言語側に近い語

順に並べ替える前処理の効果は見られなかった．

次に対訳データを用いた統語的前処理について評価した．実験によってオラクルの翻訳精度が向上していることが確認され，アノテーションの結果を模倣するモデルを構築することで，T2S の精度改善を図れることが分かった．アノテーション済みの構文木を生成する手法としては，目的言語側の情報が付与されたラベルを多値分類器によって生成する手法，再学習された構文解析器によって生成する手法の 2 つの手法を提案した．しかし，これらの手法によって生成された構文木は，オラクルの構文木に対する精度が低く，翻訳性能の改善が見られなかった．本研究で提案した手法では，目的言語側の情報が付与された構文木を生成することが困難であるため，今後はモデルの構築方法を工夫していく必要がある．

第 6 章 結言

6.1 本論文のまとめ

本研究の目的は，統語的前処理を用いて統語ベース翻訳の更なる精度向上を図ることであった．そのために，PBMT で既に有効性が示されている，ルールに基づく統語的前処理を T2S に適用し，どのような処理が有効であるかに着目した．さらに，このようなルールに基づく統語的前処理を多言語にも対応するため，対訳データを用いた手法についても提案した．

第 2 章では，SMT の要素技術について説明し，PBMT と統語ベース翻訳の 2 つの翻訳方式の比較について記述した．

第 3 章では，PBMT と統語ベース翻訳における統語的前処理の先行研究について説明した．統語ベース翻訳では PBMT においてよく用いられるルールに基づく統語的前処理の適用例がないことや，言語的な知見を統語的前処理に用いる際に人手によりアノテーションされたデータが必要とされる点を問題点として述べた．

第 4 章では，統語ベース翻訳の方式である T2S に対する 2 つの統語的前処理を提案した．1 つ目の手法はルールに基づく統語的前処理であり，PBMT において有効な手法を T2S に対して適用するものである．2 つ目の手法は対訳データを用いた統語的前処理であり，言語的な知見に基づくルールをデータから抽出し，統語的前処理に取り入れるものである．

第 5 章では，実験により提案法の効果を確かめた．ルールに基づく統語的前処理に関しては，英日翻訳において英語側に日本語の擬似助詞を挿入する処理による翻訳精度の向上が見られ，2 言語間の乖離を埋めるための単語に関する処理が特に有効であることが分かった．一方，対訳データを用いた統語的前処理は，翻訳精度の向上が見られなかった．本研究では，多値分類によって目的言語情報側の情報がアノテーションされたラベルを生成する手法，再学習された構文解析器によってアノテーションされた構文木を生成する手法の 2 種類を提案したが，生成される構文木の精度が低く，翻訳精度の向上には寄与しなかった．現在のモデルでは，原言語側の構文木に付与された目的言語側の情報を扱うことが困難であり，この点を改善することを今後の課題として述べた．

6.2 今後の課題

今後の課題を以下に挙げる．

1 つ目は，他の言語対におけるルールに基づく統語的前処理の開発を行うことである．本研究では T2S による英日翻訳において，英語の構文木に対応する日本語の擬似助詞を挿入するという非常に単純な統語的前処理が有効であることが分かった．T2S におけるルールに基づく統語的前処理は研究例が少なく，様々な言語対において，T2S の誤り傾向を捉えることでそれに対処するためのルールベースの前処理を開発できる可能性がある．

2 つ目は，対訳データを用いた統語的前処理において，4.2.1 節で述べたアノテーションの方法を改善することである．本研究で用いた手法では，アノテーションされるラベルの数が多く，モデル化する際に目的言語側の情報が付与されたラベルを上手く扱うことができなかった．それゆえ，原言語側に付与する目的言語側の情報をそれぞれの特徴に応じてクラスタリングを行い，擬似的なラベルに置き換え，数を少なくすることで提案手法の問題点が改善される可能性がある．また，更なる発展として，言語対に応じて適切な情報を自動的にアノテーションできる枠組みを提案し，前処理において 2 言語の統語情報を適切に扱う手法を開発することが考えられる．統語ベース翻訳において，2 言語の統語情報を用いた研究例としては，S2T 翻訳において原言語側の情報をソフトな制約として対数線形モデルに取り入れた研究 [50] などがある．しかし，英日翻訳において日本語の助詞が上手く翻訳できない問題など，2 言語間の乖離に基づく翻訳誤りに対しては，翻訳パターンに直接その情報を埋め込むことが有効である．今回提案した対訳データを用いた統語的前処理の枠組みに対して，このような情報を自動的に抽出しアノテーションする手法を適用することで，様々な言語対の翻訳精度の向上に寄与すると考えられる．

謝辞

学情報科学研究科の中村哲教授には主指導教官として，研究全般に渡り大変貴重な御助言を頂き，本論文を執筆することができました．心より感謝致します．

本学情報科学研究科の松本裕治教授には副指導教官として，研究全般に渡り貴重な御助言を頂きました．心より感謝致します．

本学情報科学研究科の戸田智基准教授には副指導教官として，研究全般に渡り貴重な御助言を頂きました．心より感謝致します．

本学情報科学研究科の Graham Neubig 助教には研究における御指導や，大変貴重な御助言を頂きました．また対外発表論文執筆や語学面においても的確なアドバイスを頂きました．心より感謝致します．

本学情報科学研究科の Sakriani Sakti 助教には研究における貴重な御助言を頂きました．心より感謝致します．

知能コミュニケーション研究室の秘書である松田真奈美様には，諸手続き等で大変お世話になりました．心より感謝致します．

知能コミュニケーション研究室の先輩諸氏には，公私に渡り大変お世話になりました．そして同期諸君には，研究面においてのアドバイスなどお世話になりました．心より感謝致します．

最後に陰ながら支えてくれた母と父に心から感謝いたします．

参考文献

[1] Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical phrase-based translation. In *North American Chapter of the Association for Computational Linguistics*, pp. 48–54, 2003.

[2] Sonja Nießen and Hermann Ney. Improving SMT quality with morpho-syntactic analysis. In *Proceedings of the 18th conference on Computational linguistics-Volume 2*, pp. 1081–1085, 2000.

[3] Eleftherios Avramidis and Philipp Koehn. Enriching morphologically poor languages for statistical machine translation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 763–770, 2008.

[4] Fei Xia and Michael McCord. Improving a statistical MT system with automatically learned rewrite patterns. In *International Conference on Computational Linguistics (COLING)*, p. 508, 2004.

[5] Hideki Isozaki, Katsuhito Sudoh, Hajime Tsukada, and Kevin Duh. Head finalization: A simple reordering rule for SOV languages. In *Proceedings of the Joint 5th Workshop on Statistical Machine Translation (WMT) and MetricsMATR*, pp. 244–251, 2010.

[6] Kenji Yamada and Kevin Knight. A syntax-based statistical translation model. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 523–530, 2001.

[7] David Burkett and Dan Klein. Transforming trees to improve syntactic convergence. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 863–872, 2012.

[8] Sergei Nirenburg. Knowledge-based machine translation. *Machine Translation*, Vol. 4, No. 1, pp. 5–24, 1989.

[9] Makoto Nagao. A framework of a mechanical translation between japanese and english by analogy principle. 1984.

[10] Peter F Brown, John Cocke, Stephen A Della Pietra, Vincent J Della Pietra, Fredrick Jelinek, John D Lafferty, Robert L Mercer, and Paul S Roossin. A statistical approach to machine translation. *Computational linguistics*, Vol. 16, No. 2,

pp. 79–85, 1990.

[11] CE Shannon. A mathematical theory of communication. technical journal. *AT & T Bell Labs*, 1948.

[12] Peter F Brown, Vincent J Della Pietra, Stephen A Della Pietra, and Robert L Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, Vol. 19, No. 2, pp. 263–311, 1993.

[13] Stanley F Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 310–318, 1996.

[14] Franz Josef Och. Minimum error rate training in statistical machine translation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 160–167, 2003.

[15] William H Press. *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge university press, 2007.

[16] Christoph Tillmann. A unigram orientation model for statistical machine translation. In *North American Chapter of the Association for Computational Linguistics*, pp. 101–104, 2004.

[17] Jonathan Graehl, Kevin Knight, and Jonathan May. Training tree transducers. *Computational Linguistics*, pp. 391–427, 2008.

[18] Yang Liu, Qun Liu, and Shouxun Lin. Tree-to-string alignment template for statistical machine translation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 609–616, 2006.

[19] Fernando Pereira and Yves Schabes. Inside-outside reestimation from partially bracketed corpora. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 128–135, 1992.

[20] Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. What's in a translation rule. Technical report, DTIC Document, 2004.

[21] Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. Scalable inference and training of context-rich syntactic translation models. In *Annual Meeting of the Association for Computa-*

tional Linguistics (ACL), pp. 961–968, 2006.

[22] Graham Neubig and Kevin Duh. On the elements of an accurate tree-to-string machine translation system. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 143–149, 2014.

[23] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 311–318, 2002.

[24] Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. Automatic evaluation of translation quality for distant language pairs. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 944–952, 2010.

[25] Sho Hoshino, Yusuke Miyao, Katsuhito Sudoh, and Masaaki Nagata. Two-stage pre-ordering for Japanese-to-English statistical machine translation. In *International Joint Conference on Natural Language Processing (IJCNLP)*, 2013.

[26] Peng Xu, Jaeho Kang, Michael Ringgaard, and Franz Och. Using a dependency parser to improve SMT for subject-object-verb languages. In *North American Chapter of the Association for Computational Linguistics*, pp. 245–253, 2009.

[27] Graham Neubig, Taro Watanabe, and Shinsuke Mori. Inducing a discriminative parser to optimize machine translation reordering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 843–853, 2012.

[28] Roy Tromble and Jason Eisner. Learning linear ordering problems for better translation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1007–1016, 2009.

[29] Gumwon Hong, Seung-Wook Lee, and Hae-Chang Rim. Bridging morpho-syntactic gap between source and target sentences for English-Korean statistical machine translation. In *International Joint Conference on Natural Language Processing (IJCNLP)*, pp. 233–236, 2009.

[30] Hirotoshi Taira, Katsuhito Sudoh, and Masaaki Nagata. Zero pronoun resolution can improve the quality of JE translation. In *Workshop on Syntax and Structure in Statistical Translation*, pp. 111–118, 2012.

[31] Taku Kudo, Hiroshi Ichikawa, and Hideto Kazawa. A joint inference of deep case analysis and zero subject generation for Japanese-to-English statistical machine translation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 557–562, 2014.

[32] Bing Xiang, Xiaoqiang Luo, and Bowen Zhou. Enlisting the ghost: Modeling empty categories for machine translation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 822–831, 2013.

[33] Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Computational linguistics*, pp. 19–51, 2003.

[34] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, Vol. 9, pp. 1871–1874, 2008.

[35] David Hall, Greg Durrett, and Dan Klein. Less grammar, more features. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 228–237, 2014.

[36] Isao Goto, Masao Utiyama, and Eiichiro Sumita. Post-ordering by parsing for Japanese-English statistical machine translation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 311–316, 2012.

[37] Michael Collins, Philipp Koehn, and Ivona Kučerová. Clause restructuring for statistical machine translation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 531–540, 2005.

[38] Slav Petrov and Dan Klein. Improved inference for unlexicalized parsing. In *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, pp. 404–411, 2007.

[39] Atsushi Fujii, Masao Utiyama, Mikio Yamamoto, Takehito Utsuro, Terumasa Ehara, Hiroshi Echizen-ya, and Sayori Shimohata. Overview of the patent translation task at the NTCIR-7 workshop. In *Proceedings of the 7th NTCIR Workshop Meeting*, pp. 389–400, 2008.

[40] Andreas Stolcke, Jing Zheng, Wen Wang, and Victor Abrash. SRILM at sixteen: Update and outlook. In *IEEE Automatic Speech Recognition and Understanding*

Workshop (ASRU), p. 5, 2011.

[41] Yusuke Miyao and Jun'ichi Tsujii. Maximum entropy estimation for feature forests. In *Proceedings of the second international conference on Human Language Technology Research*, pp. 292–297, 2002.

[42] Graham Neubig, Yosuke Nakata, and Shinsuke Mori. Pointwise prediction for robust, adaptable Japanese morphological analysis. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 529–533, 2011.

[43] Jonathan H Clark, Chris Dyer, Alon Lavie, and Noah A Smith. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 176–181, 2011.

[44] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. Moses: Open source toolkit for statistical machine translation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 177–180, 2007.

[45] Graham Neubig. Travatar: A forest-to-string machine translation engine based on tree transducers. *Annual Meeting of the Association for Computational Linguistics (ACL)*, p. 91, 2013.

[46] Philipp Koehn. Statistical significance tests for machine translation evaluation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 388–395, 2004.

[47] Jason Riesa and Daniel Marcu. Hierarchical search for word alignment. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 157–166, 2010.

[48] 小田悠介, Graham Neubig, 波多腰優斗, Sakriani Sakti, 戸田智基, 中村哲. 解析失敗の発生しにくい PCFG-LA 句構造構文解析. 言語処理学会第 21 回年次大会 (NLP2015), 2015.

[49] 波多腰優斗, Graham Neubig, Sakriani Sakti, 戸田智基, 中村哲. Tree-to-string 翻訳における構文解析器の自己学習の効果. 言語処理学会第 21 回年次大会

(NLP2015), 2015.

[50] Matthias Huck, Hieu Hoang, and Philipp Koehn. Preference grammars and soft syntactic constraints for GHKM syntax-based statistical machine translation. In *Workshop on Syntax and Structure in Statistical Translation*, 2014.

発表リスト

国際会議

[1] **Yuto Hatakoshi**, Graham Neubig, Sakriani Sakti, Tomoki Toda, Satoshi Nakamura. Rule-based Syntactic Preprocessing for Syntax-based Machine Translation, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST), 2014.

大会講演

[1] **波多腰 優斗**, Graham Neubig, Sakriani Sakti, 戸田智基, 中村 哲. Tree-to-String 翻訳における構文解析器の自己学習の効果, 言語処理学会第 21 回年次大会 (NLP2015), 2015 年.

研究会

[1] **波多腰 優斗**, Graham Neubig, Sakriani Sakti, 戸田智基, 中村 哲. 統語ベース翻訳に対する統語的前処理の適用, 情報処理学会第 217 回自然言語処理研究会 (SIG-NL), 2014 年.