事前並べ替えを利用したヒンディー語—英語統計的機械翻訳

村松航平　Kevin Duh　松本裕治
奈良先端科学技術大学院大学情報科学研究科
{muramatsu.kohei.mb7,kevinduh,matsu}@is.naist.jp

1 はじめに

日本語のようなSOV型言語と、英語のようなSVO型言語間の統計的機械翻訳ではそれぞれの言語の語順の違いが大きな問題となっている。SOV型の言語とSVO型の言語では語順が異なり、従来の統計的機械翻訳手法では対処が難しい問題とされてきた。こういった問題に対処する方法として、事前並べ替え、事後並べ替え、Tree-to-String翻訳といった手法などがある。

事前並べ替えは原言語文を目的言語の語順に並べ替えてから、通常の機械翻訳を行う手法である。並べ替えには構文解析器を用いて原言語文を構文解析し、それをもとに並べ替えルールを人手で作成する方法 [1]、構文解析器・対訳データ・アライメントを利用して、並べ替えモデルを自動で学習する方法 [2]、対訳データ・アライメントから並べ替えのための構文解析器を自動的に学習する手法 [3] などが考えられている。

2 背景

並べ替えの手法は多くの場合、構文情報を利用するため構文解析器を必要とする。そのため英語のような高精度の構文解析器を持つ言語ならば精度の高い並べ替えが期待出来るが、充実した構文解析器を持たない言語の並べ替えは困難なものである。

そうした言語の１つ、ヒンディー語はインドを中心とした南アジアで話されている言語で、話者数は世界で約５億人であり世界でも３番目に多くの人に話されている言語である。また、近年経済発展の目覚ましいインドの連邦公用語であり、ますます需要が高まっている言語の一つである。言語的な重要性が高まっているのに関わらず、ヒンディー語の言語処理ツールはいまだ十分ではない。

またヒンディー語は英語と比べてSOV型言語の言語として知られるが、その語順は図１のように比較的自由で目的語が主語の前に移動したりと、ときにOSVとなることもある。修飾語は英語と同じく被修飾語の

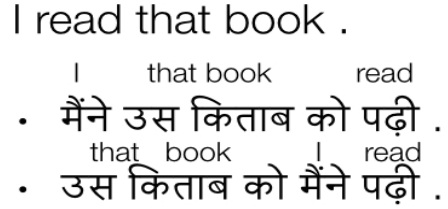

図 1: ヒンディー語の語順の例

前に置かれ、日本語と同じく前置詞ではなく後置詞が取られる。動詞・名詞の性・数による屈折が豊富であるのもヒンディー語の特徴である。また動詞に関しては、2つ以上の動詞が連なって用いられることが多い。使われるそれぞれの動詞の種類によって文全体の意味が変わってくる。

本稿では、言語処理ツールの乏しい言語としてヒンディー語をあげ、構文解析器を用いない手法で並べ替えを行い、それを応用した英語—ヒンディー語間、ヒンディー語—英語間の機械翻訳を行った。

3 関連研究

英語-ヒンディー語の事前並べ替えを利用した機械翻訳の研究としては [4, 5, 6] がある。[5, 6] は英語からヒンディー語への機械翻訳が行われており、並べ替えには構文解析器や形態素解析器を利用されている。[4] は並べ替えを巡回セールスマン問題として捉え、対訳コーパスとアライメントのみで並べ替えモデルを学習している。またこれはヒンディー語—英語、英語—ヒンディー語、ウルドゥー語—英語の機械翻訳の精度向上に成功している。[5] では英語文が構文解析され、人手で作成された並べ替えルールをもとに並べ替えがなされる。[6] では英語が構文解析され、原言語・目的言語双方の品詞タグを利用し語順関係を学習し英語文の並べ替えが行われる。

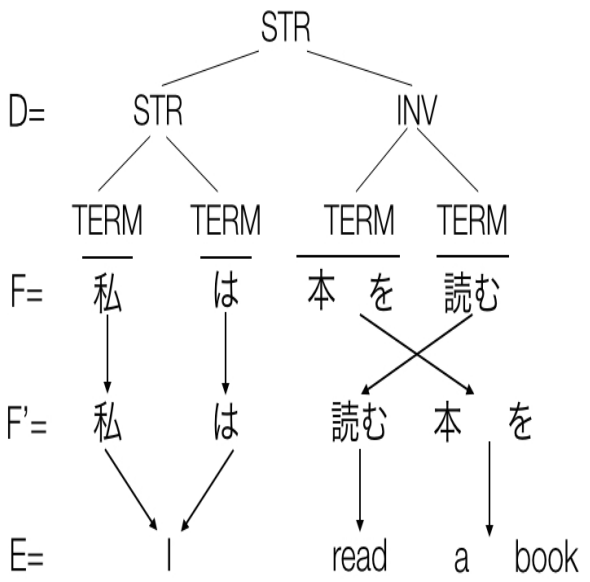

図 2: BTG による並べ替えの例

4 提案手法

ヒンディー語が十分な構文解析器を持たない言語であるので、本稿では構文解析器を用いない手法でヒンディー語の並べ替えを行い、機械翻訳に応用する。構文解析器を用いない並べ替え手法として [3] や [4] がある。[4] は対訳データとアライメントを用いて並べ替えモデルを学習するものだが、[3] は対訳データとアライメントを用いて並べ替えのための構文解析器を作成する。本論では構文解析器を用いずに並べ替えを行うことが出来る手法の中でも、未だヒンディー語での応用がなされていない [3] の手法を試みる。[3] では統計的機械翻訳の並べ替えを改善させるための教師なし識別構文解析器として lader(Latent Derivation Reorderer) が提案されている。対訳コーパスとアライメントから、並べ替えの精度が最大化するように括弧反転トランスダクション文法 (Bracketing Transduction Grammar, BTG) を教師なし学習し並べ替えを行う。

図 2 に BTG による並べ替えの例を示した。BTG の導出 D からは (STR) と (INV) のラベルが付いた非終端記号と (TERM) のラベルが付いた終端記号の直前を意味する記号が生成される。F と F' それぞれは原言語の文と、その並べ替え後の文を意味し、E は目的語の文を意味する。非終端記号に (STR) のラベルが付いた場合は終端記号はそのまま生成され、(INV) のラベルが付いた場合は終端記号は逆順になって生成される。また (TERM) は文字列 f を生成する。このように並べ替えの語順は木構造と非終端記号のラベル (STR) と (INV) によって決まる。それぞれの部分木は原言語の文字列 f とその並べ替え後の文字列 f' を示している。f の左側の子 f_1 と右側の子 f_2 に対し、もし非終端記号のラベルが (STR) ならば並べ替えられた文は $f'=f'_1f'_2$ といったように生成され、(INV) ならば $f'=f'_2f'_1$ と生成される。

BTG により生成された並べ替えのうち最も良い並べ替えを見つけるためにスコア関数が定義されている。ある入力文 F から並べ替え文 F' を導く導出 D は複数生成されると考えられる。複数生成された導出 D の中から最適な並べ替えを行う D を選択する必要がある。そこで BTG の導出 D から抽出された素性とその重み付き和をもとにスコアを計算し、そのスコアが最大となった導出 D を用いて F から F' を導く。

並べ替えられた原言語文を用いて通常のフレーズベース翻訳を行う。学習の際、構文情報や品詞タグなどの情報も素性に含めて学習を行うことも可能である。[3] は統計に基づいた手法なので並べ替えルールを人手で作成する必要が無い。そのためその言語に関する詳細な知識が無くても並べ替えをすることが出来る。また、学習に用いるアライメントの精度が並べ替えに影響するため、高精度なアライメントを作成することが精度の高い並べ替えを行うために重要となる。[3] では日本語ー英語、英語ー日本語間の機械翻訳でフレーズベース翻訳、階層的フレーズベース翻訳よりも高い精度を出すことに成功している。

5 実験

ヒンディー語ー英語、英語ーヒンディー語間の統計的機械翻訳における lader による事前並べ替えの有効性を調べるため、フレーズベース翻訳と事前並び替えを用いた翻訳を比較した。

5.1 実験データ

実験データにはヒンディー語ー英語バイリンガルコーパス (HindEnCorp)[1] とヒンディー語モノリンガルコーパス (HindMonoCorp)[2]、WMT2014 で用いられた英語モノリンガルコーパス [3] を用いる。それぞれウェブ上の様々なサイトをクロールしたテキストである。

[1] https://lindat.mff.cuni.cz/repository/xmlui/handle/11858/00-097C-0000-0023-625F-0
[2] https://lindat.mff.cuni.cz/repository/xmlui/handle/11858/00-097C-0000-0023-6260-A
[3] http://www.statmt.org/wmt14/translation-task.html

HindEnCorp から学習データに 25 万文、ディヴェロップメントデータに 1000 文、テストデータに 2500 文を用いた。さらに言語モデル作成にヒンディー語と英語、それぞれ 4 千万文と 2 千万文のモノリンガルコーパスを用いた。

5.2 ベースライン

ベースラインには通常のフレーズベース翻訳を設定した。単語アライメントの獲得には GIZA++を用い、言語モデルの作成には SRILM を用いた。また、デコーダーには Moses、チューニングには MERT を用いた。

5.3 事前並べ替え

事前並べ替えには lader を用いる。lader の学習には HindEnCorp から取り出した対訳文とそのアライメントを用いる。学習に用いる対訳文の量を 1000 文、10000 文と変化させ、精度の変化を見た。また、学習に用いたアライメントは GIZA++で取得した。

原言語側の並べ替えを行った後は、ベースラインと同じように Moses で通常のフレーズベース翻訳を行った。

システム	en-hi BLEU/RIBES	hi-en BLEU/RIBES
PBMT	18.00/63.04	18.63/63.37
PBMT+lader(1000)	18.94/63.87	18.77/63.97
PBMT+lader(10000)	19.06/64.73	19.89/64.51

表 1: lader による事前並べ替えの結果

6 実験結果

実験結果を表 1 に示す。PBMT は Moses によるフレーズベース翻訳、PBMT + lader(1000)、PBMT + lader(10000) はそれぞれ 1000 文、10000 文の対訳コーパスとアライメントで lader を学習させ並べ替えを行い、フレーズベース翻訳を行ったものである。

自動評価には BLEU と RIBES の 2 つを用いた。ヒンディー語―英語、英語―ヒンディー語翻訳どちらの場合も、ベースラインに設定した Moses のフレーズベース翻訳結果と lader での事前並べ替えを行った翻訳結果を比べると、1000 文での学習と 10000 文での学習どちらの場合でも lader で事前並べ替えを行った翻訳結果の方がベースラインを上回っていた。表 1 の結果からヒンディー語―英語、英語―ヒンディー語間における機械翻訳での lader による事前並べ替えの有効性が示されている。

今回の実験では lader に学習させる対訳文とアライメントの量を変化させ、その精度の変化を見た。英語―ヒンディー語翻訳では、1000 文の学習の場合では BLEU/RIBES ともに 0.94/0.83、10000 文の学習の場合では BLEU/RIBES それぞれ 1.06/1.69 ベースラインから向上した。ヒンディー語―英語翻訳では、1000 文の学習の場合では BLEU/RIBES ともに 0.14/0.60、10000 文の学習の場合では BLEU/RIBES それぞれ 1.26/1.14 ベースラインから向上した。英語―ヒンディー語、ヒンディー語―英語翻訳のどちらも 1000 文の学習結果よりも 10000 文での学習の結果の方が精度が高いことが示されているので、lader での学習量が多いほど翻訳精度も高くなることがわかる。

7 考察

lader によって並べ替えられた英語・ヒンディー語の例を図 3・図 4 に示す。図 3 を見ると 1000 文での学習と 10000 文での学習の場合どちらも英語文がヒンディー語の語順に並び変わっているのがわかる。

図 4 にはヒンディー語の並べ替え例を示した。1000 文での学習の場合だいたいの語順は英語の語順と一致しているが、ところどころフレーズがバラバラになっていたり、同一フレーズ内でも並び順が逆になっている箇所が見られる。10000 文での学習の場合、フレーズがバラバラになってしまう現象や並び順が逆になってしまう現象は改善されたように見える。

表 1 を見ると、ヒンディー語―英語翻訳の方が英語―ヒンディー語翻訳よりも RIBES の向上が少ないことがわかる。これは英語の並べ替えと比べてヒンディー語の並べ替えがうまくいっていないことを示している。その要因として、英語の文法は基本的に語順により格を示しており、それと比べてヒンディー語は SOV 型言語ではあるが英語のような厳しい語順制限は無く、比較的容易に語順が変化することが考えられる。

8 おわりに

本稿では言語処理ツールが十分ではない言語としてヒンディー語をあげ、ヒンディー語と語順が離れた言語である英語との機械翻訳において事前並べ替えを用

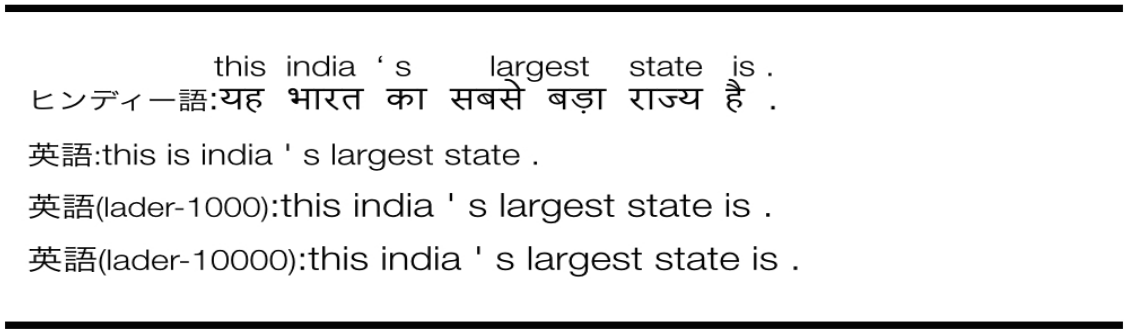

図 3: 英語の並べ替え結果の例

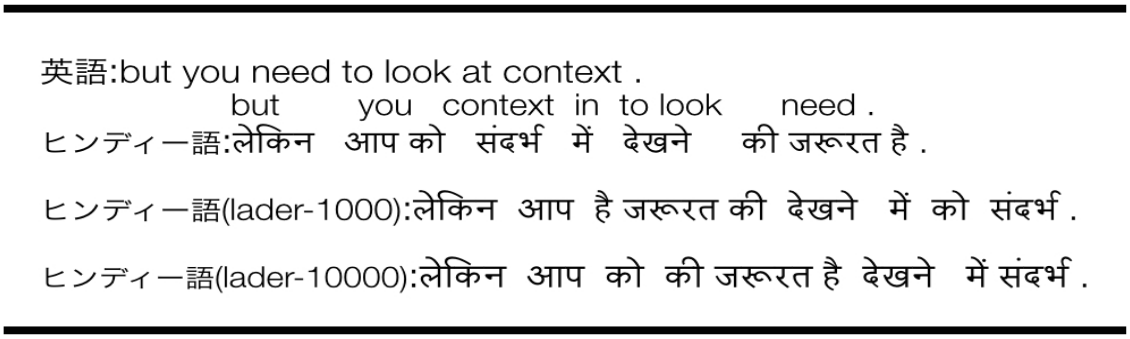

図 4: ヒンディー語の並べ替え結果の例

いた。並べ替えは対訳文とアライメントから並べ替えのための構文解析器を導出するという形で並べ替えを行い、結果として機械翻訳の精度を上げることに成功した。

lader によりヒンディー語の並べ替えに成功したが、機械翻訳の精度をさらに上げるためには語順以外の要素を考えて翻訳システムを構築する必要があるだろう。例えばヒンディー語の文法において特徴的な性・数による動詞や名詞の屈折や多数の動詞が連接した複合動詞などが解決すべき問題として考えられる。

参考文献

[1] Hideki Isozaki, Katsuhito Sudoh, Hajime Tsukada, Kevin Duh: Head Finalization: A Simple Reordering Rule for SOV Languages, in Proc. of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR, pp.244-251, 2010.

[2] Fei Xia, Michael McCord: Improving a Statistical MT System with Automatically Learned Rewrite Patterns, in Proc. of ICCL, pp.508-514, 2004.

[3] Graham Neubig, Taro Watanabe, Mori Shinsuke: Inducing a Discriminative Parser to Optimize Machine Translation Reordering, in Proc. of EMNLP-CoNLL, pp.843-853, 2012.

[4] Karthik Visweswariah, Rajakrishnan Rajkumar, Ankur Gandhe, Ananthakrishnan Ramanathan, Jiri Navratil: A Word Reordering Model for Improved Machine Translation, in Proc. of EMNLP, pp.486-496, 2011.

[5] Raj Nath Patel, Rohit Gupta, Prakash B. Pimpale and Sasikumar M: Reordering rules for English-Hindi SMT, in Proc. of the Second Workshop on Hybrid Approaches to Translation, pp.3441, 2013.

[6] Taraka Rama, Karthik Gali, Avinesh PVS: Does Syntactic Knowledge help English-Hindi SMT ?, in Proc. of the NLP Tools contest, ICON 2008.