

Recursive Neural Network を用いた事前並び替えによる 英日機械翻訳

瓦 祐希[†] Chenhui Chu[‡] 荒瀬 由紀[†]

[†] 大阪大学大学院情報科学研究科 [‡] 大阪大学データビリティフロンティア機構

{kawara, arese}@ist.osaka-u.ac.jp, chu@ids.osaka-u.ac.jp

1 はじめに

統計的機械翻訳 (SMT) において、翻訳言語間の語順の大きな違いは翻訳精度に影響を与えることが知られている [15, 9, 3]。この語順の問題に対し、デコードの際に翻訳するフレーズの順番を考慮するモデルが提案されている [15]。しかし、このような並び替えモデルは長距離の並び替えに弱く、計算コストが高い。そこで、翻訳機に入力する前に原言語文を目的言語の語順に近づくよう並び替える事前並び替え [9]、並び替えを考慮せずに翻訳した目的言語文を並び替える事後並び替えが提案されている [3]。特に事前並び替えは翻訳機の学習時に対訳フレーズ抽出の精度を向上できるため、翻訳性能の改善に効果的である [9]。

事前並び替えの手法として、人手で並び替えルールを定めて並び替えを行う手法 [5]、データから自動で並び替えルールを抽出する手法 [2] がある。構文木に対して機械学習を用いて並び替えモデルの学習を行う手法も提案されており、Hoshino ら [4] は原言語の構文木の各ノードで子ノードの順序を入れ替えるかどうかを学習する手法を提案している。Neubig ら [11] や Nakagawa [9] は原言語文から 2 分木を構築しつつ並び替えを行う手法を提案している。これらの手法では、翻訳する言語ペアに応じて特徴量を人手で設計する必要がある。近年では特徴量の設計を必要としないニューラルネットワークを用いた手法 [1] も提案されており、構文木の節ノードで並び替えするかどうかを学習させる。しかしこの手法では注目しているノードの子ノードのみ考慮して並び替えを行うかを判定しており、並び替えにおいて重要な情報を含むと考えられる部分木や部分木に対応した部分フレーズ全体を考慮していない。

本論文では Recursive Neural Network (RvNN) を用いた特徴量設計が不要な事前並び替え手法を提案する。RvNN を用い構文木上でボトムアップに計算を行

うことで部分木全体を考慮した事前並び替えを行う。具体的には、原言語の構文木¹において、単語と品詞、および構文タグのベクトル表現を用いて、構文木上の各節ノードで並び替えを行うかどうかを判定する学習を行う。フレーズベースの統計的機械翻訳 (PBSMT) とニューラル機械翻訳 (NMT) を翻訳機として用い、英日翻訳における評価実験を行なった。その結果、翻訳精度は NMT には及ばないが、PBSMT においては提案手法が特徴量設計を要する事前並び替えの state-of-the-art [9] と匹敵する精度を達成できることを確認した。

2 RvNN による事前並び替え

2.1 正解ラベルの付与

構文木の各節ノードにおいて子ノードの順序を入れ替えるかどうかのラベル付けを行い、訓練データを作成する。ラベルは式 (1) で計算される順位相関係数の一つである Kendall の τ [6] を用いて決定する。

$$\tau = \frac{4 \sum_{i=1}^{n-1} \sum_{j=i}^n \delta(\mathbf{y}_i < \mathbf{y}_j)}{n(n-1)} - 1 \quad (1)$$

$$\delta(x) = \begin{cases} 1 & (x \text{ is true}) \\ 0 & (\text{otherwise}) \end{cases} \quad (2)$$

\mathbf{y} は単語アラインメントのインデックスを並べたベクトルである。各節ノードにおいて、子ノードを入れ替えた際に Kendall の τ が大きくなる場合は並び替えを行う “Inverted” ラベルを、そうでない場合はそのままの順序を保持する “Straight” ラベルを付与する。

2.2 事前並び替えモデル

2 分木の構造を持つ構文木にしたがって RvNN を構築し、各節ノードで 2.1 節で付与したラベルを予測す

¹本論文では構文木は 2 分木を想定する。

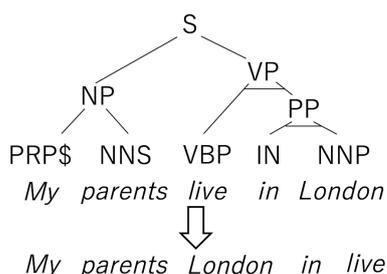


図 1: “My parents live in London” の RvNN による並び替え（横線が引いてある節ノードは “Inverted”）

る。葉ノードからボトムアップに計算を行うことで、部分フレーズを考慮しつつ並び替えるか否かを決定できる。図 1 に “My parents live in London” という文に対して RvNN を用いた並び替えの例を示す。ここで “live in London” のフレーズに対応したノードにおいて、式 (3) に従い “live” と “in London” の子ノードを考慮してベクトルを計算する。

$$\mathbf{p} = f(W[\mathbf{p}_l; \mathbf{p}_r] + \mathbf{b}) \quad (3)$$

$$\mathbf{s} = W_s \mathbf{p} + \mathbf{b} \quad (4)$$

f は ReLu 関数、 W は重み行列、 \mathbf{p}_l 、 \mathbf{p}_r はそれぞれ左、右の子ノードのベクトル、 W_s は出力層における重み行列、 \mathbf{b} はバイアス項を表す。また $[l; r]$ は l と r をこの順に結合することを表す。 $\mathbf{s} \in R^2$ は各ラベルに対する重みのベクトルであり、ソフトマックス関数に入力することで “Straight” および “Inverted” ラベルの確率を計算する。葉ノードでは、単語ベクトルを入力とし、式 (6) により得たベクトルを式 (3) に代入する。

$$\mathbf{e} = W_E \mathbf{x} \quad (5)$$

$$\mathbf{p}_e = f(W_l \mathbf{e} + \mathbf{b}) \quad (6)$$

ここで \mathbf{x} は入力単語を表す one-hot ベクトル、 W_E は単語分散表現を表す行列である。損失関数は式 (7) で定義される交差エントロピーを用いる。

$$L(w) = - \sum_{n=1}^N \sum_{k=1}^K y_{nk} \log p_j(p; w) \quad (7)$$

品詞、および構文タグを考慮するモデルでは、式 (3) に代わり式 (8) を用いる。

$$\mathbf{p} = f(W[[\mathbf{p}_l; \mathbf{p}_r]; \mathbf{e}_{\text{tag}}] + \mathbf{b}) \quad (8)$$

\mathbf{e}_{tag} は品詞・構文タグの情報を表現するベクトルで、各ノードの品詞または構文タグを表す one-hot ベクトルを入力とし、式 (5)、(6) と同様に計算する。

3 英日翻訳による評価実験

3.1 実験設定

ASPEC コーパス [10] を用いて英語から日本語への翻訳を行なった。ASPEC コーパスに含まれる訓練データは約 300 万文対、開発データは 1,790 文対、テストデータは 1,812 文対である。ASPEC コーパスは Utiyama and Isahara [16] の類似度に基づいて対訳文がランク付けされている。本研究では上位 50 万文対から 10 万文対をサンプリングして並び替えの訓練データとした。英語文は Stanford CoreNLP²で単語分割と品詞タグ付けを、enju³で構文解析を行なった。日本語文は MeCab⁴を用いて形態素解析したものを用いた。単語アラインメントには GIZA++⁵を用い、IBM model 1 と HMM を用いて両方向からの単語アラインメントを計算し、intersection のルールにより最終的な単語アラインメントを獲得した。

RvNN は Chainer⁶を用いて実装し、語彙は頻度が高いものから 5 万語を用いた。最適化には Adam[7] に重み減衰および GradientClipping を適用して行なった。ミニバッチサイズは 500 とした。Nakagawa[9] の BTG 手法を比較対象とし、公開されている実装⁷を提案手法と同一の前処理を行なった 10 万文対の ASPEC 英日データを用いて訓練し、並び替えを行なった。

翻訳機の学習には原言語、目的言語ともに 50 単語以下かつ文対の単語数の比が 9 以下の条件を満たす 180 万文対を用いた。統計的機械翻訳機として Moses⁸の PBSMT を用いた。5-gram 言語モデルを KenLM⁹を用いて訓練した。ハイパーパラメータのチューニングは開発セットを用いて MERT[12] で 3 回行なった。それぞれの設定でテストデータの翻訳を評価した評価値の平均を用いる。ニューラル機械翻訳機として OpenNMT¹⁰の注意機構モデル [8] を用いた。語彙は原言語、目的言語ともに頻度の上位 5 万語を用い、単語ベクトルの次元数は 500、隠れ層のベクトルの次元数は 500 とした。デフォルトの設定に従い、エンコーダ、デコーダともに 2 層の LSTM を用いた。バッチサイズは 64 文で 13 エポックの学習を行なった。評価指標には BLEU[13] および RIBES[14] を用いる。

²<http://stanfordnlp.github.io/CoreNLP/>

³<http://www.nactem.ac.uk/enju/>

⁴<http://taku910.github.io/mecab/>

⁵<http://github.com/moses-smt/giza-pp>

⁶<http://chainer.org/>

⁷<http://github.com/google/topdown-btg-preordering>

⁸<http://www.statmt.org/moses/>

⁹<http://github.com/kpu/kenlm>

¹⁰<http://opennmt.net/>

表 1: ベクトルの次元数と品詞・構文タグの有無による BLEU の変化 (50 万文対で学習)

ノードの次元	100	200	500
並び替えなし	22.73		
品詞タグなし	24.63	24.95	25.02
品詞タグあり	25.22	25.41	25.38

表 2: 翻訳性能 (180 万文対で学習): 最も性能の高いものと有意差がないもの ($p < 0.05$) を太字で表す。

	PBSMT		NMT	
	BLEU	RIBES	BLEU	RIBES
並べ替えなし	22.88	64.07	32.68	81.68
BTG	29.51	77.20	28.91	79.58
RvNN	29.16	76.39	29.01	79.63

3.2 実験結果

まず RvNN における品詞・構文タグおよびベクトルの次元数が事前並び替えおよび PBSMT の翻訳精度に与える影響を検証するため、実験を行なった。表 1 に、単語のみを入力とした場合と品詞・構文タグを付与した場合の BLEU 値を示す。この実験では PBSMT の訓練は 50 万文対を用いて行い、歪み制約は 6 とした。品詞・構文タグの有無に関わらずベクトルの次元数が 100 の時に比べて 200 の方が BLEU が向上するが、次元数を 500 に増加させても翻訳精度の向上に寄与しないことが分かる。品詞・構文タグを用いることで、200 次元でもタグなしで 500 次元用いたときと同等の翻訳性能を達成しており、品詞・構文タグの利用は翻訳速度の向上に効果的である。

表 1 の結果からノードの次元数を 200 に固定し、180 万文対を用いて PBSMT、NMT により翻訳を行なった結果を表 2 に示す。BTG と RvNN で事前並び替えを行なったものは PBSMT において歪み制約 0 で翻訳を行い、並び替えなしのものは歪み制約 6 とした。

PBSMT で翻訳を行なった場合、並び替えなしに比べて RvNN と BTG の両方で有意に BLEU と RIBES が向上しており、事前並び替えを行うことで翻訳精度が大きく向上していることが分かる。RvNN と BTG では、BLEU および RIBES において統計的有意差は認められなかった (p 値はそれぞれ $p = 0.068$, $p = 0.226$ であった)。このことから提案手法では、煩雑な特徴量の設計を必要とすることなく、事前並び替え手法の state-of-the-art である BTG と同等の翻訳性能を達成していることが分かる。しかし、NMT では事前並び

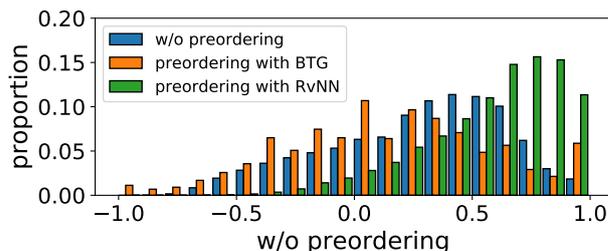


図 2: 並び替えによる順位相関係数の変化

替えをすると並び替えなしに比べて BLEU と RIBES がともに低下する。これは Sudo ら [14] の英中翻訳における実験結果と共通の現象であり、原因の一つとして、事前並び替えにより言語の構造が崩れてしまうことが考えられる。今後、提案手法を NMT のモデルに組み込み同時に学習を行うことで、NMT における翻訳精度の向上に取り組む予定である。

図 2 に並び替えの訓練データにおける Kendall の τ の割合を示す。並び替えることで並び替え前より Kendall の τ の値が高い対訳文ペアの割合が大きく向上しており、目的言語の語順に近づくように並び替えを学習できていることが分かる。また先行手法の BTG よりも RvNN による並び替えの方が Kendall の τ が高い対訳文の割合が大きくなっており、RvNN の方が事前並び替えの性能が高いことがわかる。しかし一方で RvNN は構文解析を必要とするため、構文解析が失敗した文は事前並び替えにも失敗しやすい。例えば、図 3 において“(1811)”というフレーズが誤って二つの句に分断されており、その結果事前並び替えにも失敗している。今後、構文解析を事前に行うのではなく、並び替えに適した 2 分木の構築を同時に行うモデルを検討する。

表 3 に図 3 に示した入力文の BTG および RvNN による事前並び替えの結果、および翻訳結果を示す。並び替えなしの訳文に比べると、並び替えを行うことで翻訳文の品質が向上していることが分かる。

4 終わりに

本論文では RvNN を用いた特徴量設計を必要としない事前並び替え手法を提案した。提案手法は複雑な特徴量設計に基づく事前並び替え手法の state-of-the-art [9] と同等の翻訳性能を達成している。今後、構文解析とノードの並び替えを同時に行うモデル、またその NMT への統合に取り組む予定である。

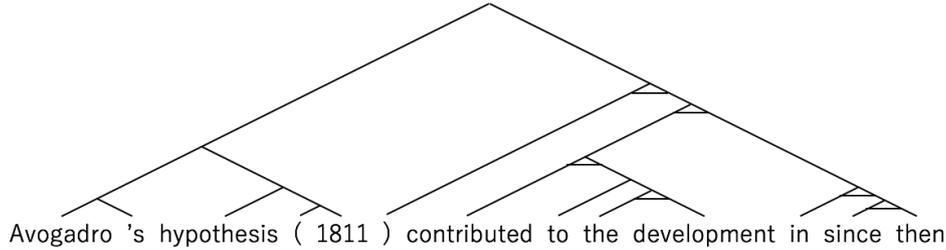


図 3: 構文解析の失敗例 (横線が引いてある節ノードは “Inverted”)

表 3: 並び替えと翻訳例

並び替え例	
原文	avogadro's hypothesis (1811) contributed to the development in since then.
BTG	avogadro's hypothesis (1811) the then since in development to contributed .
RvNN	avogadro's hypothesis (1811 then since in to development the contributed).
翻訳例	
参照訳	Avogadro の仮説 (1811) は, 以後の発展に貢献した。
並び替えなし	Avogadro の仮説 (1811) の開発に貢献し以後である。
BTG	Avogadro の仮説 (1811) 以後の発展に貢献した。
RvNN	Avogadro の仮説 (1811 以降のこれらの開発に貢献した。

謝辞

本研究は、日本電信電話株式会社 コミュニケーション科学基礎研究所の助成を受けたものである。

参考文献

- [1] A. de Gispert, G. Iglesias, and B. Byrne. Fast and Accurate Preordering for SMT using Neural Networks. In *Proc. of NAACL-HLT*, pages 1012–1017, May–June 2015.
- [2] D. Genzel. Automatically Learning Source-side Reordering Rules for Large Scale Machine Translation. In *Proc. of COLING*, pages 376–384, August 2010.
- [3] K. Hayashi, K. Sudoh, H. Tsukada, J. Suzuki, and M. Nagata. Shift-Reduce Word Reordering for Machine Translation. In *Proc. of EMNLP*, pages 1382–1386, October 2013.
- [4] S. Hoshino, Y. Miyao, K. Sudoh, K. Hayashi, and M. Nagata. Discriminative Preordering Meets Kendall's τ Maximization. In *Proc. of ACL-IJCNLP*, pages 139–144, July 2015.
- [5] H. Isozaki, K. Sudoh, H. Tsukada, and K. Duh. Head Finalization: A Simple Reordering Rule for SOV Languages. In *Proc. of WMTMetricsMATR*, pages 244–251, July 2010.
- [6] M. G. Kendall. A New Measure of Rank Correlation. *Biometrika*, 30(1/2):81–93, 1938.
- [7] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *CoRR*, abs/1412.6980, December 2014.
- [8] T. Luong, H. Pham, and C. D. Manning. Effective Approaches to Attention-based Neural Machine Translation. In *Proc. of EMNLP*, pages 1412–1421, September 2015.
- [9] T. Nakagawa. Efficient Top-Down BTG Parsing for Machine Translation Preordering. In *Proc. of ACL-IJCNLP*, pages 208–218, July 2015.
- [10] T. Nakazawa, M. Yaguchi, K. Uchimoto, M. Utiyama, E. Sumita, S. Kurohashi, and H. Isahara. ASPEC: Asian Scientific Paper Excerpt Corpus. In *Proc. of LREC*, pages 2204–2208, May 2016.
- [11] G. Neubig, T. Watanabe, and S. Mori. Inducing a Discriminative Parser to Optimize Machine Translation Reordering. In *Proc. of EMNLP-CoNLLs*, pages 843–853, July 2012.
- [12] F. J. Och. Minimum Error Rate Training in Statistical Machine Translation. In *Proc. of ACL*, pages 160–167, July 2003.
- [13] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proc. of ACL*, pages 311–318, July 2002.
- [14] K. Sudoh and M. Nagata. Chinese-to-Japanese Patent Machine Translation based on Syntactic Pre-ordering for WAT 2016. In *Proc. of WAT*, pages 211–215, December 2016.
- [15] C. Tillmann. A Unigram Orientation Model for Statistical Machine Translation. In *Proc. of HLT-NAACL*, pages 101–104, May 2004.
- [16] M. Utiyama and H. Isahara. A Japanese-English Patent Parallel Corpus. In *Proc. of the Machine Translation Summit XI*, September 2007.