事前並び替え位置表現を用いたTransformerによる日英機械翻訳

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.yuki, arase}@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相違は翻訳精度に大きな影響を与えることが知られている [4]。この問題を解決する手法の一つとして、翻訳器に入力する前に原言語文の語順を目的言語の語順に近づくように並び替える事前並び替え手法が提案されている [2, 4]。先行研究により、特に英語・日本語のように語順の大きく異なる言語対において、事前並び替え手法は SMT の翻訳精度を大幅に改善することが示されてきた [2, 4]。しかし ニューラル機械翻訳 (NMT) では、事前並び替えを行った原言語文を入力すると、並び替えをせずそのまま翻訳を行うよりも翻訳精度が低下することが報告されている [1, 2]。

近年では Transformer モデル [7] が提案され、再帰型ニューラルネットワーク (RNN) による NMT の翻訳精度を大きく上回った。前のタイムステップにおける出力が再帰的に入力される RNN とは異なり、Transformer モデルは各単語を独立にエンコードするため、単語の位置を考慮することが出来ない。そのため、単語の位置情報として絶対的位置表現 [7] や相対位置表現 [6] が用いられている。しかし、これらの手法では原言語文と目的言語文の語順の違いは一切考慮されない。

本研究では、事前並び替え手法により計算した原言語・目的言語間の語順の差異を Transformer モデルで活用する手法を提案する。具体的には、事前並び替えを行った後の単語の位置情報を事前並び替え位置表現として入力することで、原言語と目的言語の双方における語順を考慮したエンコーディングを行う。前述の通り、NMT モデルに事前並び替えを行った文を入力する直接的な方法では翻訳精度が低下する。そこで Transformer における位置表現に事前並び替えの結果得られた語順を適用し、ソフトな制約として語順を考慮することで、翻訳精度を向上できると期待する。

ASPEC コーパス [5] を用いた翻訳実験の結果、提案手法を用いることで日英翻訳において翻訳精度が向上することを確認した。純粋な Transformer と比較して、特に長文において提案手法の翻訳精度が向上することを示す。

2 提案手法

本章では、まず提案手法の基礎となる Transformer における位置表現の一つである相対的位置表現について説明する。次に提案手法である事前並び替え位置表現について述べる。

2.1 相対的位置表現

Shaw ら [6] は Transformer モデルにおいて、単語の絶対的な位置を用いる絶対的位置表現の代わりに相対的な位置を用いる相対的位置表現を提案した。前の層から $\mathbf{e} = \{\mathbf{e}_1, \mathbf{e}_1, \cdots, \mathbf{e}_u\} \in \mathbb{R}^{d_e}$（$u$ はエンコーダでは原言語文の長さ、デコーダでは目的言語文の長さを表す）を受け取り、アテンションを計算する。その後、計算したアテンションに基づいて各単語の隠れ表現 $\mathbf{z} = \{\mathbf{z}_1, \mathbf{z}_1, \cdots, \mathbf{z}_u\} \in \mathbb{R}^{d_z}$ を計算し出力する。i 番目の単語の隠れ表現を計算する際の j 番目の単語の相対的位置表現 ($\mathbf{a}^K_{ij}$、$\mathbf{a}^V_{ij} \in \mathbb{R}^{d_z}$) の式を以下に示す。

$$\mathbf{a}^K_{ij} = \mathbf{rel}_{\mathrm{clip}(j-i,k)} E^K_a \quad (1)$$

$$\mathbf{a}^V_{ij} = \mathbf{rel}_{\mathrm{clip}(j-i,k)} E^V_a \quad (2)$$

$$\mathrm{clip}(x,k) = \max(-k, \min(k,x)) + k$$

$\mathbf{a}^K_{ij}$ はアテンションのスコアの計算に使用され、$\mathbf{a}^V_{ij}$ は単語の隠れ表現の計算に使用される。k はクリップ幅であり、この k に基づいてモデルは i 番目の単語を中心とした $2k+1$ 単語の位置を考慮することが可能になる。$\mathbf{rel}_i \in \mathbb{R}^{2k+1}$ はワンホットベクトルであり、E^K_a、

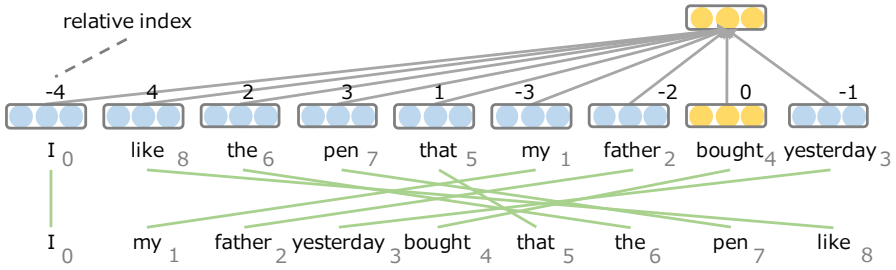

図 1: “bought” の隠れ表現を計算する際の事前並び替え位置表現

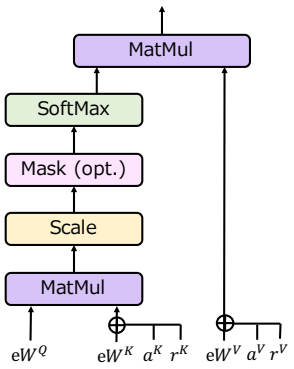

図 2: 事前並び替え位置表現を用いたアテンションモデル

$E_a^V \in \mathbb{R}^{(2k+1)\times d_z}$ は相対的位置表現の行列である。この行列は学習されるパラメータである。

相対的位置表現は式 (3)、(4) において用いられる。

$$\mathbf{z}_i = \sum_{j=1}^{n} \alpha_{ij}(\mathbf{e}_j W^V + \mathbf{b}^V + \mathbf{a}_{ij}^V) \quad (3)$$

$$\alpha_{ij} = \frac{\exp(\mathbf{s}_{ij})}{\sum_{k=1}^{u}\exp(\mathbf{s}_{ik})}$$

$$\mathbf{s}_{ij} = \frac{(\mathbf{e}_i W^Q + \mathbf{b}^Q)(\mathbf{e}_j W^K + \mathbf{b}^K + \mathbf{a}_{ij}^K)^{\mathrm{T}}}{\sqrt{d_z}} \quad (4)$$

W^V、W^Q、$W^K \in \mathbb{R}^{d_e\times d_z}$ は重み行列であり、$\mathbf{b}^V$、$\mathbf{b}^Q$、$\mathbf{b}^K \in \mathbb{R}^{d_z}$ はバイアスである。まず相対的位置表現を用いて i 番目の単語の隠れ表現を計算する際の j 番目の単語のスコア $\mathbf{s}_{ij}$ を計算し（式 (4)）、そのスコアに基づいて各単語の隠れ表現のアテンション α_{ij} が計算される。単語の新たな隠れ表現 $\mathbf{z}_i$ はアテンション α_{ij} による重み付き和として計算される。

2.2 事前並び替え位置表現

事前並び替え位置表現では、事前並び替え手法によって並び替えを行なった原言語文の位置情報を用いて、相対的位置表現と同じ方法で並び替えの隠れ表現を学習しアテンションの計算を行う。図 1 に “I like the pen that my father bought yesterday” という文に対して事前並び替え位置表現を用いた例を示す。事前並び替えの結果、“I my father yesterday bought that the pen like” という語順の文が得られ、元の原言語文に対する並び替え後の文の単語インデックス列 $\mathbf{p}$ は $\{0,8,6,7,5,1,2,4,3\}$ となる。例として “bought” の隠れ表現を計算する際、各単語の相対位置は $\{-4,4,2,3,1,-3,-2,0,-1\}$ となる。並び替えた後の “pen” の位置は 7 であり、“bought” の位置は 4 であるため、“pen” の相対位置は $7-4=3$ となる。また、並び替えた後の “yesterday” の位置は 3 であるため相対位置は $3-4=-1$ となる。

事前並び替え位置表現は式 (1)、(2) と同様に計算する。図 2 に相対的位置表現を用いたアテンションモデルの概要を示す。事前並び替え表現の式を以下に示す。

$$\mathbf{r}_{ij}^K = \mathbf{rel}_{\mathrm{clip}(\mathbf{p}_j-\mathbf{p}_i,k)}E_r^K$$
$$\mathbf{r}_{ij}^V = \mathbf{rel}_{\mathrm{clip}(\mathbf{p}_j-\mathbf{p}_i,k)}E_r^V$$

E_r^K、$E_r^V \in \mathbb{R}^{(2k+1)\times d_z}$ は事前並び替え位置表現の行列である。$\mathbf{r}_{ij}^K$、$\mathbf{r}_{ij}^V$ は式 (3)、(4) で足し合わされる。

$$\mathbf{z}_i = \sum_{j=1}^{n} \alpha_{ij}(\mathbf{e}_j W^V + \mathbf{b}^V + \mathbf{a}_{ij}^V + \mathbf{r}_{ij}^V)$$

$$\mathbf{s}_{ij} = \frac{(\mathbf{e}_i W^Q + \mathbf{b}^Q)(\mathbf{e}_j W^K + \mathbf{b}^K + \mathbf{a}_{ij}^K + \mathbf{r}_{ij}^K)^{\mathrm{T}}}{\sqrt{d_z}}$$

これによって、並び替えを考慮した単語の隠れ表現 $\mathbf{z}_i$ を計算できる。

3 翻訳実験

3.1 実験設定

ASPEC コーパス [5] を用いて日英・英日翻訳実験を行なった。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

表 1: テストデータにおける翻訳性能を示す。太字は最高スコアならびに $p < 0.05$ で最高スコアと有意差がないものを表す。+BTG と+RvNN はそれぞれ提案手法における事前並び替え手法として BTG と RvNN を用いたものを表す。「BTG のみ」、「RvNN のみ」は事前並び替えを行なった文を直接入力する場合の結果を表す。

	英日		日英	
	BLEU	RIBES	BLEU	RIBES
baseline	**35.53**	**83.68**	23.94	**76.06**
+BTG	34.50	**83.35**	**25.28**	**76.52**
+RvNN	35.05	**83.42**	**25.06**	**76.52**
BTG のみ	31.20	81.15	22.11	73.51
RvNN のみ	32.18	81.83	20.99	73.88
オラクル	44.84	89.85	36.45	87.91

タは 1,812 文対である。翻訳器の学習には上位 200 万文対のうち、原言語、目的言語ともに 50 単語以下、かつ単語数の比が 9 未満である約 180 万文対を用いた。

英語文の単語分割および品詞タグ付けは Stanford Core NLP[1]で行い、構文解析は Enju[2]で行なった。日本語文は Juman[3]で形態素解析を行い、構文解析は Ckylark[4]で行なった。

事前並び替え手法として、BTG モデル [4] と RvNN モデル [2] を用いた。BTG モデル[5]は訓練データからランダムにサンプリングした 10 万文対で訓練を 20 イテレーション行なった。単語クラスタ数は 256 に設定した。RvNN モデルは BTG の学習に用いた 10 万文と同じ文で 5 エポック訓練した。語彙サイズは 5 万、ミニバッチサイズは 500 とした。また、単語および品詞ベクトルの次元は 200 とした。単語アライメントは MGIZA[6]を用いた。

提案手法は OpenNMT-py[7]の Transformer モデルに実装した。絶対的位置表現と相対的位置表現を共に用いたものをベースラインとした。語彙サイズは 5 万に設定し、エンコーダとデコーダは両方とも 6 層とした。単語ベクトル、隠れ層の次元は 512 とし、マルチヘッドアテンションのヘッド数は 8 とした。相対的位置表現と事前並び替え位置表現のクリップサイズは 4 とした。学習率は 0.001 とし、Adam [3] を用いて学習を行なった。訓練は 25 万イテレーション行い、開発

[1]https://stanfordnlp.github.io/CoreNLP/
[2]https://github.com/mynlp/enju
[3]http://nlp.ist.i.kyoto-u.ac.jp/index.php?JUMAN
[4]https://github.com/odashi/ckylark
[5]https://github.com/google/topdown-btg-preordering
[6]https://github.com/moses-smt/mgiza
[7]https://github.com/OpenNMT/OpenNMT-py

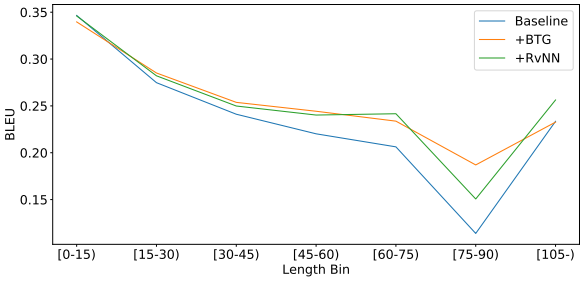

図 3: 日英翻訳における各文長ごとの BLEU 値の平均

表 2: テストデータ 200 文における訳抜けと重複訳の文数。BTG と RvNN は提案手法による翻訳結果を示す。

	訳抜け	重複訳
Baseline	21	1
+BTG	18	3
+RvNN	19	0

データにおいてパープレキシティが最小のモデルでテストデータの翻訳を行い、翻訳精度の評価を行なった。各モデルにおいて訓練は 3 回行なってそれぞれテストデータを評価し、その平均を最終的な評価値とした。

3.2 実験結果

表 1 に実験結果を示す。英日翻訳では、BLEU 値が BTG を用いたものでは 1.03 ポイント、RvNN を用いたものでは 0.48 ポイント有意に低下したが、日英翻訳においては、BLEU 値が BTG を用いたものでは 1.34 ポイント、RvNN を用いたものでは 1.12 ポイント有意に向上した。

一方、事前並び替えを行った文を直接入力する「BTG のみ」「RvNN のみ」では、ベースラインに比べて、英日翻訳では、BLEU 値が「BTG のみ」では 4.33 ポイント、「RvNN のみ」では 3.35 ポイント、RIBES 値が「BTG のみ」では 2.53 ポイント、「RvNN のみ」では 1.85 ポイント低下した。日英翻訳では、BLEU 値が「BTG のみ」では 1.83 ポイント、「RvNN のみ」では 2.95 ポイント、RIBES 値が「BTG のみ」では 2.55 ポイント、「RvNN のみ」では 2.18 ポイント低下した。これらは先行研究 [1, 2] と一貫した結果であり、提案手法において事前並び替えを位置表現として利用する有効性が示された。

表 1 の「オラクル」は、提案手法において理想的な事前並び替えを行ったものである。オラクルな事前並

表 3: 日英翻訳における翻訳例

原言語文	調査が潮間帯に限られたため得られた種類数は少なかった。
参照訳	A few species were obtained, **since the survey was limited** to the intertidal zone.
ベースライン	The number of species obtained in the intertidal zone was small.
提案手法 (RvNN)	The number of species obtained was small because the survey was limited to the intertidal zone.
提案手法 (BTG)	The number of species obtained was small because the investigation was limited to intertidal zone.

び替えは MGIZA で計算された単語アライメントをもとに、アライメントの交差がなくなるようヒューリスティックに並び替えたものである。ベースラインと比較して、「オラクル」では日英・英日翻訳ともに、翻訳精度が大幅に向上している。提案手法の翻訳精度が英日対では低下した一因として、事前並び替えの失敗が翻訳に悪影響を与えていることが考えられる。

3.3 分析

図 3 に原言語文の長さごとに評価した BLEU 値のグラフを示す。原言語文が 45 単語までの文においては各モデルでの翻訳精度に大きな差はないが、45 単語から 90 単語までの文において、提案手法の翻訳精度はベースラインの翻訳精度を大きく上回っている。

表 2 にテストデータから 200 文をランダムサンプルしたうちの訳抜けと重複訳の数を示す。ベースラインと比較して、提案手法を用いた翻訳では訳抜けの数が減少している。また、表 3 に日英翻訳における翻訳例を示す。ベースラインでは “since the survey was limited” に当たる部分が出力されていないが、提案手法を用いた翻訳では出力されている。これは事前並び替えによって原言語文の理想的な翻訳順序が考慮されることで、訳抜けが減少するためと考えられる。

4 まとめ

本稿では Transformer モデルにおいて事前並び替え手法を活用する、事前並び替え位置表現の提案を行った。評価実験により、純粋な Transformer と比較して翻訳精度が向上することを示し、特に長文の翻訳において高い効果を発揮することを示した。

今後の課題として、事前並び替えと翻訳モデルを同時に学習する手法を検討する予定である。

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および科研費#19K20343 の助成を受けたものである。

参考文献

[1] Jinhua Du and Andy Way. Pre-reordering for neural machine translation: Helpful or harmful? *The Prague Bulletin of Mathematical Linguistics (PBML)*, Vol. 108, pp. 171–182, June 2017.

[2] Yuki Kawara, Chenhui Chu, and Yuki Arase. Recursive neural network based preordering for english-to-japanese machine translation. In *Proc. of ACL-SRW*, pp. 21–27, July 2018.

[3] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *CoRR*, Vol. abs/1412.6980, , December 2014.

[4] Tetsuji Nakagawa. Efficient Top-Down BTG Parsing for Machine Translation Preordering. In *Proc. of ACL-IJCNLP*, pp. 208–218, July 2015.

[5] Toshiaki Nakazawa, Manabu Yaguchi, Kiyotaka Uchimoto, Masao Utiyama, Eiichiro Sumita, Sadao Kurohashi, and Hitoshi Isahara. ASPEC: Asian Scientific Paper Excerpt Corpus. In *Proc. of LREC*, pp. 2204–2208, May 2016.

[6] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In *Proc. of NAACL-HLT*, pp. 464–468, June 2018.

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proc. of NIPS*, pp. 5998–6008. December 2017.