フレーズベース機械翻訳における 単語間の情報を考慮した深層学習による語彙化並べ替えモデル

叶内 晨* 小町 守

首都大学東京

1 はじめに

近年，統計的機械翻訳のひとつとして，フレーズベース機械翻訳 [5]（以下，PBSMT と呼ぶ）が広く使われている．PBSMT における問題点の一つが語順並べ替えであり，特に，日英などの語順が大きく異なる言語対における翻訳では深刻な問題である．語順並べ替えには様々な手法が存在し，従来の並べ替えモデル [15, 1] に加えて，原言語と目的言語の語順を近づける事前並べ替え [14, 3, 9] や事後並べ替え [12] が存在する．

本論文では，従来の並べ替えモデルのうち，PBSMT で広く使われている語彙化並べ替えモデル [13, 4, 2] に焦点を当てる．Li ら [7] は，並べ替えスコアを計算する際に，深層学習の手法により，現在の原言語側と目的言語側の句ペアに加えて一つ前の句ペアを利用することで，データスパースネスの問題を解決した．その結果，並べ順の曖昧性が解消し，中英翻訳の精度が向上したと報告している．しかし Li らは，広い範囲の句情報を取り入れることを目的とした一方で，句の内部構成や，対訳ペア同士の句の関係は考慮していない．

そこで本研究では，Li らの手法に加え，句ベクトルの構成時に，単語のアライメント情報を利用し，句内部の単語同士の関係を考慮する．さらに，句の翻訳確率を利用することで，対訳ペアとなる 2 言語の句の繋がりを考慮した並べ替えモデルを提案する．また，日英翻訳の並べ替えの特徴に合わせ，語彙化並べ替えモデルの複数の並べ順を定義し，実験を行った．

その結果，Li らの手法より並べ替えの精度が 1.67 ポイント上昇し，提案したモデルを翻訳に組み込んだ結果，Moses に対して有意水準 p = 0.01 で有意差が得られた．

2 語彙化並べ替えモデル

語彙化並べ替えモデルは，統計的な手法により各句ペアに対して並べ替え確率を計算する．原言語文 $\boldsymbol{f}$ が与えられたとすると，文を句 $\boldsymbol{f} = \overline{f}_{a_1}, .., \overline{f}_{a_i}, .., \overline{f}_{a_I}$ に分割し，各句毎に目的言語の句 $\overline{e}_i$ へ翻訳し，並べ替えを行うことによって目的言語文 $\boldsymbol{e} = \overline{e}_1, .., \overline{e}_i, .., \overline{e}_I$ を生成する．このとき，$\boldsymbol{a} = a_1, .., a_I$ は，原言語側と目的言語側の句 $\overline{f}_{a_i}, \overline{e}_i$ が対応する句アライメントを表す．

目的言語側で i および $i-1$ 番目の句 $\overline{e}_i, \overline{e}_{i-1}$ に対応する原言語側の句 $\overline{f}_{a_i}, \overline{f}_{a_{i-1}}$ の位置関係を，4 つの並べ順 [8] で分類し，o で表す．

$$o(a_i, a_{i-1}) = \begin{cases} \text{Mono} & (a_i - a_{i-1} = 1) \\ \text{Swap} & (a_i - a_{i-1} = -1) \\ \text{D}_{\text{right}} & (a_i - a_{i-1} > 1) \\ \text{D}_{\text{left}} & (a_i - a_{i-1} < -1) \end{cases} \quad (1)$$

目的言語側で隣接する句 $\overline{e}_i, \overline{e}_{i-1}$ に対して，Monotone（Mono）は原言語側の句 $\overline{f}_{a_i}, \overline{f}_{a_{i-1}}$ が隣接かつ目的言語側と同順であり，Swap は $\overline{f}_{a_i}, \overline{f}_{a_{i-1}}$ が隣接かつ目的言語側と逆順である．Discontinuous-right（D_{right}）は $\overline{f}_{a_i}, \overline{f}_{a_{i-1}}$ の両者が分離かつ目的言語側と同順であること，Discontinuous-left（D_{left}）は $\overline{f}_{a_i}, \overline{f}_{a_{i-1}}$ の両者が分離かつ目的言語側と逆順であることを表す．語順並べ替えの少ない言語対では，D_{right} と D_{left} をまとめて Discontinuous として扱うのが一般的である．

句毎の並べ替え確率を $P(o(a_i, a_{i-1}) \,|\overline{f}_{a_i}, \overline{e}_i)$ とし，目的言語側の全ての隣接する句に対して計算を行うと，文の並べ替えスコアは以下となる．

$$P(a_1^I|\boldsymbol{e}) = \sum_{i=1}^{I} P(o(a_i, a_{i-1})|\overline{f}_{a_i}, \overline{e}_i) \quad (2)$$

語彙化並べ替えモデルは，全ての句毎に並べ替えの確率を統計的に計算したシンプルな手法で，機械翻訳の精度向上に貢献している．しかし語彙化並べ替えモデルには，3 つの問題点が挙げられる．

- 句毎の並べ順 o の曖昧性が高い．単一の句ペアでは，その句の前後に来る句の並べ順は一意に決まらず，翻訳時に並べ順が一意に決まらない．
- データスパースネスの問題がある．並べ替え確率は句ペア毎に計算され，各句ペアにスコアが存在する．しかし，学習時に一度しか出現していない句ペアは

*kanouchi-shin@ed.tmu.ac.jp

フレーズテーブル全体の 95%[1] にのぼり，それらの句ペアは並べ替え確率を学習できていない．

- 単語間の対応が考慮されていない．句ペアは複数の単語とそのアライメントからなるが，句内の単語間の関係や，句ペアの繋がりの強さが考慮されていない．

並べ順の曖昧性解消の問題に対して，句の 4 つ組み $\overline{f}_{a_i}, \overline{e}_i, \overline{f}_{a_{i-1}}, \overline{e}_{i-1}$ 毎に最尤推定を行う方法が考えられるが，データスパースネスの問題によって現実的でない．そこで語彙化並べ替えモデルの改善として，深層学習を用いたモデル [6, 7] が提案されている．

3 深層学習による並べ替えモデル

3.1 句の分散表現

Socher ら [11] の Recursive Autoencoder は単語ベクトルを再帰的に Autoencoder で次元圧縮し，文や句の意味ベクトルを教師無しで生成するモデルである．このモデルでは，文や句の意味ベクトルを全て同じ次元で表現することが可能である．u 次元の単語ベクトルを $x \in \mathcal{R}^u$，エンコード用の重み行列を $W_e \in \mathcal{R}^{u \times 2u}$，バイアス項を b_e とすると，句ベクトル $p_{1:2}$ は以下で表現できる．

$$p_{1:2} = f(W_e[x_1; x_2] + b_e) \tag{3}$$

句の単語数が 3 語以上からなる場合，句ベクトル $p_{1:n}$ を句ベクトル $p_{1:n-1}$ と単語ベクトル x_n から得る．

$$p_{1:n} = f(W_e[p_{1:n-1}; x_n] + b_e) \tag{4}$$

入力ベクトルと Autoencoder によって再構築されたベクトルの平均二乗誤差を最小化するようにパラメータを調整することで，適切な句ベクトル $p_{1:n}$ を得る．

3.2 ニューラル並べ替えモデル（NRM）

入力層とソフトマックス層からなる，ニューラル並べ替えモデル [6] を定義する．現在と 1 つ前の原言語側と目的言語側の句ベクトル $p(\overline{f}_{a_i}), p(\overline{e}_i), p(\overline{f}_{a_{i-1}}), p(\overline{e}_{i-1})$ を入力とし，ソフトマックス層で並べ替えスコア $P(o_i)$ を出力する．このとき，同じ言語の句ベクトルは同じパラメータの Recursive Autoencoder で学習する．

$$P(o_i) = \frac{\exp g(o_i)}{\Sigma_{o' \in \{M,S,D_r,D_l\}} \exp g(o')} \tag{5}$$

$$g(o_i) = f(W_r[PH_i; PH_{i-1}] + b_r) \tag{6}$$

$$PH_i = [p(\overline{f}_{a_i}); p(\overline{e}_i)] \tag{7}$$

$o \in \{\mathrm{Mono}, \mathrm{Swap}, \mathrm{D_{right}}, \mathrm{D_{left}}\}$ は 2 節で説明した並べ方を表し，$W_r \in \mathcal{R}^{1 \times 4n}$ は重み行列，PH_i は句ベクトル $p(\overline{f}_{a_i}), p(\overline{e}_i)$ の連結，b_r はバイアス項である．

クロスエントロピーを利用し，句ペア毎のニューラル並べ替えモデルにおける誤差 $E_{nrm}(\theta)$ を計算する．

$$E_{nrm}(\theta) = -\sum_o d_i(o) \log P(o) \tag{8}$$

[1] 京都フリー翻訳タスクコーパスに対し実験を行った．

表 1: 単語アライメント情報 A のベクトル化

次元	条件	
1	単語の翻訳確率 $P(e\|f)$	
2	単語の逆翻訳確率 $P(f\|e)$	
3	単語のアライメント先が，	左端の場合 1
4	反対言語側の句における	中央の場合 1
5	どこの単語に対応付くか	右端の場合 1
6	単語が NULL アライメントの場合 1	

d_i は並べ順の正解を 4 次元の確率分布で表したもので，各次元が $\mathrm{Mono}, \mathrm{Swap}, \mathrm{D_{right}}, \mathrm{D_{left}}$ を意味する．例えば，正解が Swap の場合，$d_i = [0, 1, 0, 0]$ となる．

最後に，4 つの Recursive Autoencoder の誤差 $E_{rae}(\theta)$ とニューラル並べ替えモデルの誤差 $E_{nrm}(\theta)$ の合計の誤差を $J(\theta)$ とし，誤差逆伝播を行った．α は比率を調整するハイパーパラメータ，λ は L_2 正則化係数を表す．

$$J(\theta) = \alpha E_{rae}(\theta) + (1 - \alpha) E_{nrm}(\theta) + \lambda ||\theta||^2 \tag{9}$$

4 単語間の情報を考慮したモデル

句ペア $\overline{f}_{a_i}, \overline{e}_i$ の並べ順はそれぞれの句ペアに依存し，句ペアは 2 言語におけるアライメントされた単語から構成される．しかし NRM では，単に単語ベクトルを畳み込むことで句ベクトルを生成しており，句内部の単語間の関係や，対訳ペアとなる句ペアの繋がりの強さは考慮されていない．そこで本論文では，句ペアを構成する際に鍵となる，単語毎のアライメントを考慮する．さらに，2 言語の対応する句 $\overline{f}_{a_i}, \overline{e}_i$ の翻訳確率をモデルへ加えることで，句ペアの繋がりの強さを考慮したモデルを提案する．図 1 にモデルの全体図を示す．

4.1 単語アライメントの利用

句ペアは内部の単語アライメントの組み合わせからなる．しかし，アライメントされた単語ペアの翻訳確率は単語毎に様々である．また，一部の単語はアライメントされてはおらず，NULL アライメントも存在する．

提案手法では，これらのアライメント情報を句ベクトルに伝播させるため，単語ベクトル x に，6 次元からなる単語のアライメント情報 A を加えたものを新しい単語ベクトル x' として定義する．

$$x' = [x; A] \in \mathcal{R}^{u+6} \tag{10}$$

表 1 に，A の各次元の説明を示す．例えば，図 1 における “日本” という単語は，アライメント先の “Japan” が句の中央に位置するので，A の 4 次元目が 1 となる．

$$A = [P(\text{Japan}|\text{ 日本}), P(\text{日本 }|\text{Japan}), 0, 1, 0, 0] \tag{11}$$

4.2 句の翻訳確率の利用

対応する句ペアの繋がりの強さを考慮するため，2 言語の句の関係を 4 次元のベクトル $FE(\overline{f}_{a_i}, \overline{e}_i)$ で定義する．

$$\begin{aligned} FE(\overline{f}_{a_i}, \overline{e}_i) \quad = \quad & [P(\overline{e}_i|\overline{f}_{a_i}), P(\overline{f}_{a_i}|\overline{e}_i), \\ & lex(\overline{e}_i|\overline{f}_{a_i}), lex(\overline{f}_{a_i}|\overline{e}_i)] \end{aligned} \tag{12}$$

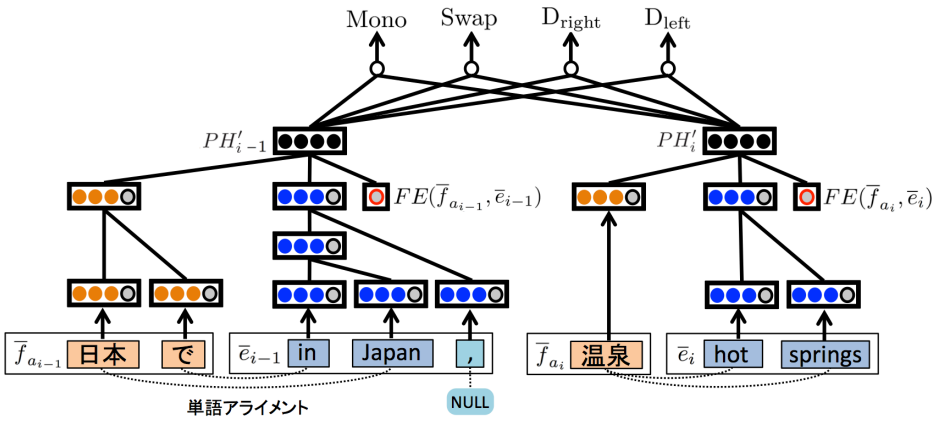

図 1: 単語アライメントと句の翻訳確率を考慮したニューラル並べ替えモデル

表 2: 並べ替え精度と各ラベルの正解率

		Mono	Swap	D_{right}	D_{left}	Acc.
データ比率（%）		33.89	11.68	31.80	22.63	
ベースライン	Moses の並べ替えモデル	71.54	36.92	**95.76**	39.33	66.71
	ニューラル並べ替えモデル（NRM）	77.06	57.60	70.31	60.63	68.22
提案手法	単語アライメントの利用（NRM+A）	76.90	59.84	71.03	**62.38**	69.14
	句の翻訳確率の利用（NRM+FE）	76.70	59.78	71.34	60.07	68.64
	NRM+A+FE	**77.53**	**60.83**	72.69	61.78	**69.89**

$P(\overline{e}_i|\overline{f}_{a_i}), P(\overline{f}_{a_i}|\overline{e}_i)$ は両方向の句の翻訳確率を表し，$lex(\overline{e}_i|\overline{f}_{a_i}), lex(\overline{f}_{a_i}|\overline{e}_i)$ は両方向の句内の単語の翻訳確率の平均を表している．$FE(\overline{f}_{a_i}, \overline{e}_i)$ と PH_i を連結し，重み行列 W_q を使うことで，新しい句ペアベクトル PH'_i を定義する． b_q はバイアス項とする．

$$PH'_i = f(W_q[PH_i; FE(\overline{f}_{a_i}, \overline{e}_i)] + b_q) \quad (13)$$

5 実験

句の 4 つ組 $(\overline{f}_{a_i}, \overline{e}_i, \overline{f}_{a_{i-1}}, \overline{e}_{i-1})$ を与えた場合に，どの程度句の並べ順 o_i を推定できるか実験を行った．

5.1 実験設定

日英の対訳データに，京都フリー翻訳タスク [2]（KFTT）を用いた．日本語の単語分割には KyTea [3]（ver.0.4.7）を，単語の対応付けには GIZA++ [4] と制約 grow-diag-final-and を用いた．翻訳ツールキットの Moses [5] を利用し，フレーズテーブル作成時のアライメント情報から，9,000 万セットの句の 4 つ組みを抽出し，正解の並べ順を付与した．クリーニングとして句の共起度が 1 回のものを取り除き，残った 600 万セットからランダムに訓練データ，テストデータ，チューニングデータを作成した [6]．訓練データを 100 万セット，テストデータ，チューニングデータを 1 万セットとした．

深層学習には Chainer [7]（ver.1.5.0.2）を利用し，単語 x の次元数を $n = 25$ とし，単語ベクトルの初期値として，各言語の Wikipedia で word2vec [8] を学習した．

5.2 実験結果

表 2 に並べ替えの各ラベルの正解率を示す．Moses の正解率は，学習時に作成される並べ替えテーブルのスコアから並べ順 o を計算した [9]．Moses の並べ順の結果は 2 つの句ペアの出現頻度によるスコアであるため，学習データで出現頻度が高い並べ順である Mono と D_{right} を優先的に当てる分類器になっている．一方，ニューラル並べ替えモデルでは，句の 4 つ組みからスコアを計算しているため，どの並べ順も同程度推定することに成功している．その結果として，D_{right} の 正解率のみ Moses に劣るが，Swap と D_{left} は勝り，全体の正解率も改善した．

NRM に加え，単語アライメントを考慮した場合 0.92 ポイント，フレーズの翻訳確率を考慮した場合 0.44 ポイント正解率が向上した．また，単語アライメントと句の翻訳確率の両方を考慮することで，1.67 ポイント正解率が向上し，より正確な句の並べ順を得られた．

5.2.1 学習データのサイズと並べ替えの正解率

表 3 に，学習データのサイズを変化させた時の提案手法の並べ替えの正解率を示す．ニューラル並べ替えモデルは，テストデータにおける単語が既知語の場合，未知の句に対しても，似た句ベクトルを表現することができる．そのため，5 万セットの学習データでも，全ての句

[2] http://www.phontron.com/kftt/index-ja.html
[3] http://www.phontron.com/kytea/index-ja.html
[4] http://www.statmt.org/moses/giza/GIZA++.html
[5] http://www.statmt.org/moses/
[6] その際，各データは異なる文の句を使用した．
[7] http://docs.chainer.org/en/stable/
[8] https://code.google.com/p/word2vec/
[9] 既知の句にのみスコアが計算されるため，Moses のスコア計算時のみ，学習データにテストデータを混ぜた．

表 3: データサイズと並べ替えの正解率

データサイズ	単語の異なり数 日本語	英語	学習が w2v のみの単語 日本語	英語	未知語 日本語	英語	未知の句 日本語	英語	Acc.
1 万	4,906	4,820	1,532 (44%)	1,604 (45%)	0.66%	13%	4,875 (61%)	5,173 (63%)	63.50
5 万	10,833	10,880	828 (24%)	883 (25%)	0.66%	11%	3,829 (48%)	4,153 (51%)	66.88
20 万	18,239	18,375	432 (12%)	491 (14%)	0.63%	8.0%	2,822 (36%)	3,223 (39%)	68.45
100 万	26,978	27,152	233 (6.7%)	280 (7.8%)	0.61%	4.9%	1,915 (24%)	2,262 (28%)	**69.89**

(注：単語の異なり数は学習時のものを表し，学習が word2vec のみの単語，未知語，未知の句はテストデータにおける数を表す．テストデータにおける単語の異なりは，日英でそれぞれ 3,470 語，3,583 語であり，句の異なりはそれぞれ 7,945 句，8,187 句である．)

表 4: 翻訳における精度

手法	日英 BLEU	RIBES	英日 BLEU	RIBES
Moses	18.45	65.64	21.37	67.01
NRM	19.14†	**66.07**	22.75†	**68.89**†
NRM+A+FE	**19.27**†	65.67	**22.89**†	68.87†

短剣符は Moses との有意差を示す ($\dagger : p < 0.01$)

が既知である Moses に勝る正解率を示している．

一方，テストデータにおける単語が未知語の場合，その単語には word2vec のベクトルが初期値として与えられる．しかしこのベクトルは，並べ替えモデルのベクトルとしては最適に学習されておらず，うまく並べ替えのための句ベクトルを構成できていない可能性がある．また，word2vec にさえ出現しない単語には同一の未知語ベクトルが与えられている．データサイズが増加するごとに未知語，未知の句は単調に減少し，正解率も向上している．さらにデータを増やし，未知語・未知の句を減らすことで，正解率が向上すると考えられる．

5.3 機械翻訳における精度

提案したモデルを利用したときの翻訳精度を確認した．提案手法をデコーディングに組み込むには手間がかかるため，N-best リランキングを用いて実験を行った．データは KFTT を用い，チューニングには MERT を利用し，BLEU で最適化を行った．チューニングデータに対して翻訳の候補を 1000-best で出力し，そのうちの語彙化並べ替えモデルのスコアを提案手法のスコアに書き換えた後，MERT を回し直すことで素性の重みを調整し直した．その後，テストデータにおいて 1000-best の再評価を行い，もっともスコアの高い文を最終的な翻訳結果とした．評価尺度を BLEU と RIBES とした．

翻訳結果を表 4 に示す．提案手法を Moses と比べた場合，日英・英日翻訳において，BLEU で有意差を確認できた．一方，RIBES では最適化していないため，精度がモデル毎に揺れる結果となった．また，提案手法は NRM と比べて有意差を確認できなかった．この理由としては，NRM と提案手法は並べ替えの正解率で 1.67 ポイント差があるものの，ラベル毎に見ると変化は小さく，翻訳結果への影響も少なかったと考えられる．

6 おわりに

本研究では，PBSMT における語順並べ替えの問題に対し，深層学習を利用し改善を行った．句ベクトルの構成時に，単語アライメント，句ペアの繋がりを考慮したニューラル並べ替えモデルを提案することで，句の並べ順の正解率が向上した．さらに日英・英日翻訳において実験を行い，Moses の語彙化並べ替えモデルに対して BLEU で有意差を得た．Future work として，事前並べ替えを適応した実験，並べ替えの少ない言語対における実験，さらに大きな学習データによる実験を行いたい．

参考文献

[1] D. Chiang. Hierarchical phrase-based translation. *Computational Linguistics*, Vol. 33, No. 2, 2007.

[2] M. Galley and C. D. Manning. A simple and effective hierarchical phrase reordering model. In *EMNLP*, 2008.

[3] S. Hoshino, Y. Miyao, K. Sudoh, K. Hayashi, and M. Nagata. Discriminative preordering meets Kendall's τ maximization. In *ACL-IJCNLP*, 2015.

[4] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, and R. Zens. Moses: Open source toolkit for statistical machine translation. In *ACL(demo)*, 2007.

[5] P. Koehn, F. J. Och, and D. Marcu. Statistical phrase-based translation. In *NAACL-HLT*, 2003.

[6] P. Li, Y. Liu, and M. Sun. Recursive autoencoders for ITG-based translation. In *EMNLP*, 2013.

[7] P. Li, Y. Liu, M. Sun, T. Izuha, and D. Zhang. A neural reordering model for phrase-based translation. In *COLING*, 2014.

[8] M. Nagata, K. Saito, K. Yamamoto, and K. Ohashi. A clustered global phrase reordering model for statistical machine translation. In *COLING-ACL*, 2006.

[9] T. Nakagawa. Efficient top-down BTG parsing for machine translation preordering. In *ACL-IJCNLP*, 2015.

[10] K. Papineni, S. Roukos, T. Ward, and W. Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002.

[11] R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, and C. D. Manning. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *EMNLP*, 2011.

[12] K. Sudoh, X. Wu, K. Duh, H. Tsukada, and M. Nagata. Post-ordering in statistical machine translation. In *MT Summit*, 2011.

[13] C. Tillmann. A unigram orientation model for statistical machine translation. In *HLT-NAACL*, 2004.

[14] X. Wu, K. Sudoh, K. Duh, H. Tsukada, and M. Nagata. Extracting pre-ordering rules from predicate-argument structures. In *IJCNLP*, 2011.

[15] K. Yamada and K. Knight. A syntax-based statistical translation model. In *ACL*, 2001.