構文情報を利用した事前並べ替えとニューラルネットワーク機械翻訳の同時学習

外山 翔平 † 橋本 和真 ‡ 鶴岡 慶雅 †
† 東京大学 電子情報工学科, ‡ 東京大学 工学系研究科
{toyama, hassy, tsuruoka}@logos.t.u-tokyo.ac.jp

1 はじめに

近年のニューラルネットワークの様々な手法の開発により、統計的機械翻訳においてもニューラルネットワークを利用した機械翻訳が台頭してきている [3]。現在のニューラルネット機械翻訳では主に西洋言語間の翻訳が盛んに研究されており、中でも文頭から単語を入力し文頭から単語を直接出力するという Recurrent Neural Network（RNN）による統一的な翻訳モデル [2] が用いられている。

このような RNN を利用して翻訳を行う場合、言語間で対応している単語の位置関係が性能に大きく関わるという報告がある [8]。また、フレーズベース機械翻訳など多くの機械翻訳の手法において、語順の大きく異なる言語間の翻訳を行う場合は、事前並べ替えが有効である [1]。これらの理由から、英日翻訳のような文法や語順の大きく異なる言語間の翻訳を RNN を用いて行う場合においても、事前並べ替えを行うことが有効であると考えられる。しかしルールベースによって事前並べ替えを行う場合、翻訳する言語間にそれぞれルールを生成しなければならず、また RNN に最適化した並び替えを行うことは困難である。

そこで本研究では、構文情報を用いた事前並べ替えと RNN による翻訳を合わせて統一的な翻訳モデルを生成し、同時に学習するモデルを提案する。事前並べ替えは、構文木の各ノードにおける対数線形モデルによって決まる操作の組み合わせで表現する。それにより、翻訳する言語間の文法や語順の違いを効率的に学習でき、翻訳する言語の組み合わせに依らずそれぞれに最適な事前並べ替えが学習されることが期待される。

比較的短い文における英日翻訳の実験を行ったところ、事前並べ替えが言語の文法や語順の差に応じて適切に行われていることを確認した。

2 Recurrent Neural Network 機械翻訳

RNN はニューラルネットワークの一種で、中間層を再帰的に生成するため、入出力の数が定まっていない機械翻訳に適している。Cho らは Recurrent Neural Network Encoder-Decoder（RNN-ED）からなるモデルを構築し、単語列から単語列への直接の翻訳を可能にした [2]。

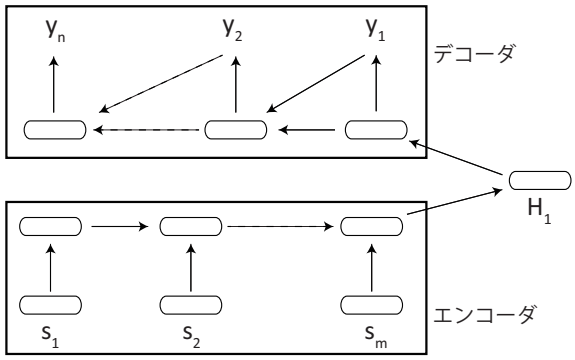

図 1: 簡略化した RNN-ED

これらの RNN 機械翻訳モデルは大きく分けて次の 2 段階からなる。

1. RNN に翻訳元言語の文を構成する単語のベクトル列を入力し、文ベクトルを得る（Encode）。
2. 翻訳元言語の文ベクトルを別の RNN に入力し、翻訳先言語の単語列を出力する（Decode）。

これらのエンコーダとデコーダについて、Cho らのモデルを簡略化したものを次に示す（図 1）。

2.1 簡略化した RNN-ED

2.1.1 原言語文から文ベクトル生成

文頭から単語のベクトルを再帰的に合成していくことで、文を構成する単語の情報を持った文ベクトルを生成する。翻訳元言語の各単語 x_i に対して単語ベクトル s_i を与えておく。

翻訳する文の各単語 $x_1, x_2, \cdots x_m$ に対応するベクトルを $s_1, s_2, \cdots s_m$ とする。これらのベクトル列を入力として、文ベクトル h_m を出力する RNN を式 (1) から求める。翻訳元言語の文ベクトル h_m に重み行列 W_3 をかけ、翻訳先言語の文ベクトル $H_1 = W_3 h_m$ を生成する。

$$h_k = \begin{cases} s_1 & (k=1) \\ \tanh(W_1 h_{k-1} + W_2 s_k) & (1 < k \leq m) \end{cases} \quad (1)$$

2.1.2 文ベクトルから翻訳文の展開

翻訳先言語の文ベクトル H_1 を用いて、文頭から順に一単語ずつ出力していくことで翻訳文を生成する。はじめに、翻訳先言語の各単語 y_i に対して単語ベクトルと t_i と重みベクトル V_i を与えておく。

前項の RNN から与えられた翻訳先言語の文ベクトル H_1 から、ソフトマックス関数を用いて 1 番目の単語 y_1 を出力する。また、文ベクトルに出力した単語の情報を入れることで H_2 を作る。これを式 (2) 、(3) に従って再帰的に繰り返すことで、終端記号が出力されるまで y_n, H_n を順次求めていく。

$$y_n = \underset{i}{\operatorname{argmax}} \frac{\exp(V_i \cdot H_n + b_i)}{\sum_j \exp(V_j \cdot H_n + b_j)} \quad (2)$$

$$H_n = \begin{cases} W_3 h_m & (n=1) \\ \tanh(W_4 H_{n-1} + W_5 t_{n-1}) & (n > 1) \end{cases} \quad (3)$$

2.1.3 目的関数と学習

目的関数 J は出力の対数尤度とし、確率的勾配降下法によってパラメータを更新する。すなわち、ある入力文に対する出力文が $Y_1, Y_2, \cdots Y_n$ のとき、目的関数は式 (4) のようになる。

$$J = \sum_{1 \leq i \leq n} \log \left(\frac{\exp(V_{Y_i} \cdot H_i + b_{Y_i})}{\sum_y \exp(V_y \cdot H_n + b_y)} \right) \quad (4)$$

3 事前並べ替えと RNN-ED の同時学習

英仏翻訳において単語を逆順にして RNN に入力すると、対応する単語間の距離が近くなるため翻訳精度が向上する [8] など、RNN による翻訳において入出力の言語の語順関係は重要な要素である。英日翻訳などの文法や語順の大きく異なる言語間で同様に RNN による翻訳をする場合、語順が対応するよう適切に入力文の事前並べ替えを必要があると考えられる。

そこで、RNN による翻訳と合わせて翻訳器全体が単一のモデルとなるよう事前並び替えを対数線形モデルで構築する。事前並べ替えと RNN-ED を同時に効率よく学習することができ、翻訳する言語間に応じて並べ替えの最適化を行うことができる。このような事前並べ替え (Pre Ordering) を RNN-ED に組み込んだ翻訳モデルを PO-RNN-ED と呼ぶことにする。

翻訳元言語の単語列の並べ替えを考えるにあたり、全ての順列を考えるのは非効率である。英語と日本語の語順の違いについてしばしば取り上げられる例として、英語は SVO 型であるのに対し日本語は SOV 型であるという点がある。このように入れ替わりは句単位で起こるものであり、その生起確率は句の文中での役割やカテゴリ (名詞句・動詞句など) に依存すると仮定し、構文解析によって得られる構文木の上で並べ替えをモデル化する。

PO-RNN-ED は次の手順で並べ替えを行う (図 2)。

1. 翻訳する文を構文解析する。
2. 構文木の各ノードにおいて操作の生起確率を求める。
3. 操作の組み合わせで生じうる並べ替え候補とその生起確率を列挙する。
4. 並べ替え候補の重ね合わせから擬似単語ベクトル列を得る。
5. 擬似単語ベクトル列を RNN-ED に入力する。

これらの手順のうち本研究に大きく関わる手順 2〜4 について以下で詳細を述べる。

3.1 操作の生起確率の計算

並べ替えは句単位で起こると考え、構文木における各ノードでの操作の組み合わせで事前並べ替えができると仮定する。ここで、構文解析によって得られる構文木は二分木であるとする。今回の実験ではノードでの操作として「何もしない」・「子ノードの順序を反転させる」の 2 種類のみを扱うことにする。

構文情報から作成したノード n の素性ベクトル f_n から操作 i の生起スコアを求め、そのスコアをソフトマックス関数にかけて正規化することで、操作の生起確率 $p_{n,i}$ を求める (式 (5)) 。操作の生起確率は句のカテゴリや周辺の句との関係で決定するという仮定から、ノードの素性ベクトルとしては、そのノードのカテゴリ・子ノードのカテゴリ・子ノードのカテゴリの組み合わせ、の 4 種とした。

$$p_{n,i} = \frac{\exp(W_i f_n + b_i)}{\sum_{j \in operations} \exp(W_j f_n + b_j)} \quad (5)$$

3.2 並べ替え候補の生成

事前並べ替えを構文木の各ノードの操作の組み合わせによって行う。ある並べ替え r の生起確率 P_r は、式 (6) のように各ノードで行った操作の生起確率の積で表される。生起する可能性のある全ての並べ替えの中から、その生起確率の高いものを列挙する。

$$P_r = \prod_{n \in nodes} p_{n,i_n} \quad (6)$$

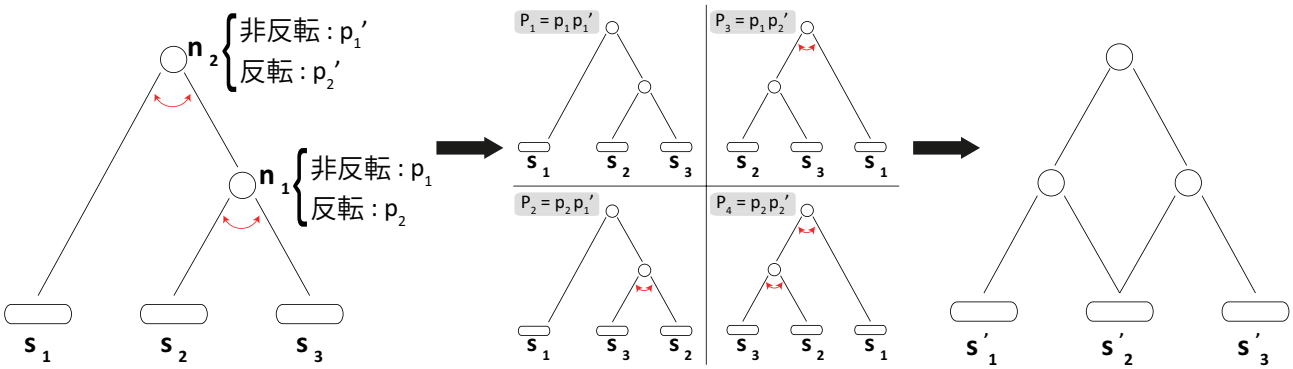

図 2: PO-RNN-ED

3.3 擬似単語ベクトル列の合成

列挙した並べ替え候補をその生起確率によって重ね合わせ、その結果を RNN-ED に入力して学習する。並べ替え候補を重ね合わせると単語のベクトルも重ね合わさり、特定の単語とは対応しないベクトルが生成される。これを擬似単語ベクトルと呼ぶことにする。

前節で挙げた並べ替え候補のうち、翻訳する文の単語ベクトル列 $s_1, s_2, \cdots s_m$ が $s'_{1,r}, s'_{2,r}, \cdots s'_{m,r}$ と並べ替えられる確率が P_r であるとき、RNN-ED に入力する擬似単語ベクトル列 $S_1, S_2, \cdots S_m$ はそれらの候補の重ね合わせで表される (式 (7)) 。このようにして得られた擬似単語ベクトル列を RNN に入れて翻訳を行う。

$$S_i = \sum_{r \in candidate} P_r s'_{i,r} \tag{7}$$

4 実験

英日翻訳において、比較的文の短く語彙の少ないコーパスで PO-RNN-ED を学習させた。学習のした結果、PO-RNN-ED の学習データに対する対数尤度が RNN-ED に比べて改善されたことを確かめた上で、事前並べ替えモデルに構文解析した英文を入力し、事前並べ替えを適切に行うことができるかどうかを確認した。

4.1 コーパス

今回構築した PO-RNN-ED では、未知語や長い文に対して頑健になるニューラルネットワークの処理 [6, 7] をしていない。また、ノードの個数の累乗に比例して並べ替えの候補数が増大していくため、学習の初期段階は短い文で行うことが望ましいと考えられる。それらの理由から比較的語彙が少なく文が長すぎないコーパスとして、中学英語例文集 [1] を利用した。コーパスの統計情報を表 1 に示した。なお、前処理としてアルファベットの小文字化と構文解析に含まれない記号 (“.” や “?”) は取り除いた。

表 1: 対訳コーパス

言語	文数	語彙	単語数
英語	960	951	7427
日本語	960	970	12061

英語の構文解析には Enju[2] を利用し、RNN-ED には節 2 のモデルを使用した。素性に利用するカテゴリは、名詞・動詞など基本的な 13 分類のみとした。また、日本語の形態素解析には KyTea[3] を利用した。

4.2 パラメータの学習

RNN-ED 中のベクトルの次元数は 1000 次元とし、学習用データを 100 回学習させた。パラメータの初期値としては、ソフトマックス関数に使われるものは 0 とし、その他は Glorot らの初期化方法 [4] に従い、$\pm\sqrt{6/(1000+1000)} \simeq \pm 0.0548$ 間の一様乱数とした。学習率のスケジューリングは AdaDelta [9] ($\rho = 0.97, \varepsilon = 10^{-4}$) を利用した。

5 実験結果・考察

学習データについて対数尤度を求めた結果、事前並べ替えを行うことで-56.7 から-17.8 に改善されていることが確かめられ、事前並べ替えが学習データの正確な表現に大きく貢献していると言える。今後の課題として、より大規模なデータについて精度が改善したかを実験を行っていく必要があると考えている。

学習した事前並べ替えモデルに構文解析した文を入力し、生起確率の高い候補と原文について生起確率を調べた。そのうち、英日翻訳における並び替えが適切に学習できている例を 2 つ示す。

[1] http://english-writing.mobi/

[2] http://www.nactem.ac.uk/enju/index.html

[3] http://www.phontron.com/kytea/index-ja.html

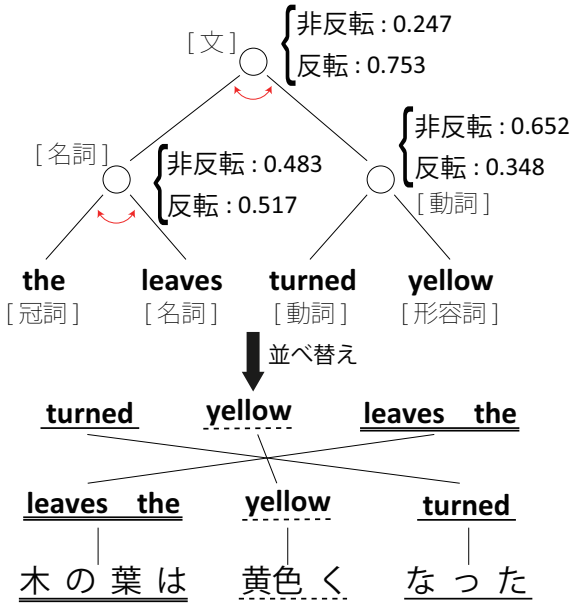

図 3: 並べ替えの生起例 1

5.1 例 1：“the leaves turned yellow”

“the leaves turned yellow” という英文を構文解析し、学習した事前並べ替えモデルにかけたところ、最も生起確率の高い並べ替えは “turned yellow leaves the” となった（図 3）。この結果から、日本語の並びと逆順になるように並べ替えが起こりやすくなっていることが分かる。これは並べ替えの起こりにくい西洋言語間においても逆順で RNN に入力すると精度が上がる、という Sutskever らの結果 [8] と一致する。

一方で、“turned yellow”（VC）に関しては、反転が起こりにくくなっていると学習されている。これは PO-RNN-ED が単純に英語を逆順に並び替えているのではなく、日本語との対応が逆順になるように並べ替えているからである。つまり、英語の VC が日本語では CV と入れ替わることを学習できていることになる。また、“the” と “leaves” は反転する確率としない確率がほぼ同じとなっている。これは、冠詞に対応する日本語がないため、反転の有無があまり翻訳に関係しないことを示していると考えられる。

以上から、PO-RNN-ED により英語と日本語の語順や文法の違いを自動で学習できていることが言える。

5.2 例 2：“i am having lunch now”

同様に、“i am having lunch now”（私は今昼食を食べているところです。）という英文の並べ替え候補を生成し、原文と生起確率の高かった 2 文を表 2 に示す。

この例でも PO-RNN-ED が日本語との対応が逆順になるように並べ替えを行っていることが分かる。すなわち、単純に逆順にするだけでなく、“having lunch”（VO）を “lunch having”（OV）と並び替えを行っており、日本語と英語の文法の違いを学習できていると言える。また、英語で現在進行形であることを表している “am” が文頭と文末の両方に現れているが、これは日本語の「は」と「です」に対応しているとみなすことができる。

表 2: 並べ替えの生起例 2

並べ替えられた文	生起確率
I am having lunch now	0.027
am having lunch now I	0.127
having lunch now am I	0.144

6 おわりに

本研究では、構文情報を用いた事前並べ替えと RNN-ED の同時学習が可能であることを示した。事前並べ替えができることにより、西洋言語間の翻訳で行われている RNN の様々な手法を、英日翻訳などの語順の違う言語間の翻訳でも適用することができるようになる。また、一方の言語の構文解析が可能であれば、提案手法を用いて事前並べ替えや文法の違いを学習することができると考えられる。

今後の課題としては、先行研究で行われている Long Short-Term Memory [5] や低頻度語の処理 [6] を RNN-ED に組み込み、より大規模なコーパスにおいて性能実験をすることを考えている。

参考文献

[1] J. Cai, M. Utiyama, E. Sumita, and Y. Zhang. Dependency-based pre-ordering for chinese-english machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2014.

[2] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on EMNLP*, 2014.

[3] J. Devlin, R. Zbib, Z. Huang, T. Lamar, R. Schwartz, and J. Makhoul. Fast and robust neural network joint models for statistical machine translation. In *52nd Annual Meeting of the ACL*, 2014.

[4] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics*, pp. 249–256, 2010.

[5] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[6] T. Luong, I. Sutskever, Q. V. Le, O. Vinyals, and W. Zaremba. Addressing the rare word problem in neural machine translation. *arXiv preprint arXiv:1410.8206*, 2014.

[7] J. Pouget-Abadie, D. Bahdanau, B. van Merriënboer, K. Cho, and Y. Bengio. Overcoming the curse of sentence length for neural machine translation using automatic segmentation. *arXiv preprint arXiv:1409.1257*, 2014.

[8] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pp. 3104–3112, 2014.

[9] M. D. Zeiler. Adadelta: An adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.