目的言語における係り受け構造を考慮したニューラル機械翻訳

江里口 瑛子[1], Kyunghyun Cho[2], and 鶴岡 慶雅[1]

[1] 東京大学 工学系研究科, {eriguchi, tsuruoka}@logos.t.u-tokyo.ac.jp
[2]New York University, kyunghyun.cho@nyu.edu

1 はじめに

ニューラル機械翻訳モデルは，単一のニューラルネットワークで記述された翻訳モデルである．従来の統計的機械翻訳手法に対して，比較的単純なモデル構造でありながら，その翻訳性能の高さから注目を集めている．また，句構造情報，係り受け情報，単語の品詞情報などの統語情報を導入したモデルも提案されており，翻訳性能の改善が報告されている [5, 9]．しかしながら，いずれの研究も原言語側における利用にとどまっており，目的言語側において 1 つのニューラルネットワークの翻訳モデルとして記述した研究はまだない．

一方，英語の係り受け解析タスクの近年の精度向上は目覚ましいものがある．単語の係り受け関係は shift-reduce のアクション列に置き換えることが可能であり，係り受け解析器は，スタックとバッファを利用した遷移型モデルとして記述することができる．この遷移型モデルに基づいて、feed-forward ネットワークを利用したモデル [2] や，系列に基づくリカレントニューラルネッワークを利用したモデル [3, 4] などが提案されている．このうち，Recurrent Neural Network Grammar (RNNG) [4] は，句構造解析に加えて，対象文を生成する文生成を兼ねた同時学習モデルとして提案され，英語の句構造解析タスク，ならびに，係り受け解析タスクにおいて，最高性能を報告している [1].

本研究では，係り受け解析器から得られた係り受け情報を，正確なデコードを誘導するための情報として利用する新たなニューラル機械翻訳モデルを提案する．翻訳タスクに加え，目的言語側における構文解析タスクを組み合わせたモデルには，すでにマルチタスク学習モデル [7] があり，複数のタスクを 1 つのニューラルネットワークで同時学習し，タスク間でパラメータを適切に共有することで，タスク全体の性能を改善できることが示されている．既存研究から，構文解析器などから得られる情報の活用や，異なるタスク間の情報の共有は，ニューラル機械翻訳モデルにおいても有用であることが示唆されており，それらを一つのモデルとしてより直接的に記述することで更なる性能改善が期待できる．提案手法では，解析器から得られた係り受け解析情報を正解とみなし，これにより翻訳の出力が誘導される．そして，提案手法内の係り受け解析モデルと翻訳モデルは，先のマルチタスクモデルと同様に，パラメータの一部をモデル間で共有するだけではなく，各モデルのネットワークがより密接に接続されている．日英翻訳コーパスを用いた翻訳実験によって，従来のニューラル機械翻訳モデルよりも提案モデルの BLEU スコアが 0.88 ポイント改善したことを確認した．

2 構文解析と文生成の同時学習

2.1 Recurrent Neural Network Grammar

Recurrent Neural Network Grammar は，句構造解析と文生成の同時学習モデルであり，単一のニューラルネットワークからなる．ここでは，句構造解析は shift-reduce 操作による遷移型モデルによって記述されており，単語はスタックとバッファへ移動 (shift) し，スタック中にある単語に対して適当なアクションを適用 (reduce) させていくことで，スタック上に文全体の句構造が構築されていく．RNNG では，さらに，shift アクション時に単語生成を行っており，文生成タスクとの同時学習モデルとなっている．

生成された文に対する全アクションはアクション列 $\boldsymbol{a} = (a_1, a_2, \cdots, a_L)$ からなり，ステップ t におけるアクション a_t の生成確率は，スタック，アクション，バッファの隠れ層 $\boldsymbol{h}_t^s$，$\boldsymbol{h}_t^a$，$\boldsymbol{h}_t^b \in \mathbb{R}^{n\times 1}$ それぞれから，

$$p(a_t|\boldsymbol{a}_{<t-1}) = softmax(\boldsymbol{W}_v \boldsymbol{u}_t + \boldsymbol{b}_v), \quad (1)$$

$$\boldsymbol{u}_t = \tanh(\boldsymbol{W}_u[\boldsymbol{h}_t^s;\ \boldsymbol{h}_{t-1}^a;\ \boldsymbol{h}_t^b] + \boldsymbol{b}_u), \quad (2)$$

と計算される．ここで，$\boldsymbol{W}_v \in \mathbb{R}^{|V_a|\times n}, \boldsymbol{W}_u \in \mathbb{R}^{n\times 3n}$ と $\boldsymbol{b}_v \in \mathbb{R}^{|V_a|\times 1}, \boldsymbol{b}_u \in \mathbb{R}^{n\times 1}$ は，それぞれ重み行列とバイアス項であり，; は行列同士の結合を表す．$|V_a|$ は，アクションの種類数を表す．$\boldsymbol{h}_t^s$，$\boldsymbol{h}_t^a$，$\boldsymbol{h}_t^b \in \mathbb{R}^{n\times 1}$

は，Stack LSTM [3] により，

$$\boldsymbol{h}_t^s = StackLSTM(\boldsymbol{h}_{top}^s, r_t), \quad (3)$$
$$\boldsymbol{h}_t^a = StackLSTM(\boldsymbol{h}_{t-1}^a, V_a(a_{t-1})), \quad (4)$$
$$\boldsymbol{h}_t^b = StackLSTM(\boldsymbol{h}_{t-1}^b, V_y(y_{t-1})), \quad (5)$$

と計算される．ここで，$\boldsymbol{h}_{top}^s$ はスタック上で TOP に指されているユニットである．Stack LSTM では，各 LSTM ユニットはスタック上で扱われ，TOP の指すユニットと入力を受け取ることで新たな LSTM ユニットの生成が行われる．また，$r_t \in \mathbb{R}^{m\times 1}, V_a(a_{t-1}) \in \mathbb{R}^{l\times 1}, V_y(y_{t-1}) \in \mathbb{R}^{m\times 1}$ は，それぞれ，スタックへの入力として与えられる単語ベクトル (或いは係り受け構造の句ベクトル)， ステップ $(t-1)$ のアクションベクトルと目的言語における単語ベクトルである．

3 提案手法

3.1 ニューラル機械翻訳

ニューラル機械翻訳モデルでは，リカレントニューラルネットワーク (RNN) を用いて入力文を固定長ベクトル空間にエンコードした後，別の RNN を用いてそのベクトル空間から出力文をデコードする．エンコードの際には，入力文の単語列 $\boldsymbol{x} = (x_1, x_2, \cdots, x_N)$ を文頭から順次受け取り，ステップ i における LSTM の前向き隠れ層 $\overrightarrow{\boldsymbol{h}}_i \in \mathbb{R}^{n\times 1}$ は，単語ベクトル $V_x(x_i)$ から $\overrightarrow{\boldsymbol{h}}_i = LSTM(\overrightarrow{\boldsymbol{h}}_{i-1}, V_x(x_i))$ として計算される．ここで $\overrightarrow{\boldsymbol{h}}_0 = \boldsymbol{0}$ とする．同様にして，文末から順次単語ベクトル列を受け取り，新たにステップ i における LSTM の後ろ向き隠れ層 $\overleftarrow{\boldsymbol{h}}_i \in \mathbb{R}^{n\times 1}$ を得る．最終的に，エンコーダにおけるステップ i の隠れ層 $\boldsymbol{h}_i \in \mathbb{R}^{2n\times 1}$ は $\boldsymbol{h}_i = [\overrightarrow{\boldsymbol{h}}_i; \overleftarrow{\boldsymbol{h}}_i]$ とし，両方向エンコーダを構成する．

デコードの際，ステップ j における単語の条件付き確率は，

$$p(y_j|\boldsymbol{y}_{<j}, \boldsymbol{x}) = \mathrm{softmax}(\boldsymbol{W}_s\tilde{\boldsymbol{s}}_j + \boldsymbol{b}_s), \quad (6)$$
$$\tilde{s}_j = \tanh(\boldsymbol{W}_c[\boldsymbol{s}_j;\ \boldsymbol{d}_j] + \boldsymbol{b}_c), \quad (7)$$

と算出される．ここで，$\boldsymbol{W}_s \in \mathbb{R}^{|V_t|\times n}, \boldsymbol{W}_c \in \mathbb{R}^{n\times 3n}$ は重み行列，$\boldsymbol{b}_s \in \mathbb{R}^{|V|\times 1}, \boldsymbol{b}_c \in \mathbb{R}^{n\times 1}$ はバイアス項である．$|V_t|$ は目的言語の語彙数を表す．隠れ層 $\tilde{\boldsymbol{s}}_j \in \mathbb{R}^{n\times 1}$ はステップ j における単語出力用の隠れ層であり，同ステップにおけるデコーダの LSTM の隠れ層 $\boldsymbol{s}_j \in \mathbb{R}^{n\times 1}$ と，アテンション機構によって得られた文脈ベクトル $\boldsymbol{d}_j \in \mathbb{R}^{2n\times 1}$ から得られる．LSTM の隠れ層 $\boldsymbol{s}_j$ は，直前の隠れ層 $\boldsymbol{s}_{j-1}$，出力された単語ベクトル $V_y(y_{j-1})$ (出力の単語列は $\boldsymbol{y} = (y_1, y_2, \cdots, y_M)$)，そして出力時に用いられた隠れ層 $\tilde{\boldsymbol{s}}_j$ から，$\boldsymbol{s}_j = LSTM(\boldsymbol{s}_{j-1}, [V_y(y_{j-1}); \tilde{\boldsymbol{s}}_{j-1}])$ として算出する．ここで使用する LSTM の計算では，エンコード時に使用した LSTM とは別のパラメータを用意する．また，アテンション機構では，ステップ j におけるデコーダの隠れ層とエンコーダの隠れ層間の関連度 $\boldsymbol{\alpha}_j(i)$ をソフトマックス関数を用いて確率分布の形で表す．この関連度を用いてエンコーダの隠れ層を重み付けし，その和を文脈ベクトル $\boldsymbol{d}_j \in \mathbb{R}^{2n\times 1}$ とする (式 (8), (9))．

$$\boldsymbol{d}_j = \sum_{i=1}^{N} \boldsymbol{\alpha}_j(i)\boldsymbol{h}_i, \quad (8)$$
$$\boldsymbol{\alpha}_j(i) = \frac{\exp(\boldsymbol{h}_i\boldsymbol{W}_a\boldsymbol{s}_j)}{\sum_k^M \exp(\boldsymbol{h}_k\boldsymbol{W}_a\boldsymbol{s}_j)}. \quad (9)$$

ここで，$\boldsymbol{W}_a \in \mathbb{R}^{n\times 2n}$ は，異なる隠れ層間の次元を同一にするための行列であり，$\boldsymbol{h}_i\boldsymbol{W}_a\boldsymbol{s}_j$ は両隠れ層間の類似度を表す．また，デコーダの隠れ層 $\boldsymbol{s}_0$ の初期化には，Tree-LSTM ユニットを用いて左右の子どもの隠れ層にエンコーダの最終隠れ層を受け取り，$\boldsymbol{s}_0 = TreeLSTM(\overrightarrow{\boldsymbol{h}}_N, \overleftarrow{\boldsymbol{h}}_N)$ として算出する．

3.2 機械翻訳と目的言語における係り受け構文解析の同時学習モデル

提案モデルでは，RNNG モデルにおけるバッファをニューラル機械翻訳モデルのデコーダそのものとみなし，単語の生成もまたニューラル機械翻訳モデルのデコーダに従うものとする．RNNG モデルのスタック，アクションの初期隠れ層 $\boldsymbol{h}_0^s, \boldsymbol{h}_0^a$ には，デコーダの隠れ層の初期化と同様に，両方向エンコーダの各最終隠れ層から Tree-LSTM ユニットを用いて $\boldsymbol{h}_0^s = TreeLSTM(\overrightarrow{\boldsymbol{h}}_N, \overleftarrow{\boldsymbol{h}}_N), \boldsymbol{h}_0^a = TreeLSTM(\overrightarrow{\boldsymbol{h}}_N, \overleftarrow{\boldsymbol{h}}_N)$ とし，それぞれ異なる TreeLSTM のパラメータを用意する．また，スタックで用いる単語ベクトルのパラメータは，デコーダの単語ベクトルを共有するものとする．予備実験では，スタックとデコーダ間で単語ベクトルのパラメータを共有することで翻訳性能が改善することを確認している．図 1 に提案モデルの概要図を示す．図中では，デコーダにおけるアテンション機構などは省略している．

スタックの Stack LSTM に入力として与える係り受け構造句のベクトル $r_t \in \mathbb{R}^{m\times 1}$ は，論文 [3] に従って，

$$r_t = \tanh(\boldsymbol{W}_r[r^d; r^p; V_a(a_{t-1})] + \boldsymbol{b}_r), \quad (10)$$

として算出する．ここで，$\boldsymbol{W}_r \in \mathbb{R}^{m\times(2m+l)}$ は重み行列，$\boldsymbol{b}_r \in \mathbb{R}^{m\times 1}$ はバイアス項である．また，$r^d \in \mathbb{R}^{m\times 1}$ は，reduce アクションによって生成される係り受け関係のうち，係り受け先の単語あるいは係り受け構造句のベクトルを表し，$r^p \in \mathbb{R}^{m\times 1}$ は係り受け元の単語あるい

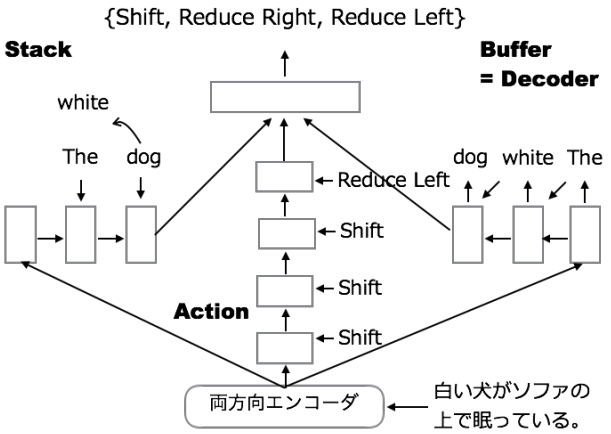

図 1: 提案モデル.

は係り受け構造句のベクトルを表す．$V_a(a_{t-1}) \in \mathbb{R}^{l \times 1}$ は，直前に選択された reduce アクションのアクションベクトルである．ただし，係り受け構造句が生成されておらず，単に単語のみが入力される場合は r_t はその入力単語のベクトルとする．

3.3 目的関数とパラメータ更新

提案モデルの目的関数は，学習に用いるパラレルコーパスのデータ $\mathcal{D}$ の入出力文の全ペアに関する対数尤度と，係り受け解析器を用いて得られた出力文に対するアクションの操作列 $\boldsymbol{a}$ に関する対数尤度からなり，

$$J(\boldsymbol{\theta}) = \frac{1}{|\mathcal{D}|} \sum_{(\boldsymbol{x},\boldsymbol{y}) \in \mathcal{D}} \log p(\boldsymbol{y}|\boldsymbol{x}) + \log p(\boldsymbol{a}|\boldsymbol{x}), \quad (11)$$

と定める．ここで $\boldsymbol{\theta}$ は，モデルに含まれる LSTM，Tree-LSTM，単語ベクトル，アテンション，隠れ層などの計算に用いられる全パラメータを表す．学習時には，パラメータ更新に確率的勾配降下法を用いる．

4 実験

4.1 実験設定

実験では，Asian Scientific Paper Excerpt Corpus (ASPEC)[1] の日英コーパスを使用した．学習データには，論文 [5] 中で使用されている小規模コーパス (10 万文対) を用いた．また，出力文の係り受け解析器には，SyntaxNet [2] を用い，英語の単語分割は SyntaxNet に従い，日本語の単語分割には KyTea [8] を用いた．使用した語彙は，学習データ中で出現回数が 2 回以上の単語とした．語彙サイズは，23,532 語 (日本語)，27,721 語 (英語) である．アクションの語彙サイズは，係り受けラベルを含め，79 種類である．学習したモデルの比較・評価には，開発データ (dev.txt; 1790 文対) を用いた．

[1] http://lotus.kuee.kyoto-u.ac.jp/ASPEC/

モデル	BLEU
ベースライン (NMT)	17.69
提案手法：+ RNNG	**18.57**
+ RNNG w/o スタック	17.64
+ RNNG w/o デコーダ	17.71
+ RNNG w/o アクション	17.83
+ Dropout ($rate = 0.2$)	**19.17**

表 1: 開発データにおける比較.

4.2 モデルパラメータの学習

モデル中のパラメータは，$[-0.1, 0.1]$ を範囲とする一様分布からの乱数に従って初期化を行った．LSTM 及び Tree-LSTM の忘却ゲートのバイアス項は 1.0 で初期化し，そのほかのバイアス項は $\mathbf{0}$ で初期化した．各パラメータの学習には，確率的勾配降下法 (学習率の初期値は 1.0) を用い，ミニバッチサイズを 128 とし，勾配ノルムは 3.0 でクリップした．開発データのパープレキシティが悪化した場合，前回のエポックで保存していたモデルを読み込み，学習率を 0.5 倍にして学習を再開するという学習率スケジュールを行った．各 LSTM，Tree-LSTM の隠れ層の次元，並びに原言語，目的言語の単語ベクトルの次元はそれぞれ 256 とし，アクションのベクトルは 128 とした．また，負例サンプリングに基づくソフトマックスの近似手法である BlackOut [6] を用いて，学習の高速化を行った．BlackOut のパラメータは，$K = 2000, \alpha = 0.4$ とした．

モデルの評価は，参照訳とモデルの出力した翻訳結果から算出した BLEU スコアを用いて行う．評価時は，提案モデルのデコーダに対してのみビーム探索を用いて翻訳を行い，RNNG モデルによるアクションの予測は行わなかった．用いたビーム幅は 20 である．

4.3 実験結果

表 1 に，開発データにおける各モデルの BLEU スコアを示す．ベースラインは，両方向エンコーダにアテンション機構を備えたニューラル機械翻訳モデルである．提案手法は，このベースラインに対して RNNG を導入したモデルである．ベースラインに対して提案手法は，BLEU スコアで 0.88 ポイントの改善を示した．また，RNNG を構成する，スタック，デコーダ，アクションに関するネットワーク構造をそれぞれ除いたモデルで評価したところ，いずれの場合も BLEU スコアは悪化した．最も下がり幅が大きかったのは，ス

入力文: 各種 トンネル 分光 技術 を 用い て 比較 研究 を 行 っ た .
参照訳: **The comparative research was carried out** using various tunneling spectroscopy technologies .
ベースラインによる翻訳文: **The comparison was compared** using the various tunneling spectroscopy .
提案手法による翻訳文: **The comparative study was carried out** using various tunneling spectroscopy .

表 2: 開発データにおける、各モデルの翻訳文とその比較.

タックを除いた場合であり，0.93 ポイント低下した．さらに、提案モデルの式 (2) および式 (7) の入力に対して Dropout ($rate = 0.2$) を適用したところ，更に 0.60 ポイントの改善が見られた．

表 2 に，ベースラインと提案手法それぞれの翻訳結果を記す．既存のニューラル機械翻訳 (ベースライン) では，原言語側の文中の「比較研究を行った」に該当する箇所は，“The comparison was compared” と翻訳された．ここでは「比較」という名詞 “comparison” と「比較する」という動詞の過去形 “compared” が生成され，意味の重複が起きている．また,「研究を行った」に該当する箇所の訳出に失敗している．これに対して，提案手法では，“The comparative study was carried out” という翻訳結果が得られた．ベースラインのような意味重複は起きておらず，また，訳出漏れがあった箇所について正しく翻訳が行なわれている．

4.4 考察

提案手法では，デコード時にニューラル機械翻訳モデルのデコーダのみを用いて訳出を行っており，係り受け情報を直接用いていないにもかかわらず，BLEU スコアの改善が見られた．ニューラル機械翻訳モデルでは， 一般に，単語のもつ意味の類義性などは学習前に与えられておらず，データから学習される．このため，学習データが小さい場合は，各単語ベクトルの類義性の獲得が不十分であり，表 2 で示したような，意味の重複した翻訳文が出力されたと考える．大量の学習データを用いることで上記問題点はある程度改善されると期待されるが，提案モデルのように，学習データから得られる係り受け関係を考慮し，一部のパラメータのモデル間共有を行うことで、モデルの性能改善につながったと考える．

また、RNNG モデルのスタック上では，アクションに従って，目的言語の単語間の係り受け句ベクトルが構成される．係り受け句ベクトルの構成要素である単語ベクトルは，この係り受け情報を直接的に考慮しながら学習される．RNNG モデルからスタックを除いた場合は，この係り受け情報を直接的に考慮できず，このため表 1 においても BLEU スコアが最も低くなったと推測する．

5 おわりに

本研究では，目的言語側における係り受け構造情報に着目し，翻訳モデルと，構文解析と文生成を行う RNNG モデルを合わせて，新たなニューラル機械翻訳モデルを提案した．そして，日英翻訳の実験を行い，既存のニューラル機械翻訳手法と比較し，提案手法の有効性を示した．提案モデルでは，解析器から得られる一意の離散的な係り受け情報のみを扱うため，これらをより抽象的に扱っていきたい．また、より大規模なデータを用いた実験と，他の言語においても同様の有効性の確認は，今後の課題としたい．

謝辞

本研究は JSPS 科研費 15J12597 の助成，JST，CREST の支援を受けたものです．

参考文献

[1] L. Kong C. Dyer G. Neubig N. A. Smith A. Kuncoro, M. Ballesteros. What do recurrent neural network grammars learn about syntax? In *Proceedings of the EACL*, 2017 (to appear).

[2] D. Andor, C. Alberti, D. Weiss, A. Severyn, A. Presta, K. Ganchev, S. Petrov, and M. Collins. Globally normalized transition-based neural networks. In *Proceedings of the ACL*, 2016.

[3] C. Dyer, M. Ballesteros, W. Ling, A. Matthews, and A. N. Smith. Transition-based dependency parsing with stack long short-term memory. In *Proceedings of the ACL and the IJCNLP*, 2015.

[4] C. Dyer, A. Kuncoro, M. Ballesteros, and A. N. Smith. Recurrent neural network grammars. In *Proceedings of the NAACL*, 2016.

[5] A. Eriguchi, K. Hashimoto, and Y. Tsuruoka. Tree-to-sequence attentional neural machine translation. In *Proceedings of the ACL*, 2016.

[6] S. Ji, S. V. N. Vishwanathan, N. Satish, M. J. Anderson, and P. Dubey. Blackout: Speeding up recurrent neural network language models with very large vocabularies. *Proceedings of ICLR*, 2015.

[7] T. Luong, Q. V. Le, I. Sutskever, O. Vinyals, and L. Kaiser. Multi-task sequence to sequence learning. *Proceedings of ICLR*, 2015.

[8] G Neubig, Y. Nakata, and S. Mori. Pointwise prediction for robust, adaptable japanese morphological analysis. In *Proceedings of the ACL*, 2011.

[9] R. Sennrich and B. Haddow. Linguistic input features improve neural machine translation. In *Proceedings of the 1st Conference on MT*, 2016.