双方向モデルの連結によるニューラル機械翻訳

和田 崇史　能地 宏　須藤 克仁　松本 裕治
奈良先端科学技術大学院大学 情報科学研究科
{wada.takashi.wp7, noji, sudoh, matsu}@is.naist.jp

1 はじめに

近年, ニューラルネットを用いた様々な翻訳モデルが提案されてきた. 中でも, 最もメジャーなモデルがatttention 機構付きの Encoder-Decoder モデル [1, 7] である. これは, 翻訳を行う文の単語を Encoder に入力し, Decoder では Encoder の各隠れ層を attnention 機構で参照しながら翻訳文の生成を行うというモデルである. Encoder-Decoder には通常 LSTM が用いられることが多いが, 近年そのほかにも CNN を用いたモデル [3] や transformer と呼ばれる特殊な構造をしたモデル [12] も新たに提案されてきた.

そして, これらの Encoder-Decoder 翻訳モデルで共通しているのは, 翻訳文を文頭から文末に, もしくは文末から文頭へと逐次生成をするように学習を行う点である. しかし, テスト時においては Decoder の毎ステップで正しい単語が生成されるとは限らず, 誤った単語が次の Deocder の状態に入力されることがある. それ故, 前半の生成のエラーが後半へと伝播していくと考えられ, もし予測困難な単語が文頭または文末にあった場合, 既存の翻訳モデルでは文全体の翻訳の質が悪くなると懸念される. この問題を解決するため, 本研究では翻訳文を文頭と文末の双方向から生成を行い, それらを文の中心で結合させる双方向連結翻訳モデルを提案する. 左右両方向モデルの前半部分の生成をお互いに組み合わせることで文の生成エラーが後半部分へと伝搬するのを防ぎ, それにより既存の双方向 reranking の手法よりも高い精度を出すことを可能とした. なお, 本提案手法は学習するモデルの構造に依存せず, 全ての seqeunce to sequence のモデルに適用可能であり, 汎用性の高い手法である.

2 関連研究

既存の Encoder-Decoder 翻訳モデルでは通常, 翻訳文を文頭から順々に生成を行なっているが, 文末から生成するモデルと組み合わせることで翻訳の精度が良くなることが知られている. 例えば, [6] では双方向のモデルでそれぞれ k-best の翻訳文を beam search で生成し, 合計 $2k$ の候補文からベストな翻訳文を選択することで, 単方向のみの翻訳文よりも大幅に精度が改善することを示している. 文の選択基準となるスコアは, 双方向のモデルそれぞれがソース文 x から翻訳文 y を生成する確率 p(y|x) の積であり, 以下の式で表される.

$$\hat{y} = \underset{y_i}{\operatorname{argmax}}\ \mathrm{p}(\mathrm{y_i}|\mathrm{x};\, \theta_{\,\mathrm{fwd}}) \times \mathrm{p}(\mathrm{r}(\mathrm{y_i})|\mathrm{x};\, \theta_{\,\mathrm{bkw}})$$
$$\mathrm{p}(\mathrm{y}|\mathrm{x},\, \theta\,) = \prod_{\mathrm{t}=1}^{\mathrm{N}} \mathrm{p}(\mathrm{w_t}|\mathrm{w_{<t}}, \mathrm{x},\, \theta\,) \tag{1}$$

ここで, y_i は i 番目の候補文 , $r(y_i)$ が y_i の単語のオーダを逆さにした文, $\theta_{\,fwd}$ と $\theta_{\,bkw}$ はそれぞれ文頭から文末に生成するモデルと文末から文頭に生成するモデルのパラメーターである. また, w_t は y の t 番目の単語, N は y の単語数を意味する.

3 提案手法

3.1 概要

本研究が提案する双方向連結翻訳モデルとは, 文頭から文末（左から右）へ生成する翻訳モデル (l2r モデル) と文末から文頭（右から左）へ生成する翻訳モデル (r2l モデル) をそれぞれ学習し, それらの出力を文の中心部分で結合させるモデルである. 文を左右両方向から生成することで, 生成の前半でのエラーが後半に伝搬する度合いを軽減することが出来る. しかし, 双方向の翻訳文を文の中心で結合するためには, 文の中心が一体どこになるのかが問題となる. そこで, 本研究では文の中心を文の主辞（ルート）であると考え, l2r モデルと r2l モデルが翻訳文を生成する際にターゲット言語の主辞がどこなのかを予測するようにそれぞれ学習を行う. 例えば, “In addition , they showed the effects of the application.” という翻訳文を生成する時, “showed” がこの文の主辞であることを明示するために <ROOT>というトークンを単語の

後に加えた “In addition , they showed <ROOT> the effects of the application.” という文を生成するように学習を行う．なお, 学習データの翻訳文の主辞は事前に CoreNLP [8] の stanford parser [2] を用いて求め, <ROOT>トークンを文に挿入する．

3.2 翻訳文の結合

<ROOT>トークンを含んだ翻訳文を生成するように l2r モデルと r2l モデルの学習を行った後, テスト時においてこれら 2 つのモデルの出力を結合させることを行う．まず, l2r モデルが左から順々に beam search をしながら生成を行い, <ROOT>トークンが出たら生成を終了する．例えば, “In addition , they showed <ROOT> ”のような翻訳文を beam 幅の数 k だけ出力する．そして同様に, r2l モデルは右から左へと文の生成を行い, <ROOT>トークンの次の単語が出力された時点で生成を止める．したがって, r2l モデルでは例えば “. application the of effects the <ROOT> showed” のような翻訳文が k 文出力される．

そして, 左右両方向から独立に生成された文の前半と後半を結合し, 新たな文を完成させる．結合する文のペアは, l2r モデルと r2l モデルが出力した主辞の単語が同じとなったものとする．従って, 翻訳文の数が最大となるのは, 双方向モデルの生成文の主辞が全て同じ単語となった場合で候補文数は k^2, 反対に最小となるのは前後で主辞が一致するペアが一つもない場合で 0 ある．

3.3 翻訳文の選択

3.2 の手法で最大 k^2 の翻訳文の候補が得られた後, l2r 及び rl2 モデルそれぞれが最後まで生成した文を候補文に加える．従って, 翻訳文の候補数は最大で $k^2 + 2k$, 最小で $2k$ となる．そして, 各文のスコアを算出し最もスコアが高い文を翻訳文とした．スコアの算出には [6] と同様に双方向の文の生成確率を用いるが, 結合により生成した翻訳文の中には非常に短い文も含まれており, 式 (1) をそのまま用いると $\mathrm{p}(\mathrm{y}|\mathrm{x}, \theta)$ の値が大きくなる短い文が多く選択されてしまう問題がある．そこで, 以下の式でスコアを算出する．

$$\hat{y} = \underset{y_i}{\operatorname{argmax}} \frac{\log(\mathrm{p}(\mathrm{y_i}|\mathrm{x};\, \theta_{\,\mathrm{fwd}})) + \log(\mathrm{p}(\mathrm{y_i}^{\mathrm{r}}|\mathrm{x};\, \theta_{\,\mathrm{bkw}}))}{N} \tag{2}$$

ここで, N は文の単語数である．なお, 参考のため, 候補文を l2r 及び rl2 モデルの出力 $2k$ 文に絞った場合と, 結合による最大 k^2 文の出力のみに絞った場合の結果も比較する．なお, k^2 文の中に前後で主辞が一致するペアが一つもない場合は, l2r モデルの翻訳文の 1 ベストを用いる．

	10	20	30	40
l2r	21.28	21.20	21.03	20.99
rl2	20.97	20.86	20.81	20.73
reranking l2r and rl2	22.83	22.91	23.08	23.11
reranking mix	22.81	**23.09**	23.13	**23.17**
reranking all	**23.05**	23.01	**23.15**	23.09

表 1: ビーム幅を変えた時の BLEU スコア

	前半 5 単語	後半 5 単語
l2r	19.47	17.54
r2l	17.91	18.10
reranking lr2 and r2l	19.67	19.17
reranking mix	19.48	18.74
reranking all	**19.97**	**19.44**

表 2: 文の前半と後半 5 単語の BLEU スコア．beam 幅は 20

4 実験

4.1 データセット

本研究の学習と評価に用いたデータは日英対訳コーパスの Asian Scientific Paper Excerpt Corpus (ASPEC) [9] である． なお, 学習用データ約 300 万文のうち，文アライメントの類似度上位 100 万文を用いた． また，英文は CoreNLP の PTBtokenizer を用いて tokenize し，和文を MeCab (バージョン 0.996，IPADIC)[1]で形態素解析を行う．そして, 最後に Moses[5] の toolkit[2]を用いて lowercase, cleaning を行い, 1 文で 50 単語を超える文を取り除いた．学習, 開発, テストに用いた文数はそれぞれ 932007, 1790, 1812 である．評価には case insensitive BLEU[10] を用い, 翻訳文を detokenize した後に moses toolkit にある multi-bleu-detok.perl を用いてスコアを算出した．

4.2 モデルと学習

本研究が用いた翻訳モデルは Attention 機構付き Encoder-Decoder[7] である． Encoder には Bi-LSTM[11] を用い, 両方向の Encoder の各隠れ層を結合した．単語の分散表現, 及び Encoder と Decoder の隠れ層の次元数は 512, 層の数は 1, また学習には

[1] http://taku910.github.io/mecab
[2] https://github.com/moses-smt/mosesDecoder

入力	その結果，中枢側群，末梢側群ともに 3 D画像が元画像より劣っていた例は半数に及んだ.	BLEU	reranking score
ref	as the result, in the half of the total cases, the 3d image was inferior to the original image both in the central group and peripheral group.	–	–
l2r + r2l	as the result, the case in which the 3d image **was inferior to the original image was inferior to the case in which the 3d image was inferior to the original image.**	0.31	**-0.79**
mix	the result showed that the 3d image was inferior to the original image in both the central group and the peripheral side group.	**0.42**	-0.94

表 3: reranking によるスコアと BLEU の大小が一致しない例

Adam[4] を用いた. 最大 epoch 数は 13 とし, 開発データの loss の最小値が 3 回連続で更新されなかった場合はそこで学習を終了させた. 語彙は学習データで 5 回以上用いられた単語に制限し, 総単語数は日本語と英語それぞれ 46k 及び 64k となった. また, decoder で beam search を行う際も式 (2) と同様に生成確率の log を文長で割ることで, 短すぎる文が生成されるのを防いだ.

5 結果

表 1 が示しているのが評価データにおける各手法の BLEU である. 以下, ビーム幅を k とする. l2r 及び rl2 は各方向のモデルでそれぞれ生成した時の結果で, reranking l2r and rl2 というのは双方向の計 $2k$ の候補文を reranking した時の結果である. また, reranking mix は 3.2 の手法で生成される最大 k^2 の候補文を reranking したもので, reranking all は最大 k^2+2k の全ての候補文を reranking したものである. いずれのビーム幅においても本提案手法である reranking all が一番高い結果となっており, 本提案手法は単純にビーム幅を増やすこと以上の効果があることが明らかとなった.

また表 2 が示しているのは, 文の前半と後半 5 単語を抽出して計算した BLUE スコアである. なお, スコアの算出対象文はレファレンス文が 10 単語以上で, かつ全ての手法で翻訳結果が 10 単語以上となった文である. 本提案の 1 番のモチベーションは文の生成エラーが後半へと伝搬することを防ぐことであるため, これらの数値が他の手法より高いと本提案手法が効果的であることが示される. 表 2 が示す通り, 本提案手法がもっとも良い結果となっており, 本提案手法の有効性が示された. また, また, 文の前半 5 単語では l2r モデルの方が rl2 モデルよりも優位なのに対し, 後半 5 単語では rl2 モデルの方がスコア高くなっている. このことから, 実際に生成の後半部分の精度は前半よりも悪くなることが示された. 本提案手法では, 双方向のモデルを文の中心で結合することにより両者の得意な生成部分を組み合わせ, 翻訳の質を高めることに成功した.

	10	20	30	40
l2r	26.50	28.65	30.00	31.00
rl2	26.77	29.11	30.35	31.14
reranking l2r and rl2	30.83	33.60	34.09	35.25
reranking mix	28.35	31.19	32.56	33.81
reranking all	**31.74**	**34.52**	**35.38**	**36.51**

表 4: 候補文からベストな文を選択した時の, 各 beam 幅における BLEU スコア.

6 分析

本研究では, 複数の候補文から式 (2) を用いて reranking を行い, 最終的に出力する文の選択を行なっているが, 必ずしも最良の文を選択出来ているとは限らない. 事実, 表 1 の結果を見ると候補文が最も多い reranking all の結果が reranking mix のみの場合よりも悪くなる場合がある. そこで, 候補文の中から実際に BLEU が最も高くなる文 (oracle) を選択出来たとした場合, 各手法の精度がどれほど改善するのかを調べた. その結果が表 4 である. 双方向モデルが生成した文に本提案手法の生成文を候補文に加えた reranking all が最も高い精度となっていることがわかる. また, 表 1 の結果と比べても, 他の手法とのスコアの差が大きくなっており, それゆえ reranking の手法によっては更なる改善が得られる可能性を示唆している. 例えば, 表 3 は reranking のスコアの大小と, 実際の BLEU スコアの大小が一致しない文の例を示したものである. l2r + r2l によって生成された文は同じフレーズの繰り

返しを含んでおり, それゆえ BLEU も mix の文よりも低くなっているが, reranking のスコアでは前者の方が高い値となっている. これは, reranking のスコアが双方向モデルの生成確率に依存しており, それゆえ各モデルがそれぞれ生成した文のスコアが結合で生成した文よりも高くなりやすいためだと考えられる. 従って, 候補文の中から最適な文を選択するために別の評価基準を設けると, より正確な文選択が行える可能性がある.

7 おわりに

本研究では双方向翻訳モデルを文の中心で結合して新たな翻訳文を生成することで, 翻訳の精度が既存の手法よりも改善することを示した. 本研究の提案する手法はデータの種類やモデルの構造に依存しないため, 汎用性が高い手法と言える. 今後, reranking のスコア算出の方法を変えることで, より正確な翻訳文を数多くの候補文の中から選択できるかを調べていきたい.

参考文献

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, 2014.

[2] Danqi Chen and Christopher Manning. A fast and accurate dependency parser using neural networks. In *Proc. of EMNLP*, 2014.

[3] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolutional sequence to sequence learning. *CoRR*, 2017.

[4] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, 2014.

[5] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *Proc. of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, 2007.

[6] Lemao Liu, Andrew Finch, Masao Utiyama, and Eiichiro Sumita. Agreement on target-bidirectional lstms for sequence-to-sequence learning. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, 2016.

[7] Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *Proc. of EMNLP*, 2015.

[8] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, 2014.

[9] 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*, 2016.

[10] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, 2002.

[11] M. Schuster and K.K. Paliwal. Bidirectional recurrent neural networks. *Trans. Sig. Proc.*, 1997.

[12] 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 *Advances in Neural Information Processing Systems 30*. 2017.