非自己回帰デコーディング型ニューラル機械翻訳の改善

Improvement of Neural Machine Translation that Uses Non-autoregressive Decoding

国立研究開発法人情報通信研究機構　先進的音声翻訳研究開発推進センター　主任研究員
今村　賢治

2004 年奈良先端科学技術大学院大学情報科学研究科情報処理学専攻博士課程修了。1985 年日本電信電話株式会社。2014 年株式会社 ATR-Trek より NICT に出向。機械翻訳の研究に従事。

1 はじめに

本稿では、非自己回帰（non-autoregressive）デコーディングを使ったニューラル機械翻訳器の改善を試みる。本稿で対象とする非自己回帰方式は、Mask-Predict[1] である。Mask-Predict 方式では、全トークンを同時並列に生成するために、まず、翻訳文の長さを予測する。そして、デコーダーに予測するした数の <mask> トークンを入力し、それらを翻訳結果になるように文を復元する。

訓練の際は、翻訳文のトークン予測と長さ予測を同時に学習する。しかし、Mask-Predict 方式は予測対象のトークン数がバッチごとに異なる。そのため、トークン予測の損失と長さ予測の損失の割合を適切に制御しないと、どちらかの予測が過学習する。本稿では、学習時にトークン予測と長さ予測の損失を適切に混合し、翻訳精度の改善を行う。

以下、２節で Mask-Predict 方式の概要を簡単に説明し、３節で損失関数を修正する。４節で実験を行い、５節でまとめる。

2 Mask-Predict 方式

非自己回帰デコーディングは、Transformer[2] の並列性を利用し、翻訳文のすべてのトークンを同時に生成する[3]、[4]、[1]。

Mask-Predict 方式は、BERT[5] でも用いられているマスク言語モデル（masked language models）によって、入力された <mask> トークンが何であったのか、復元するように翻訳文を生成する（図１）。初期入力は全トークンが <mask> である。学習時には、<mask> トークンだけ損失を計算し、パラメータを更

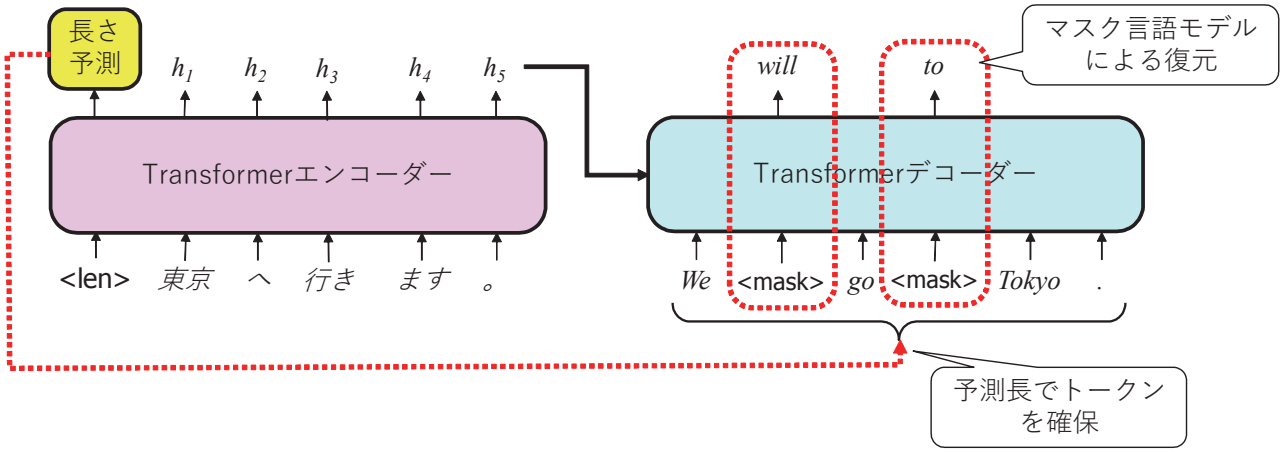

図１　Mask-Predict の動作例

新する。

トークンを並列に生成するため、あらかじめ翻訳文の長さをエンコーダー出力から推定する。これは入力文の先頭に <len> トークンを付与し、それに対するエンコーダー出力を長さ埋込として分類器を動作させ、長さを推定する。

いくつかの非自己回帰デコーディングは、１回の生成では十分な翻訳品質を確保できないため、デコーディングを繰り返し適用することで品質を向上させている（たとえば[4]）。Mask-Predict も最低４回、できれば 10 回程度の繰り返し適用をすることで、翻訳品質を向上させているが、自己回帰デコーディングの性能には届いていない。

3 Mask-Predict 方式の改善

２節で述べたとおり、Mask-Predict 方式は翻訳文の長さ推定とマスクトークンの推定を同時に行う。つまり学習時は長さ予測モデルとマスク言語モデルの学習を同時に行っている。

$$\theta' = \mathrm{Update}\left(\theta, \frac{L_{\mathrm{mask}} + L_{\mathrm{len}}}{N_{\mathrm{mask}}}\right) \quad \cdots\cdots (1)$$

ただし、Update 関数は、最適化器（本稿では Adam 最適化を使用）のパラメータ更新関数で、θ、θ′はそれぞれ更新前後のパラメータセット、L_{mask}、L_{len} はそれぞれマスク言語モデルの損失、長さ予測モデルの損失、N_{mask} はミニバッチ内のマスクトークン数である。

マスク言語モデルは、さまざまなマスク率で元トークンを推定する必要があるため、マスク数を変化させながら学習している。そのため、式(1)のN_{mask} はミニバッチ毎に著しく変化する。しかし、多くの場合はミニバッチに含まれる文数（長さ予測数 N_{len} と同じ）は比較的一定である。そのため、式(1)のように損失を N_{mask} で割っただけではパラメータの勾配がミニバッチ毎に変化してしまい、学習が適切に行われない。結果的に、どちらかのモデルが過学習となる。

そこで本稿では、式(1)を長さ損失の要素数（N_{len}）も考慮し、以下のように修正する。

$$\theta' = \mathrm{Update}\left(\theta, \frac{L_{\mathrm{mask}} + L_{\mathrm{len}}}{N_{\mathrm{mask}} + N_{\mathrm{len}}}\right) \quad \cdots\cdots (2)$$

このように修正することにより、マスク言語モデルと長さ予測モデルの勾配が適切に配分される。この修正は基本的にはバグ修正であるが、翻訳品質は向上する。

4 実験

4.1 実験設定

コーパスは ASPEC[6] の日英翻訳を使用した。これには約 300 万文の対訳を含み、dev, devtest, test の３種類のヘルドアウトセットを含んでいる。

使用したシステムは、GitHub 公開されている Mask-Predict である[1]。また、この Mask-Predict は fairseq 翻訳システムを改良したものなので、参考までに fairseq[7][2] の Transformer モデル（自己回帰デコーディング）とも比較する。

今回は、Transformer Base モデル（６レイヤー、８ヘッド、モデル 512 次元、FFN は 2,048 次元）相当のものを学習した。ただし、データの知識蒸留（knowledge distillation）は行っていない。

Mask-Predict 方式は、非自己回帰デコーディングを繰り返し適用することで、翻訳品質を上げることができる。本稿では、10 回繰り返しを行った。また、テスト時のビーム幅は Mask-Predict 方式は 5、通常の Transformer は 10 を使用した。Mask-Predict 方式は、一つの長さに対して一つの翻訳仮説しか生成しないため、ビーム幅は長さの予測数である。

4.2 実験結果

まず、訓練時におけるパープレキシティの推移を図２に示す。このグラフは、改善前後のマスク言語モデルと長さ予測モデルのエポックごとの開発セットパープレキシティの推移を示したものである。この結果では、長さ予測モデルについては、改善前後の違いがほとんどないが、マスク言語モデルに関しては、パープレキシティが低下している。これは、モデルの精度が向上していることを示している。

表１は、改善前後の BLEU スコアである。参考までに、fairseq での Transformer ベースモデルのスコアも示す。赤字は、Mask-Predict（改善前）に比べて、

1 https://github.com/facebookresearch/Mask-Predict 2020 年５月時点。

2 https://github.com/pytorch/fairseq

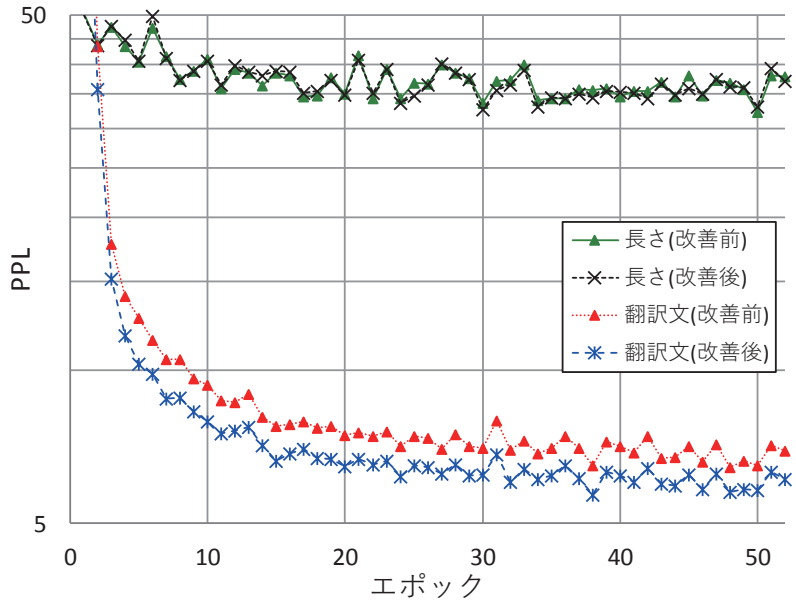

図2　マスク言語モデルと長さ予測モデルの開発セットパープレキシティ (PPL) の推移

表1　方式別 BLEU スコア

方式	BLEU		
	dev	devtest	test
Mask-Predict（改善前）	25.24	23.56	24.82
Mask-Predict（改善後）	25.28	24.13	25.08
Transformer ベースモデル	28.76	27.43	28.91

スコアが有意に向上していることを示す（$p < 0.05$）。Mask-Predict は、Trasnformer ベースモデルに比べ翻訳品質は劣る。Mask-Predict の改善前と改善後の BLEU を比較すると、有意差があるのは devtest セットのみなので、確実とは言えないが、改善後の BLEU スコアの方が高い傾向がある。

複数の損失を混合して同時学習する場合、勾配も適切に分配しないと、最終的なモデルの精度や翻訳品質が低下することを示している。

5 まとめ

本稿では、Mask-Predict 方式に焦点をあて、学習時の勾配計算の改善を行った。基本的にはバグ修正に位置づけられるが、モデルが適切に学習されることを示した。

謝辞

本件は、総務省の「ICT 重点技術の研究開発プロジェクト（JPMI00316）」における「多言語翻訳技術の高度化に関する研究開発」による委託を受けて実施した研究開発による成果です。

参考文献

[1] Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Mask-predict: Parallel decoding of conditional masked language models. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 6112–6121, Hong Kong, China, November.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. CoRR, abs/1706.03762.

[3] Jiatao Gu, James Bradbury, Caiming Xiong, Victor O. K. Li, and Richard Socher. 2017. Non-autoregressive neural machine translation. CoRR, abs/1711.02281.

[4] Jason Lee, Elman Mansimov, and Kyunghyun Cho. 2018. Deterministic non-autoregressive neural sequence modeling

by iterative refinement. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 1173–1182, Brussels, Belgium, October-November.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota, June.

[6] Toshiaki Nakazawa, Manabu Yaguchi, Kiyotaka Uchimoto, Masao Utiyama, Eiichiro Sumita, Sadao Kurohashi, and Hitoshi Isahara. 2016. Aspec: Asian scientific paper excerpt corpus. In Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2016), pages 2204–2208, Portoroˇz, Slovenia, May.

[7] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations), pages 48–53, Minneapolis, Minnesota, June.