

# 漸進的アプローチでの編集量を制御するニューラル機械翻訳

美野秀弥 衣川和堯 後藤功雄 山田一郎

NHK 放送技術研究所

{mino.h-gq, kinugawa.k-jg, goto.i-es, yamada.i-hy}@nhk.or.jp

## 概要

言語生成タスクで現在主流となっているモデルは最終結果のみ観察可能で、生成結果に対して修正を加えながら最終結果を出力する漸進的なモデルではない。本稿では生成タスクの1つである機械翻訳に焦点を当て、人間のコンテンツ生成のプロセスを参考にした漸進的アプローチによる制御を考慮したニューラル機械翻訳の手法を提案する。提案手法は途中結果を確認できる他、途中の翻訳結果に対して編集前後の編集量に応じた制御パラメータを与えることで生成結果を制御することができる。翻訳実験の結果、提案手法の翻訳精度は既存手法と比較して有意な差はなかったが、データごとに適切な制御パラメータを与えると有意に翻訳精度が向上した。

## 1 はじめに

ニューラル機械翻訳をはじめとしたテキスト生成タスクの手法は1度に正解を出力するモデルが主流であり、生成結果を段階的に編集しながら正解を出力することを想定していない。しかし、公開資料作成などのコンテンツ生成時、人間は一度にコンテンツを完成させるのではなく、段階的にコンテンツを確認、編集しながら完成させることが多い。例えば翻訳品質を保証するための要求事項を定めた国際規格であるISO17100では、翻訳、セルフチェック、バイリンガルチェック、最終検品などのプロセスが定められており、図1のように段階的な作業が必須となっている。そこで、本稿では生成タスクの1つである機械翻訳に焦点を当て、途中の生成結果を確認でき、かつ制御可能なニューラル機械翻訳の手法を提案する。制御は編集前後の編集量とする。提案手法は、翻訳元の文に加えて編集前の機械翻訳結果と制御パラメータとを入力して編集後の翻訳結果を出力するモデルであり、モデルが出力した翻訳結果と制御パラメータとを繰り返し同じモデルに入力することで最終的な翻訳結果を得る。制御パラメータに

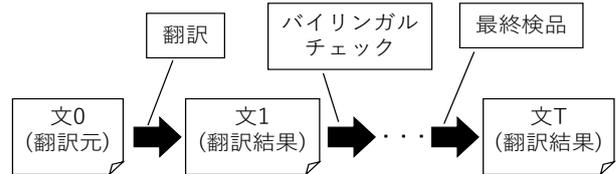


図1 漸進的な翻訳作業の例。

は編集前と編集後との翻訳結果間の編集距離をベースにしたスコアを用いる。人間のコンテンツ生成のプロセスを参考にした漸進的なアプローチを用いることで途中結果を確認できるだけでなく、途中の翻訳結果に対してどの程度編集すべきか編集距離に応じたスコアを与えることで編集量を制御することができる。科学技術論文から抜粋された日英対訳コーパスであるASPECを用いた日英・英日翻訳実験を行った結果、提案手法は既存のトランスフォーマーベースのモデルと比較して翻訳精度に有意な差はなかった。しかし、モデルが出力した機械翻訳結果に対してデータごとに適切な制御パラメータを与えることで翻訳精度が向上することを確認した。

## 2 提案手法

### 2.1 漸進的アプローチを用いたニューラル機械翻訳モデル

本稿では、Reidら[1]の手法を参考にして漸進的アプローチを用いたニューラル機械翻訳を提案する。図2に提案手法が用いるモデルの概要図を示す。翻訳元の文を $x$ とし、編集ステップごとの出力を $y_1, \dots, y_t$ とし、最終出力である翻訳結果を $y_T$ とする。この一連の編集過程を以下のようにモデル化して学習する。

$$P(y_T) = \prod_{t=1}^{T-1} p_{\theta}(y_{t+1}|y_t, x) \quad (1)$$

$\theta$ は学習パラメータである。 $y_1$ は編集前のデータが存在しないため特殊なトークン([NULL])を用いる。 $p_{\theta}(y_{t+1}|y_t, x)$ は以下の式で計算する。

$$p_{\theta}(y_{t+1}|y_t, x) = p_{\theta}^{gen}(y_{t+1}|y_t, x, e_t) \quad (2)$$

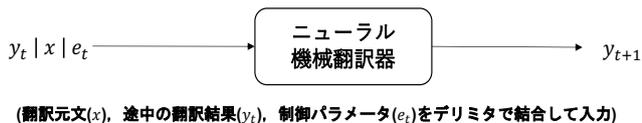


図2 提案手法のモデルの概要図.

$e_t$  は制御パラメータであり,  $y_{t+1}$  と  $y_t$  との間のトークンを単位とした編集距離<sup>1)</sup>を用いる.  $t=1$  の場合は編集前のデータがない特殊なケースと見なし, 制御パラメータも特殊な値 (999) を用いる.

翻訳時は, 式 1 に, 翻訳元データと途中の翻訳結果と制御パラメータとを併せて入力する. ステップ数  $T$ , および制御パラメータ  $e_t$  は, 開発データを用いた事前実験により固定する.

## 2.2 学習データの構築方法

提案手法のモデルを学習するためには, (翻訳元データ, 編集前の翻訳結果のデータ, 編集後の翻訳結果のデータ) の 3 つ組のデータが必要となる<sup>2)</sup>. 一般的な対訳データは (翻訳元データ  $s$ , 最終的な翻訳結果のデータ  $t$ ) から構成されており, 途中の翻訳結果のデータを入手することができないため, 新たに構築する必要がある. 人手による翻訳, 編集作業結果を用いることが理想だが, 膨大なコストがかかるため現実的ではない. そこで, 精度の異なる複数の機械翻訳器を用い, 下記の手順により既存の対訳データ ( $s, t$ ) から擬似的な編集前後の翻訳結果  $u$  を含むデータ対 ( $(s, u), t$ ) を構築する.

1. 機械翻訳器 A により翻訳元文  $s$  の機械翻訳結果  $u_A$  を獲得する.
2. 機械翻訳器 B により翻訳元文  $s$  の機械翻訳結果  $u_B$  を獲得する.
3.  $((s, u_A), t)$ ,  $((s, u_B), t)$  を学習データに追加する.
4. 参照訳  $t$  と機械翻訳  $u_A$  の編集距離  $e_{t-u_A}$ , 参照訳  $t$  と機械翻訳  $u_B$  の編集距離  $e_{t-u_B}$  を計算し,  $e_{t-u_A} < e_{t-u_B}$  の場合は  $((s, u_B), u_A)$  を,  $e_{t-u_A} > e_{t-u_B}$  の場合は  $((s, u_A), u_B)$  を学習データに追加する.

精度の異なる複数の機械翻訳器は, ビームサーチやアンサンブルモデルなどのデコード時の工夫や, 学習データを減らして学習することなどにより構築可能である. しかし, ベースの機械翻訳器はビーム幅を最大値にしたアンサンブルモデルであるため, 新

1) 文字単位で編集距離を算出するレーベンシュタイン距離を応用した.

2) 制御パラメータは編集前後のデータから算出できる.

表 1 データセットの概要 (ASPEC).

	文数	トークン数		タイプ数	
		日本語	英語	日本語	英語
学習	2,000,000	64.8M	53.4M	295K	653K
開発	1,790	54K	45K	6.0K	7.3K
テスト	1,812	54K	45K	6.2K	7.4K

たに構築する機械翻訳器はベースの機械翻訳器と比較して精度が低いものとなり, ベースの機械翻訳器の翻訳結果よりも精度の高い翻訳結果を含む学習データを構築できない. すなわち, デコード時の工夫のみで構築した機械翻訳器で作る学習データ内の編集後のデータは, 参照訳を除くと, ベースの機械翻訳器による翻訳結果よりも高い翻訳精度のデータとならない. 高い翻訳精度を出すためには, 編集前のデータにベースの機械翻訳器の翻訳精度以上の翻訳結果を入力し, 編集後のデータとして編集前よりも高い翻訳精度の結果を出力する必要があり, この条件を満たす学習データの構築が必要となる. そこで, 目的言語側のデータを特権情報として入力時に与える語彙制約付きニューラル機械翻訳器 [2] を用いる. 語彙制約付きニューラル機械翻訳器は, 目的言語側の参照訳から複数の単語を翻訳元文と併せて入力することで翻訳結果に重要な単語が含まれやすくなる. 学習時, 翻訳時ともに, 出力される単語を入力する必要があるため, 特殊な条件が必要だが, ベースとなる機械翻訳器よりも精度の高い機械翻訳器となっており, 目的に合致する. 本稿では, 美野ら [3] の "Proper-noun constraint", "Mistranslated-word constraint", "Both constraints", "Random-word constraint" の語彙制約付きニューラル機械翻訳で学習データを構築する.

## 3 実験

### 3.1 実験設定

提案手法の効果を確認するため, 科学技術論文コーパス ASPEC [4] を用いた日英・英日翻訳実験を行った. 本稿では ASPEC 中の対訳として対応度の高い 200 万文を用いた. 表 1 に本稿で用いるコーパスの統計データを示す. 英語のデータのクリーニング, および単語分割には Moses toolkit<sup>3)</sup> の clean-corpus-n.perl<sup>4)</sup>, tokenizer.perl<sup>5)</sup> を用いた. 日

3) <https://github.com/moses-smt/mosesdecoder>

4) <https://github.com/moses-smt/mosesdecoder/blob/master/scripts/training/clean-corpus-n.perl>

5) <https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/tokenizer.perl>

表2 実験結果 (BLEU).

	編集ステップ数 $t$			
	日英			英日
	1	2	3	1
従来手法	27.6	-	-	45.3
ノイズ付与	23.6	24.1	24.6	42.0
提案手法	27.2	27.7	27.9	45.0
提案手法+初期値修正	27.6	27.9	-	45.3

本語の形態素解析には, KyTea [5] を利用した. 各タスクで語彙数を制限するため, バイトペア符号化 (byte pair encoding) を用いたサブワード [6] を用いた. 語彙数は 46,000 とし, 頻度が 35 以下の語彙はサブワードに分割した. ニューラル機械翻訳には, Transformer モデル [7] を使い, Sockeye3 [8] で実装した. 本稿のモデルの学習には Adam [9] を optimizer として使い, 学習率は 0.0002 とした. バッチ内のデータは 5000 トークン以内になるように構築した. 翻訳時の各エンコーダへの入力の最大長は設定しなかった. 翻訳時は, ビームサーチを用い, ビーム幅を 10 とした. ハイパーパラメータであるステップ数  $T$  と制御パラメータ  $e_t$  は, 開発データを用いた事前実験<sup>6)</sup>により, 日英については ( $T = 3, e_2 = 5, e_3 = 3$ ), 英日については ( $T = 1$ ) となった<sup>7)</sup>. 翻訳の評価には一般的に用いられている自動評価尺度 BLEU [10] を用いた. BLEU スコアの計算には SacreBLEU<sup>8)</sup> [11] を使い, case-insensitive BLEU-4 の結果で評価した. また, サンプル数を 10000 に設定した paired-bootstrap.py<sup>9)</sup> を使い, BLEU スコアにおける提案手法と既存手法との有意差を調べた.

### 3.2 比較手法

本稿では, 下記の 2 つの手法を比較手法とした. 1 つ目は, 漸進的アプローチを用いない手法 (従来手法) である. 2 つ目は, 漸進的アプローチを用いた手法として, 拡散モデルをベースにした, 翻訳結果にノイズを混入してデータを構築する手法 [1] (ノイズ付与) である. 2 つ目の比較手法は, 学習データの構築方法を比較するための手法であり, 翻訳結果に挿入, 削除ノイズを加えることで, ノイズを加えたデータを編集前, ノイズを加える前のデータを編集後のデータとして (翻訳元データ, 編集前の翻

6) 事前実験はステップ数  $T$  の値を 1-30 にして開発データの翻訳を行い, 最も精度の高かった値を  $T$  とした.

7) 制御パラメータ  $e_t$  は特殊トークン 999 で固定.

8) <https://github.com/mjpost/sacrebleu>

9) <https://github.com/neubig/utl-scripts/blob/master/paired-bootstrap.py>

表3 データごとに制御パラメータを与えた翻訳結果.

	日英	英日
	従来手法	27.6
提案手法 (制御パラメータ付与)	33.8	50.4

訳結果のデータ, 編集後の翻訳結果のデータ) の学習データを構築する.

### 3.3 実験結果

表 2 に実験結果を示す. 有意差検定を行った結果従来手法と比較して有意な差は確認できなかったが, 日英翻訳タスクについては従来手法よりも BLEU スコアが高くなった. 英日では, 開発データを用いた事前実験の結果ハイパーパラメータであるステップ数  $T = 1$  となってしまう, 最初のステップ ( $t = 1$ ) による翻訳のみが行われ, 翻訳結果に対する編集は行われなかった.

参照訳にランダムにノイズを含ませることで学習データを構築する手法 (ノイズ付与) は翻訳精度が有意に低下した. この手法は低コストで大量の学習データを構築可能だが, 翻訳時には入力される可能性のない学習データが大量に含まれるおそれがある. これらのデータが学習時のノイズとなり精度が低下したと考えられる.

#### 初期ステップに従来手法の結果を入力するモデル

英日タスクにおいて提案手法の BLEU スコアが低下した原因として, 最初のステップ ( $t = 1$ ) で翻訳 ( $x_1$  の生成) を行い 2 回目以降のステップで編集を行っており, 翻訳と編集の 2 つのタスクを取り扱うモデルとなっているために翻訳のみに特化した従来手法よりも性能が低下したと考えられる. そこで,  $x_0$  に [NULL] ではなく, 従来手法の結果を用いた場合のモデル (提案手法+初期値修正) で実験を行った. 表 2 より, 提案手法と同様に, 従来手法と比較して有意に精度が向上することはなかったが, 日英では BLEU スコアが向上したことに加え, 英日では従来手法の結果をそのまま出力することで翻訳精度の低下を防ぐモデルとなった.

#### データごとに適切な制御パラメータを与えた場合

提案手法は, 翻訳時の制御パラメータはあらかじめ開発データをもとにタスクごとに固定値を与えた. しかし, 実際には適切な制御パラメータはデータごとに異なるはずである. そこで, 文ごとの適切な制御パラメータが分かっていると仮定し, 文ごとに参照訳との編集距離を算出してそれをオラクル値として与えた場合の結果を表 3 に示す. その結果,

表 4 大きな制御パラメータ値を与えた翻訳結果.

#1	翻訳元 参照訳 編集前 編集後	Details of dose rate of “Fugen Power Plant” can be calculated by using DERS software. DERS ソフトウェアを用いて「ふげん発電所」の線量率を詳細に計算できる。 「ふげん」の線量率の詳細は DERS ソフトウェアを用いて計算できる。 DERS ソフトウェアを用いることにより、「ふげん」の線量率の詳細を計算することができる。
#2	翻訳元 参照訳 編集前 編集後	As a result, it was found that its minimum detection strain was 10-5. 最小検出歪みは 10-5 であった。 その結果, 最小検出歪みは 10-5 であることが分かった。 その結果, 最小検出歪みは 10-5 であることが分かったため, この方法によって得られたデータについて報告した。
#3	翻訳元 参照訳 編集前 編集後	Through the detailed analysis of the intensity dependence of excited light, this is judged to be due to photoionization 励起光強度依存性の詳細解析により, 光イオン化によるものと判断した。 励起光の強度依存性を詳細に解析し, これを光イオン化によるものと判断した。 これは光イオン化によるものである。

日英, 英日ともに, 従来手法と比較して有意に精度が向上した. 提案手法は文ごとに制御パラメータを推定するようなモデルとはなっていないが, 制御パラメータを推定する機構を組み込むことで精度向上が期待できる.

#### 制御パラメータによる編集制御

提案手法は, 開発データを用いて制御パラメータを設定した結果小さい値となったが, 制御パラメータを大きくすることで正確性や流暢性のある程度維持しつつ, バリエーションのある翻訳結果を生成できる可能性がある. そこで, 英日タスクで大きな制御パラメータ値を与えてテストセットを翻訳した. 表 4 に例を示す. #1 の例では, 編集前後で文の構造が変わり, 編集後の翻訳結果が編集前と比較して参照訳の構造に近づいていることが分かる. #2 の例では, 編集前の翻訳結果の方が参照訳に近く, 編集後の翻訳結果の「この方法によって得られたデータについて報告した」の部分が過剰訳となっていることが分かる. 翻訳元にはない情報が過剰訳となることで意味が変わってしまい #3 の例では, #2 の例とは異なり, 編集後の翻訳結果に訳抜けが生じている. 「励起光強度依存性の詳細解析により」の訳抜けについては意味が変わることはないが, 「判断した」の訳抜けは意味が変わってしまい問題である. #1 の例のように, 正確性が損なわれずに文の構造を変えることができればよいが, #2 や #3 の例のように意味が変わる例もある. 翻訳結果のバリエーションが増やせない単純な文について過剰訳や訳抜けになる傾向があると考えられる. 今後, 人手評価を行い, より詳細な分析を行いたい.

## 4 関連研究

ニューラル機械翻訳には, トランスフォーマーベースの自己回帰的なモデル [7] に加え, 非自己回

帰的なモデル [12] も提案されている. 非自己回帰的なモデルの 1 つである Levenshtein Transformer [12] はトークンを並列に予測するプロセスを繰り返すことで翻訳を行う. このモデルは, 並列計算が可能となり高速な翻訳を実現するだけでなく, 反復ステップにより自動ポストエディットなどのタスクにも有効であることを示したが, 翻訳精度は自己回帰的なモデルを下回る.

Reid ら [1, 13] は, 自己回帰モデルを用いて段階的に編集を行うテキスト生成モデルを提案し [13], それを拡張して拡散モデルベースの生成モデル DIFFUSER を提案した [1]. 提案手法は Reid ら [1] の手法を参考に行っているが, 参照訳にノイズを付与して対訳データを作成してノイズを除去するように学習する拡散モデルを用いず, 順方向の機械翻訳器を用いて対訳データを作成している.

## 5 おわりに

本稿では, 翻訳元の文と機械翻訳結果とを入力して新たな機械翻訳結果を出力する, 漸進的なアプローチを用いたニューラル機械翻訳を提案してその効果を検証した. 翻訳実験の結果, 提案手法が既存手法と同等の精度を出すことを確認した. また, 入力する機械翻訳結果の品質評価を適切に行うことができれば, 既存手法よりも精度が向上することを確認した. 今後は, 文ごとに適切な制御パラメータを推定できるような品質評価の機構を組み込んだモデルを考案・実装したい. また, 本稿では, 学習データの構築時に, 学習データの目的言語文と機械翻訳結果との編集距離をもとにした制御パラメータを用いたが, 編集距離を挿入, 削除, 置換に細分化した制御パラメータを用いることでより細かな制御が可能となると考えられる. 適切な制御パラメータを検討して効果を確認していきたい.

## 謝辞

本研究成果は、国立研究開発法人情報通信研究機構の委託研究（課題 225）により得られたものです。

## 参考文献

- [1] Machel Reid, Vincent J. Hellendoorn, and Graham Neubig. Diffuser: Discrete diffusion via edit-based reconstruction, 2022. arXiv preprint arXiv:2210.16886.
- [2] Guanhua Chen, Yun Chen, Yong Wang, and Victor O.K. Li. Lexical-constraint-aware neural machine translation via data augmentation. In **Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence**, pp. 3587–3593. International Joint Conferences on Artificial Intelligence Organization, 7 2020. Main track.
- [3] Hideya Mino, Kazutaka Kinugawa, Hitoshi Ito, Isao Goto, Ichiro Yamada, and Takenobu Tokunaga. NHK’s lexically-constrained neural machine translation at WAT 2021. In **Proceedings of the 8th Workshop on Asian Translation (WAT2021)**, pp. 46–52, Online, August 2021. Association for Computational Linguistics.
- [4] Toshiaki Nakazawa, Manabu Yaguchi, Kiyotaka Uchi-moto, Masao Utiyama, Eiichiro Sumita, Sadao Kurohashi, and Hitoshi Isahara. ASPEC: Asian scientific paper excerpt corpus. In **Proceedings of the Tenth International Conference on Language Resources and Evaluation**, pp. 2204–2208, Portorož, Slovenia, May 2016. European Language Resources Association.
- [5] Graham Neubig, Yosuke Nakata, and Shinsuke Mori. Pointwise prediction for robust, adaptable Japanese morphological analysis. In **Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies**, pp. 529–533, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.
- [6] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In **Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics.
- [7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In **Advances in Neural Information Processing Systems 30**, pp. 5998–6008. Curran Associates, Inc., 2017.
- [8] Felix Hieber, Michael Denkowski, Tobias Domhan, Barbara Darques Barros, Celina Dong Ye, Xing Niu, Cuong Hoang, Ke Tran, Benjamin Hsu, Maria Nadejde, Surafel Lakew, Prashant Mathur, Anna Currey, and Marcello Federico. Sockeye 3: Fast neural machine translation with pytorch, 2022. arXiv preprint arXiv:2207.05851.
- [9] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In **3rd International Conference on Learning Representations, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings**, 2015.
- [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 of the Association for Computational Linguistics**, pp. 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.
- [11] Matt Post. A call for clarity in reporting BLEU scores. In **Proceedings of the Third Conference on Machine Translation: Research Papers**, pp. 186–191, Brussels, Belgium, October 2018. Association for Computational Linguistics.
- [12] Jiatao Gu, Changhan Wang, and Junbo Zhao. Levenshtein transformer. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, editors, **Advances in Neural Information Processing Systems**, Vol. 32. Curran Associates, Inc., 2019.
- [13] Machel Reid and Graham Neubig. Learning to model editing processes. In **Conference on Empirical Methods in Natural Language Processing (EMNLP)**, Abu Dhabi, UAE, December 2022.