Positional Encoding出力長制御を用いた英日ニューラル機械翻訳の検討

岡 佑依　帖佐 克己　須藤 克仁　中村 哲
奈良先端科学技術大学院大学
{oka.yui.ov2, k-chousa, sudoh, s-nakamura}@is.naist.jp

1　はじめに

近年，ニューラルネットを用いた機械翻訳（NMT）の手法が多く考案されている．これらの手法において，アテンション機構を用いたエンコーダ・デコーダモデルは自然性があり高い精度の翻訳結果を残している．特に，Transformer[4] は Self-Attention，Multi-Head Attention，Positional Encoding という独自の機構を利用して，高い精度の翻訳結果を残した．

しかし，これらの Attention を用いた NMT の手法には，モデルが同じ箇所を複数回訳出する重複訳問題，まだ訳出していない箇所があるにも関わらず翻訳を終了する訳抜け問題がある．Attention を用いたエンコーダ・デコーダモデルは出力長を考慮する機構が備わっていないため，このような問題を完全に避けることは難しい．

Transformer を用いた要約文生成の研究において，高瀬ら [3] は Positional Encoding に出力長の情報を与えることで出力長の長さを制御した．この手法では，終端までの残り長さを明に与える二種類の Positional Encoding が提案された．この手法は訓練データに出現していない出力長の場合でも所望の長さを出力することを可能とした．

そこで本研究では要約文生成における二種類の Positional Encoding を機械翻訳に適用する．入力長から予測した長さを Positional Encoding に入力し，出力長を制御することで重複訳問題，訳抜け問題を解決が可能になるかどうかの検討を行った．

2　Positional Encoding を用いた出力長制御

Positional Encoding は，Transformer のエンコーダ・デコーダ両者において各埋め込み表現に対し，その位置に対応した絶対的な値を足し合わせることで位置情報を与える役割を持つ．その時足し合わせる値は正弦関数と余弦関数の式で表される．トークンの位置を pos，埋め込み表現の次元数を d とすると i 番目の次元の埋め込み表現に足し合わせる PE は以下のようになる．このとき，偶数次元は正弦関数，奇数次元は余弦関数で定義される．

$$PE_{(pos,2i)} = sin\left(\frac{pos}{10000^{\frac{2i}{d}}}\right) \tag{1}$$

$$PE_{(pos,2i+1)} = cos\left(\frac{pos}{10000^{\frac{2i}{d}}}\right) \tag{2}$$

高瀬ら [3] は デコーダ側の Positional Encoding の式に所望の出力長の値を組み込んだ．これにより，文生成時に所望の出力長までの残りのトークン数を考慮することが可能である．提案された式は終端までの比率に応じた $LRPE$(length-ratio positional encoding)，終端までの差に応じた $LDPE$(length-difference positional encoding) の 2 種類があり，それぞれ以下の式で表される．

$$LRPE_{(pos,len,2i)} = sin\left(\frac{pos}{len^{\frac{2i}{d}}}\right) \tag{3}$$

$$LRPE_{(pos,len,2i+1)} = cos\left(\frac{pos}{len^{\frac{2i}{d}}}\right) \tag{4}$$

$$LDPE_{(pos,len,2i)} = sin\left(\frac{len-pos}{10000^{\frac{2i}{d}}}\right) \tag{5}$$

$$LDPE_{(pos,len,2i+1)} = cos\left(\frac{len-pos}{10000^{\frac{2i}{d}}}\right) \tag{6}$$

len は所望の出力長を表す．$LRPE$，$LDPE$ の値はベースラインの Transformer の Positional Encoding と同じように各埋め込み表現に足し合わせる．また，エンコーダ側にはベースラインの Transformer と同様の Positional Encoding の式が適用される．

3　提案手法

本研究では，LRPE と LDPE を翻訳に適用する．要約生成では，LRPE と LDPE に入力する len の値は

デコーダが出力する要約文の所望の長さである固定値であったが，翻訳では入力文によってデコーダが出力するべき長さが異なることを考慮する必要がある．この len に入力文から予測された長さを入力することで，適切な出力長制御を行う．この予測された出力長が正解であるとき，長さを考慮せず出力をするモデルより良い精度の翻訳文が出力されることが期待される．また，長さを考慮していることから，重複訳問題，訳抜け問題が解決されることが期待される．

本稿ではこのアプローチの有効性を検証することを目的として，正解トークン長に基づく出力長制御を行う．入力文からの出力長予測を含む検証については今後の課題とする．

4 実験

4.1 実験設定

学習時，テスト時の正解トークン長に対し制約を変化させ出力長制御を行ったとき，翻訳結果がどのように変化するのかを目的とし，実験を行った． 本研究では，ソース文を英語，ターゲット文を日本語とした英日翻訳をタスクとした．データセットには，パラレルコーパス ASPEC を用いた．ASPEC[1] は 1,783,817 文対の学習データ，1,790 文対の開発データ，1,812 文対のテストデータからなり，今回学習には 1,000,000 文対の学習データである train-1.txt のみを使用した．英語及び日本語の入出力はサブワードとし，Sentencepiece[1] を使いトークナイズを行った．このとき，語彙サイズは 16,000 とし，言語間で共有した．

Transformer の実装には primitiv[2] を用いた．Transformer のデコーダ・エンコーダ層は簡単化のためともに 2 とし，それ以外のハイパーパラメータは [4] と同じにした．提案手法のハイパーパラメータも同様である．

高瀬ら [3] の研究では，len にターゲット文の文字数 ($character$) を与えていたが，本研究ではターゲット文のトークン長を与える．データセットでも述べたように，このトークン長は Sentencepiece でトークナイズされたときのものである．

ベースラインは学習時，テスト時に同じ Positional Encoding の式 (1),(2) を適用する．それに対し，本研究では，学習時に LRPE または LDPE に学習データの正解トークン長 (tgt_token_len) を入力し，テスト時に実際のテストデータの正解トークン長 (ref_token_len) を入力した場合のそれぞれを検証した．また，テスト時に分散が 2 となるようランダムな値を与える場合，短めに出力するようトークン長を 0.9 倍に制限した場合，長めに出力するようトークン長を 1.1 倍に制限した場合についても検証した．これらはモデルが訳語選択をする際の揺れに対しての堅牢性を上げることを目的とした． さらに，学習時にランダムな出力長制約を与える場合についても検証した．

評価手法には機械翻訳の自動評価として一般的な BLEU[2] を用い，Unidic[3] に基づく MeCab[4] で分かち書きした形態素列に対して multi-bleu.perl[5] で計算した．

また，出力長の精度を表すために，高瀬ら [3] が用いた分散 (var) の式を以下のように用いた．

$$var = \frac{1}{n}\sum_{i=1}^{n}|l_i - ref_i|^2 \tag{7}$$

ref_i はテストデータの正解トークン長，l_i は生成された翻訳文のトークン長である．

4.2 実験結果

表 1 に各モデルの BLEU と分散を示す．

4.2.1 BLEU による評価

Transformer + $LRPE$ のモデルにおいて，学習時の len に正解トークン長 (tgt_token_len) を与え学習したとき，ベースラインより BLEU は下がった．しかし，正解トークン長 (tgt_tokeb_len) +ランダムな値を与え学習したとき，ベースラインのモデルより高い BLEU41.56 が得られた．Transformer+$LDPE$ のモデルにおいても同様に，学習時の len に正解トークン長 (tgt_token_len) を与え学習したとき，ベースラインよりスコアは下がった．$LRPE$ と $LDPE$ で比較すると，$LRPE$ の方が BLEU 値は全体的に高いことがわかる．また，$ref\_token\_len \times$ 0.9 以外の実験ではベースラインより高い BP が得られたが，BLEU スコアの向上に繋がらなかった． 正解トークン長 (tgt_tokeb_len) + ランダムな値を与え学習したモデルが最も高い BLEU

[1] https://github.com/google/sentencepiece
[2] https://github.com/primitiv/primitiv
[3] https://unidic.ninjal.ac.jp/
[4] http://taku910.github.io/mecab/
[5] https://github.com/moses-smt/mosesdecoder

表 1: 実験結果

モデル	学習時の len	テスト時の len	BLEU	BP	分散
Transformer + PE (ベースライン)			41.21	0.926	22.312
Transformer + $LRPE$	tgt_token_len	ref_token_len	39.79	0.992	**0.167**
		$ref\_token\_len \pm 0$〜2	39.22	0.992	2.086
		$ref\_token\_len \times$ 0.9	38.53	0.926	5.992
		$ref\_token\_len \times$ 1.1	36.27	1.000	9.726
	$tgt\_token\_len \pm 0$〜2	ref_token_len	**41.56**	0.978	**3.986**
Transformer + $LDPE$	tgt_token_len	ref_token_len	39.32	0.992	**0.001**
		$ref\_token\_len \pm 0$〜2	38.97	0.992	2.042
		$ref\_token\_len \times$ 0.9	37.96	0.925	5.757
		$ref\_token\_len \times$ 1.1	35.91	1.000	10.455

を得られた理由として，重複訳，訳抜けの改善があげられる．これらについては 4.2.3 で詳しく述べる．

4.2.2 出力長の分散における評価

Transformer + $LRPE$ のモデルにおいて，学習時に tgt_token_len を与えたとき，分散値は 22.312 から 0.167 とベースラインより小さくなったが，BLEU も下がった結果となった．Transformer + $LDPE$ のモデルにおいても同様，分散値は 0.001 へと小さくなり，BLEU も下がった．しかし，$LRPE$ より $LDPE$ の方が分散は小さく，トークン長制御に関しては $LDPE$ の方が優れていることがわかる．これは，高瀬ら [3] の実験でも $LDPE$ の方が分散が小さいことが示されており，同様の結果が得られていることがわかる．

$tgt\_token\_len \pm 0$〜2 を与えるモデルは分散 3.986 であるがベースラインより BLEU は上がった．

4.2.3 訳抜け，重複訳

表 2 に訳抜け，重複訳が解決されている例を示す．提案手法 $LDPE$，$LRPE$ は，学習時に tgt_token_len，テスト時に ref_token_len を入力したモデルを示す．訳抜けが解決している例では，ベースラインに比べ，提案手法が正しいトークン長を得た時，訳抜けしていた部分が補われていることがわかる．次に，重複訳が解決している例では，$LRPE$ を適用したとき，重複訳の問題が解決していることがわかる．しかし，逆に訳抜け，重複訳が発生した例も数件見受けられた．学習データに含まれていない単語を出力する必要があるとき，ベースラインのように長さ制御をしていないモデルは短く出力することで翻訳文の精度を維持するが，長さを制御する提案手法では，長さが与えられていることから重複して単語を出力する．このようなことが原因で，重複訳が起きてしまう．この問題の改善も今後の課題としてあげられる．

5 おわりに

本稿では，要約に用いられる LDPE，LRPE モデルにトークン長を与えるニューラル機械翻訳モデルを提案した．実験から，正確でないトークン長を与えた時と比べ，正確なトークン長を与えた時，BLEU が向上することがわかった，また，学習時 len にランダムな値を与えることで，BLEU は向上する．提案手法により，訳抜け，重複訳が解決する例も確認したが，いくつかの例では訳抜け，重複訳が逆に発生した．

今後の課題として，新たに発生した訳抜け，重複訳の改善，また学習時 len にランダムな値を与える手法の分析があげられる．さらに，予測モデルの提案，予測モデルによって予測されたトークン長をテスト時に与えたときの評価・分析もあげられる．

謝辞

本研究の一部は JSPS 科研費 JP17H06101 の助成を受けたものである．

表 2: 提案手法における訳抜け，重複訳の解決例と発生例．Reference*となっている参照訳は原文と意味がより合致するように改変を行った.【】内はトークン数を表す.

訳抜けが解決している例

Source	These data can be converted by a program and analyzed by personal computer .
Reference*	これらのデータは, 変換プログラムで変換でき，パソコンで分析できる。
Baseline	これらのデータはパソコンで変換できる。【8】
LDPE	これらのデータを**プログラムで変換し**, パソコンで解析することが可能である。【14】
LRPE	これらのデータを**プログラムで変換し**, パーソナルコンピュータで解析できる。【14】
LRPE ±0〜2	これらのデータを**プログラムで変換し**, パーソナルコンピュータで解析できる。【13】

重複訳が解決している例

Source	On a test, it was possible to analyze features of city water , pure water , city water treated by magnetized device by changing frequency.
Reference*	実験では, 水道水, 純水, 及び磁化デバイスで処理した水道水の特性を周波数を変化させて分析することができた。
Baseline	試験では, 周波数を変えて磁化装置で処理した水, 純水,**水, 水**などの特性を解析することができた。【23】
LDPE	試験では, 周波数を変えて水, 純水, 都市水, 水, 磁化装置で処理する特徴を分析した。【25】
LRPE	試験では, 周波数を変えて磁化した装置で処理した水, 純水, 水の特徴を分析することもできた。【25】
LRPE ±0〜2	試験では, 周波数を変えて磁化デバイスで処理した水, 純水, 水道水の特徴を解析することができた。【23】

重複訳が発生している例

Source	Here was proposed a real - time phase stamp range finder (RT-PSRF) using three - phase correlation image sensor (3PCIS).
Reference	3 相相関イメージセンサー (three − phase correlation image sensor:3PCIS) を使った実時間の位相スタンプ・レンジファインダー (phase − stamp range finder:PSRF) を提案する。
Baseline	3 相相関イメージセンサ (3PCIS) を用いた リアルタイム位相スタンプ距離計 (RT- PSRF) を提案した。【25】
LDPE	三次元位相相関イメージセンサ (3PCIS) を用いた 実時間位相スタンプレンジファインダ (RT- PSRF）を提案し, その動作作理と**特徴, 特徴, 特徴, 特徴, 特徴**などを紹介した。【44】
LRPE	3 相相関イメージセンサ (3PCIS) を用いた 実時間位相スタンプ距離計 (RT- PSRF) を提案し, その**原理, 原理, 原理, 原理, 原理, および原理, 原理**, 特性について述べた。【45】
LRPE ±0〜2	3 相相関イメージセンサ (3PCIS) を用いた 実時間位相スタンプレンジファインダ (RT- PSRF) を提案し, **その有効性を検証するために,3 相相関イメージセンサ (3PCIS) を用いた。** 【45】

参考文献

[1] Toshiaki Nakazawa, Manabu Yaguchi, Kiyotaka Uchimoto, Masao Utiyama, Eiichiro Sumita, Sadao Kurohashi, and Hitoshi Isahara. Aspec: Asian scientific paper excerpt corpus. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2016)*, pp. 2204–2208, Portorož, Slovenia, may 2016. European Language Resources Association (ELRA).

[2] 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.

[3] Sho Takase and Naoaki Okazaki. Positional encoding to control output sequence length. 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)*, pp. 3999–4004, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

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