球体表面を利用した位置符号化

岡 佑依 田中 貴秋 平尾 努 永田 昌明
NTT コミュニケーション科学基礎研究所
yui.oka.vf@hco.ntt.co.jp

概要

系列変換モデルを使った文生成では，原文の構文構造を活用することで性能が向上することが報告されている．Transformer 内部で各単語の位置を表現する位置符号化において，単語の絶対位置と構造位置 (単語の依存構造木での深さ) を同時に符号化することの有効性が示されている．一方，既存法では，符号化された値の衝突が起きたり，単語の順序が保存されなくなるという問題がある．本研究では，この問題を解決するため，単語の絶対位置と構造位置を球体表面で表現する位置符号化を提案する．翻訳タスクでの実験を行った結果，既存手法と比べ提案手法は BLEU 値の改善傾向が見られた．また，文の長さ，依存構造木の深さ別での評価を行った結果，構造位置が長文において有効であることがわかった．

1 はじめに

系列変換モデル Transformer [1] の登場によって文生成の性能は大きく向上した．しかし，非常に流暢な文を生成できるものの，原文にはない情報や原文とは異なる内容の文をしばしば生成することがある．固有表現のような簡単な語の誤りに関しては大規模な事前学習済み言語モデルを導入することで改善が期待できるが，原文と同じ単語を使いながら違う意味を生成する場合には不十分である．こうした問題を解決するため，原文の構文構造を活用する手法が提案されている [2, 3, 4]. その中でも，Wang ら [2] は Transformer 内部で単語の位置情報を取り扱う位置符号化において，各単語の依存構造木中における深さと文中の絶対位置 (文の先頭から何単語目か) とを加算する構造化位置符号化を提案した．このように位置情報を扱う埋め込みで系列文以外の情報を表現し単語分散表現に足し合わせる手法は，さまざまなタスクで使われており [5, 6]，効果的である．一方で，この符号化を用いると同じ値をとる単語が複数でてきてしまう，または．通常の絶対位置の順序が変わる可能性がある．絶対位置は生成性能に大きく影響を与える [2] ことから，言語生成タスクにとって重要な性質であり，保持すべき情報である．

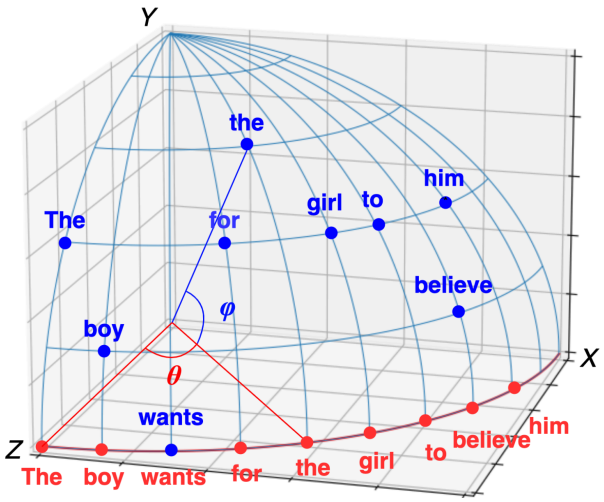

図 1 提案手法の概要図

本稿では，Wang らの方法 [2] の問題を解決するため，単語の絶対位置と依存構造上での深さを球体表面で表現する位置符号化を提案する．図 1 に提案手法の直感的な例を示す．Transformer の位置符号化は，単語の位置を円周上の赤い点として表現する．つまり，角 θ を用いて単語の位置を表す．一方，提案法では，単語を球体表面上の青い点として表現する．つまり，単語の絶対位置は Tranformer の位置符号化と同様，円周上の点として角 θ を用いて表し，依存構造木における深さを仰角 ϕ を用いて表す．このように θ と ϕ を用いて単語の位置を符号化することで値が重複することを回避する．文生成タスクとして機械翻訳を採用し，中英，英日，英独翻訳における評価実験を行った結果，通常の Transformer, Wang らの手法と比較して提案法は BLEU 値の改善傾向がみられた．また，原文の構文の複雑さを文の長さと構文木の深さで分類し，評価した結果から，より複雑な文において提案法が有効であることが明らかとなった．

2 関連研究

2.1 正弦波位置符号化

Transformer では，正弦波位置符号化 (Sinusoidal Positional Encoding, SPE) を用いて文中の単語の絶対位置を符号化し，その値を単語の分散表現に足し合わせる．SPE では，単語の先頭からの絶対位置を pos，埋め込み表現の次元数を d としたとき，単語分散表現の i 番目の要素に足し合わせる値 $E_{\mathrm{SPE}}(pos, i)$ は以下の式で定義される．

$$E_{\mathrm{SPE}}(pos, 2i) = sin(\frac{pos}{\theta}) \tag{1}$$

$$E_{\mathrm{SPE}}(pos, 2i+1) = cos(\frac{pos}{\theta}) \tag{2}$$

このとき，$\theta = 10000^{2i/d}$ であり，偶数次元は正弦関数，奇数次元は余弦関数で定義される．

さらに，SPE は絶対位置のみだけでなく，文中の単語間の相対的位置も表現している．文中の位置 pos から任意の距離 k 離れた位置 $pos+k$ における SPE の値 $E_{\mathrm{SPE}}(pos+k, 2i)$ は $E_{\mathrm{SPE}}(pos, 2i)$ と回転角が $-k/\theta$ の回転行列の線形関数として以下の式で表現できる．

$$\begin{bmatrix} cos(\frac{k}{\theta}) & sin(\frac{k}{\theta}) \\ -sin(\frac{k}{\theta}) & cos(\frac{k}{\theta}) \end{bmatrix} \begin{bmatrix} sin(\frac{pos}{\theta}) \\ cos(\frac{pos}{\theta}) \end{bmatrix} = \begin{bmatrix} sin(\frac{pos+k}{\theta}) \\ cos(\frac{pos+k}{\theta}) \end{bmatrix} \tag{3}$$

よって，SPE は k と pos の位置関係を，回転角が $-k/\theta$ の回転行列を使ってアフィン変換によって表現していると捉えることができる [6]．このように単語間の相対位置を符号化することは機械翻訳の性能向上に寄与することが報告されており [7]，言語生成タスクにとって重要な性質の一つと考えられる．

2.2 構造的位置符号化

前節で説明した Transformer の位置符号化 SPE は文を単語列として扱うための機構であり，そのままでは構文木から得られる情報を符号化することができない．そこで，Wang ら [2] は，位置符号化内部で依存構造木における単語の深さを符号化するための構造的位置符号化 (Structural Positional Encoding, StrcPE) を提案した．dep を依存構造木の根となる単語の深さを 0 とした場合の各単語の依存木における深さとして，通常の絶対位置を符号化したものと dep を符号化したものを加算することで構文を考慮した符号化を行う (式 (4))．

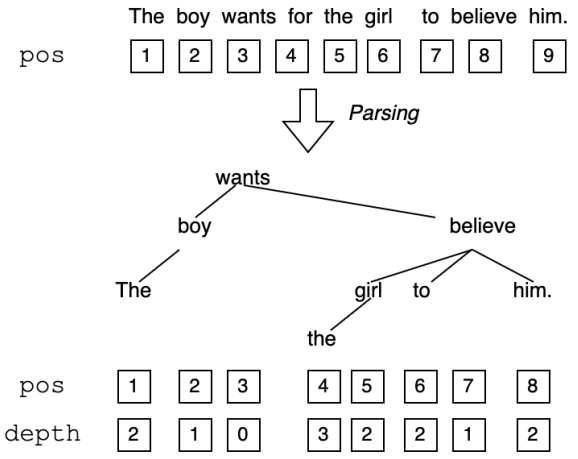

図 2 通常の絶対位置表現 pos と構造的位置表現 $depth$ による表現例

$$E_{\mathrm{StrcPE}}(pos, dep, i) = E_{\mathrm{SPE}}(pos, i) + E_{\mathrm{SPE}}(dep, i) \tag{4}$$

しかし、単語の絶対位置の符号化した値と依存木における深さを符号化した値を加算するだけでは，図 2 にあるように The と boy の値が同一になる，つまり異なる位置に存在する単語の StrcPE 値が重複する．さらにこの符号化は，全次元で絶対位置と構造的位置を足し合わせているため，本来の SPE が持つ絶対位置表現や相対性 (式 (3)) が保存されないという問題点もある．

3 提案手法

単語の符号化された値が重複しない，かつ SPE の持つ相対性を限定的に保持したまま Transformer 内で原文の構文構造を考慮した符号化を実現するため，球体表面を利用した位置符号化 (Hyperspherical Positional Encoding, HPE) を提案する．正弦波位置符号化では単語を $2\pi/10000^{2i/d}$ の等間隔で円周上に配置していると捉えることができる．構造的位置符号化も同様，同じ円周上で絶対位置と依存構造木における深さという 2 つの情報をあわせて表現するが，前節で説明したとおり，符号化された値が重複することがあり得る．提案手法は，これの重複を防ぐため球体表面で依存構造木に位置を表現する．つまり，単語の絶対位置を円周上の点として表すためのパラメタ θ に対して単語の依存構造木中での深さを球面上の点として表すためのパラメタ ϕ を導入する．

球体表面を用いて単語の絶対位置と依存構造木における深さを表現するため，三次元極座標を元に式

(1) と (2) を以下のように変形する．

$$E_{\mathrm{HPE}}(pos, dep, 2i, 2j) = sin(\frac{pos}{\theta})cos(\frac{dep}{\varphi}) \quad (5)$$

$$E_{\mathrm{HPE}}(pos, dep, 2i, 2j+1) = sin(\frac{pos}{\theta})sin(\frac{dep}{\varphi}) \quad (6)$$

$$E_{\mathrm{HPE}}(pos, dep, 2i+1) = cos(\frac{pos}{\theta}) \quad (7)$$

$$\theta = 256^{2i/d}, \varphi = 64^{2j/2d}, j = 2i \quad (8)$$

ϕ を導入するため，SPE の偶数次元 (式 (1)) を式 (5) と式 (6) へと分割し，奇数次元 (式 (7)) は式 (2) と同様とした．この時，通常の絶対位置が生成性能に大きな影響を与える [2] ことから，半分の次元で絶対位置のみを保持し，残り半分の次元で絶対位置と依存構造木中での深さを表現するようにした．さらに，値の増減が極端に小さくなるのを避けるため，パラメータ θ とパラメータ ϕ の分母をそれぞれ $256^{2i/d}$ と $64^{2j/d}$ とする．pos または dep を固定した時，SPE が持つような相対性 (式 (3)) を HPE が持つことは自明である．

4 実験

設定 英独，中英，英日の翻訳タスクにおいて，SPE,StrcPE,HPE を採用した Transformer の比較評価を行なった．英独は，WMT14 データセットを用いた．440 万文対の学習データ，3,000 文対の開発データ，2,737 文対のテストデータからなる．実際のデータは Stanford NLP group[1)]が公開しているものを利用した．中英は，NIST データセットを用いた． 125 万文対の学習データ，878 文対の開発データ，5,262 文対のテストデータからなる．[2)]英日は ASPEC [9] データセットを用いた．178 万文対の学習データ，1,790 文対の開発データ，1,812 文対のテストデータからなる．学習には [10] に倣って 100 万文対の学習データである train-1.txt のみを使用した．入出力はサブワードとし，Sentencepiece [11] を使いトークナイズを行った．このとき，語彙サイズは英独，中英で 32,000，英日で 16,000 とし，言語間で共有した．実装には fairseq [12] を用い，ハイパーパラメータは全てにおいて文献 [1] と同じに設定した．原文の依存構造の解析には spaCy[3)]を用いた．英語の解析には，en_core_web_sm-3.2.0 モデル, 中国語の解析には zh_core_web_sm-3.2.0 モデルを用いた．評価指標には BLEU[13] を用いた．実験は異なるシードを用いて各 3 回行い，その平均を報告する．

1） https://nlp.stanford.edu/projects/nmt/

2） [2, 8] にならい，学習には LDC2002E18, LDC2003E07, LDC2003E14, そして，LDC2004T07, LDC2004T08, LDC2005T06 の Hansards 箇所を用いた．開発には NIST2002 テストセット，テストには NIST2003,2004,2005,2006 のテストセットを用いた．

3） https://spacy.io/

	英日	中英	英独
SPE	41.46	47.46	30.30
StrcPE	41.76	47.73	30.16
HPE	**41.86**	**47.90**	**30.43**

表 1 英独，中英，英日翻訳実験における BLEU

	英日	中英
HPE	**41.6**	**47.3**
$\frac{depth}{\varphi} \Leftrightarrow \frac{pos}{\theta}$	35.4	40.5

表 2 HPE と，全次元で $depth$ を考慮するよう pos/θ と $depth/\varphi$ を入れ替えた場合の英日，中英翻訳実験における BLEU

結果 表 1 に実験結果を示す．全てのデータセットに対し，改善幅は小さいものの HPE の BLEU 値は SPE に対して勝っている．一方，StrcPE は HPE よりもさらに BLEU 値の改善幅が小さく，英独では SPE よりも低下した．これらの結果より，HPE は StrcPE よりも依存構造における単語の深さを効果的に符号化できたと考える．

5 分析

5.1 構造的位置の影響

絶対位置 pos を利用しない場合，翻訳性能が大きく低下することが Wang らによって報告されている [2]，HPE では，絶対位置 pos はベクトルの全ての次元で，構造的位置 dep はベクトルの半分の次元でのみで表現される．そこで，構造的位置 dep が符号化にどの程度影響を与えるかを調べた．

HPE において，ベクトルの全ての次元で dep を表現するように式 (5)-(7) の pos/θ と $depth/\varphi$ を入れ替え，英日，英中のデータセットを用いて実験を行った．[4)]なお，θ と φ の分母は式 (8) と同様である．この入れ替えにより，pos はベクトルの半分の次元でのみ表現されることに注意されたい．表 2 に実験結果を示す．どちらの実験でも，ベクトルの全次元で dep を考慮すると，BLEU 値は大きく低下した．このことから，位置符号化において絶対位置が深さよりも明らかに支配的な役割を持つことがわかる．

4） 実験は 1 回のみ行った．

サブワード数	0-10	11-20	21-30	31-40	41-50	51-60	61-70	71-
文数	231	1419	1387	1087	603	301	134	100
SPE	**41.21**	42.18	44.44	44.22	43.41	42.62	41.27	41.21
StrcPE	39.82	42.15	44.73	44.54	43.66	42.76	42.03	41.95
HPE	39.47	**42.59**	**44.86**	**44.67**	**43.83**	**43.10**	**42.34**	**42.20**

表 3　中英翻訳実験におけるサブワード長別の BLEU

木の深さ	1	2	3	4	5	6	7-
文数	93	450	1137	1498	1142	614	328
SPE	**35.02**	41.80	43.10	43.69	45.48	45.00	43.55
StrcPE	33.50	41.63	42.77	**44.28**	45.76	45.18	43.58
HPE	34.70	**41.89**	**43.44**	44.21	**45.81**	**45.19**	**44.21**

表 4　中英翻訳実験における構文木の最大の深さ別の BLEU

5.2　構文の複雑さにおける評価

深さを符号化する有効性を詳細に調べるため，文のサブワード数と依存構造木の最大の深さごとに BLEU 値を計算した．

表 3,4 に中英データセットにおける結果を示す．表 3 から，11 サブワード以上の場合，HPE の BLEU が全体的にやや高いことから，短い文以外には若干の効果があると考えられる．SPE と HPE を比較すると，61 サブワード以上の場合には BLEU 改善幅が顕著である．このことから，構造位置は短い文ではあまり影響を与えることはないが，長文だと効果があることがわかる．一方，10 サブワード以下の非常に短い文の場合では SPE が最も良いことから，単純な絶対位置のみで十分であることがわかる．

表 4 より，表 3 と同じく，ほとんど場合で HPE は SPE，StrcPE よりも BLEU 値が良い．1 文あたりの依存構造木の最大の深さごとにみると，木の最大の深さが 1 といった単純な構造木の時，HPE は SPE よりも悪い．一方，最大の深さが 6 までのとき，StrcPE や HPE の BLEU 値はやや向上している．最大の深さが 7 以上といった複雑な構造を持つ文の場合，BLEU は 0.6 ポイントほど向上した．これらより，単純な構造を持つ文では，構造位置の効果が小さく，長い文または複雑な文の時にそれが有効であることがわかった．

6　まとめ

本稿では Transformer において，単語の絶対位置と構造位置 (依存構造を用いた深さ) を球体表面で表現し符号化する，球体表面を利用した位置符号化 (Hyperspherical Positional Encoding, HPE) を提案した．英日，中英，英独の機械翻訳データセットを用いて提案法を従来からの位置符号化法と比較評価した結果，顕著ではないものの BLEU 値の改善傾向がみられた．文長，依存構造木の深さごと評価結果から，長い文，依存構造が深い文においては依存構造を用いた深さは有効であることがわかった．さらに，同じ深さを表現する既存研究と比較した場合，提案手法は BLEU 値の改善傾向が見られた．

参考文献

[1] 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.

[2] Xing Wang, Zhaopeng Tu, Longyue Wang, and Shuming Shi. Self-attention with structural position representations. 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)**, pp. 1403–1409, Hong Kong, China, November 2019. Association for Computational Linguistics.

[3] Chuan Wang, Nianwen Xue, and Sameer Pradhan. A transition-based algorithm for AMR parsing. In **Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, pp. 366–375, Denver, Colorado, May–June 2015. Association for Computational Linguistics.

[4] Yudai Kishimoto, Yugo Murawaki, and Sadao Kurohashi. Adapting BERT to implicit discourse relation classification with a focus on discourse connectives. In **Proceedings of the Twelfth Language Resources and Evaluation Conference**, pp. 1152–1158, Marseille, France, May 2020. European Language Resources Association.

[5] Angela Fan, Claire Gardent, Chloé Braud, and Antoine

Bordes. Using local knowledge graph construction to scale Seq2Seq models to multi-document inputs. 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)**, pp. 4186–4196, Hong Kong, China, November 2019. Association for Computational Linguistics.

[6] Vighnesh Leonardo Shiv and Chris Quirk. Novel positional encodings to enable tree-structured transformers, 2019.

[7] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In **Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)**, pp. 464–468, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[8] Xing Wang, Zhaopeng Tu, Deyi Xiong, and Min Zhang. Translating phrases in neural machine translation. In **Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing**, pp. 1421–1431, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[9] 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).

[10] Yui Oka, Katsuhito Sudoh, and Satoshi Nakamura. Length-constrained neural machine translation using length prediction and perturbation into length-aware positional encoding. 自然言語処理, Vol. 28, No. 3, pp. 778–801, 2021.

[11] Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In **Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations**, pp. 66–71, Brussels, Belgium, November 2018. Association for Computational Linguistics.

[12] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. In **Proceedings of NAACL-HLT 2019: Demonstrations**, 2019.

[13] 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.

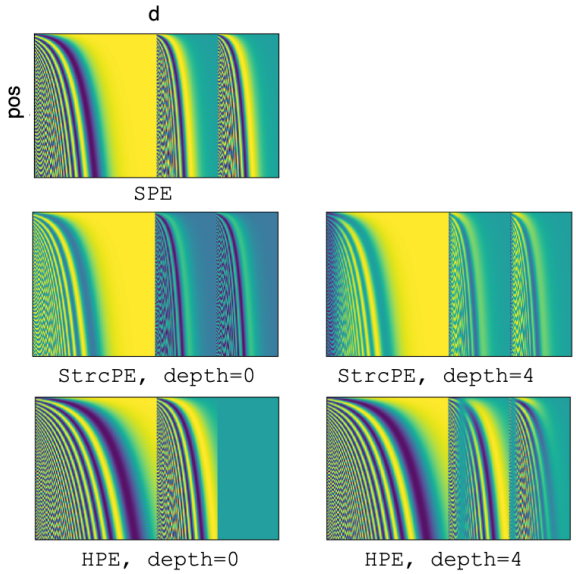

図 3 SPE(上段)，Wang らの StrcPE(中段)，提案手法の HPE(上段) をそれぞれ視覚化したもの．縦軸が単語の絶対位置 pos を表す．横軸は次元数 d であり，式 (8)，式 (6)，式 (7) の次元別に合わせて表している．

A 視覚化

先行研究との比較のため，図 3 に各位置符号化のヒートマップを示す．明晰化のため，横軸を式 (8)，式 (6)，式 (7) のベクトルの次元別に対応して表している．StrcPE ではベクトルの全次元において絶対位置 pos と構造的位置 dep を考慮しているため，dep の値に関わらず全次元でヒートマップの変域が SPE と比べ小さくなっており，位置符号化の表現の幅を狭める可能性がある．一方で，HPE ではベクトルの偶数次元でのみ，絶対位置 pos と構造的位置 dep を考慮し，奇数次元では絶対位置 pos のみ考慮する．そのため，奇数次元は SPE と全く同じであり，$depth = 0$ の時，3/4 の次元で SPE と同じ値を取る．さらに，$dep = 0$ 以外の時でも，変域は SPE と似たような分布になっており，位置符号化の表現の幅を最大限使っていると考えられる．

B パラメータ

HPE では，同じような式変形を式 (6) に施すことで，さらにパラメータ tmp,λ を追加していくことも可能である．

$$E_{\mathrm{HPE}}(pos, dep, tmp, 2i, 2j{+}1, 2k) = sin(\frac{pos}{\theta})sin(\frac{dep}{\varphi})cos(\frac{tmp}{\lambda})$$

$$E_{\mathrm{HPE}}(pos, dep, tmp, 2i, 2j{+}1, 2k{+}1) = sin(\frac{pos}{\theta})sin(\frac{dep}{\varphi})sin(\frac{tmp}{\lambda})$$