ByT5 の Attention を用いたトークン結合

田中康紀　須藤克仁　中村哲
奈良先端科学技術大学院大学
{tanaka.koki.tk9,sudoh,s-nakamura}@is.naist.jp

概要

ニューラルネットを用いた自然言語処理では，事前学習済みモデルを fine-tuning することでタスクを高精度に解ける．自然言語の入力は多くのモデルにおいて単語やサブワード等トークンへの分割を前提としている．この手法は誤分割や，語彙の修正後にモデルの再学習が必要な欠点を持つ．これらは文字レベルのトークン分割により緩和できるが，長系列化と，トークンと意味の不一致を起こす．本研究では，バイトレベルに分割するモデル ByT5 を文字レベル分割とみなし，Attention を元にトークンを結合し，両者の解決を試みた．テキスト分類実験においてランダムに結合する手法に比べ 1.8 ポイントの分類精度向上，ByT5 に比べ 1.78 倍の高速化を示した．

1　はじめに

ニューラルネットを用いた自然言語処理では，事前学習済みモデルを用いて下流タスクに fine-tuning させることで，生成タスクや分類タスクを高精度に解くことができる．現在ある多くのモデルは入力文字列を単語やサブワード等のトークンに分割するトークン化を行う手法が主流となっている．しかし，トークン化は語彙に無いトークンに対する過剰分割や，未知語に起因する精度低下，トークナイザを更新するとモデルの再学習が必要であり技術的負債を生むという欠点を持つ．文字列を全て文字に分解する文字レベルトークン化を用いると，これらの問題を緩和できる．一方で，文字レベルトークン化は文字トークンと意味が対応しない，シーケンス長も単語やサブワードを用いた場合と比べて長くなる問題が起こる．本研究では，バイトレベルにトークン化する ByT5 のトークンを，英語データセットに対して文字レベル分割とみなし，Attention 情報を利用しモデル内部でトークン結合を行う方法を提案する．これにより，処理速度の向上と意味のまとまりを持ったトークンの保持による精度の向上を目指す．英語のテキスト分類タスクでランダムに結合する手法と比べ分類精度の向上，ByT5 と比べ単位時間あたりの処理数の向上を実験により確認した．

2　関連研究

Sutskever ら [1] はニューラルネットを用いた翻訳で，文字レベル入力を導入した．Peter ら [2] は文字レベルの入力を結合する CNN 層を導入したモデルを提案した．Radford ら [3] は，入力文字列をトークン化する際に，語彙にバイトレベルの文字を含めて細分化するバイトレベル Byte Pair Encoding を導入することでことで未知語を極めて少なくした．明示的にトークナイザを用いない Transformer ベースのアーキテクチャとして charfomer [4]，CANINE [5]，characterBERT [6] がある．これらは入力を文字レベルにし，語彙用の層を事前学習で獲得する手法である．ByT5 [7] では，基本構成は T5 [8] と変えず，その層数を変えることで，語彙用の特別な層を持たずして，サブワードレベルのモデルである mT5 [9] と一部タスクで近い性能を出すことを示した．一方で，文字レベルのモデルとサブワードレベルのモデルが同程度の性能を達成した際の各々の獲得した隠れ層や，位置埋め込み，Attention 行列の具体的な差異に関しては議論がまだ活発ではない．

Transformer モデルの長文入力への対応方法として，Zaheer ら [10] は，Attention を疎な行列計算とみなし，計算量を減らした．Kitaev ら [11] は近傍のトークンに注目することで Attention の計算量をシーケンス長 L に対して $O(L^2)$ から $O(L \log L)$ に削減した．Beltagy ら [12] は複数サイズのウィンドウにより Attention の見る位置を絞ることで，長文であっても Attention の計算量を減らすことを可能にした．Goyal ら [13] は，BERT [14] の Attention が強いもののみを抽出することで，トークンを削除し，処理の高速化を実現した．ByT5 は計算量の削減をシーケンス長を減らすのではなく，Deocder を浅くすることで実現している．本研究では，ByT5 に対して，

語彙に依存しない点を残したまま，シーケンス長の削減を導入する．

3 提案手法

本研究ではバイトレベルにトークン化するモデルである ByT5 のトークンをモデル内部で Attention に基づいて結合する手法を提案する．提案手法の概要を図 1 に示す．英語のみ扱うタスクへ fine-tuning することから，ByT5 が取り扱う UTF-8 形式では，概ね 1 文字 1 バイトに対応すると仮定できる．

長さ L の文字列 $\boldsymbol{c} = (c_1, ..., c_i, ..., c_L)$ が与えられ，各 c に対応する隠れ層出力を $\boldsymbol{v}$ とすると，以下のような処理により新たなベクトル $\boldsymbol{v}'$ を得る．

はじめに，局所的な Attention 情報を利用したいことから，浅い層である Encoder ブロック 2 層目の Attention 行列を参照し，Head 方向へ平均をとる．

$$\bar{A} = \frac{1}{H}\sum_{h}^{H} A_h$$

次に，Attention 行列を query 方向へ平均をとり，これを Significance Score とする．

$$\mathrm{Sig}(c) = \frac{1}{L}\sum_{l}^{L} \bar{A}[c_l, c] \tag{1}$$

続いて，各文字ごとの Significance Score を並べ，その中から上位 k 個を抽出し，$\boldsymbol{c}'$ とする．

$$\mathrm{Sig}(\boldsymbol{c}) = (\mathrm{Sig}(c_1), ..., \mathrm{Sig}(c_L)) \tag{2}$$

$$\boldsymbol{c}' = \mathrm{topk}(\mathrm{Sig}(\boldsymbol{c})) = (c'_1, ..., c'_k) \tag{3}$$

c'_i から c'_{i+1} の一つ前までの c に対応する文字列を segment と呼び，$\boldsymbol{s}_i$ とおく．各 $\boldsymbol{s}_i$ の含む文字 c に対応する隠れ層出力の平均をとり，新たに $\boldsymbol{v}_i$ とする．

$$\boldsymbol{v}'_i = \frac{1}{|\boldsymbol{s}_i|}\sum_{c\in\boldsymbol{s}_i} \boldsymbol{v}_c \tag{4}$$

先頭文字トークン c_1 が上位 k 個に選択されなかった場合，c_1 から c'_1 の前までの c に対応する文字列を $\boldsymbol{s}_0$ とし，その文字に対応するベクトルの平均を先頭に追加する．

$$\boldsymbol{v}'_0 = \frac{1}{|\boldsymbol{s}_0|}\sum_{c\in\boldsymbol{s}_0} \boldsymbol{v}_c$$

また，終端トークン </s> は special token であることから，常に一つのトークンとして扱う．よって上位 k 個に選択されなかった場合，以下のようにベクトルを追加する．

$$\boldsymbol{v}'_{k+1} = \boldsymbol{v}_{\text{</s>}}$$

以上の操作で得られたベクトル $\boldsymbol{v}'$ を Encoder ブロックの 3 層目に入力し，以降の層へ伝播させる．パラメータは事前学習済みモデルから再利用し，下流タスクへ fine-tuning する．

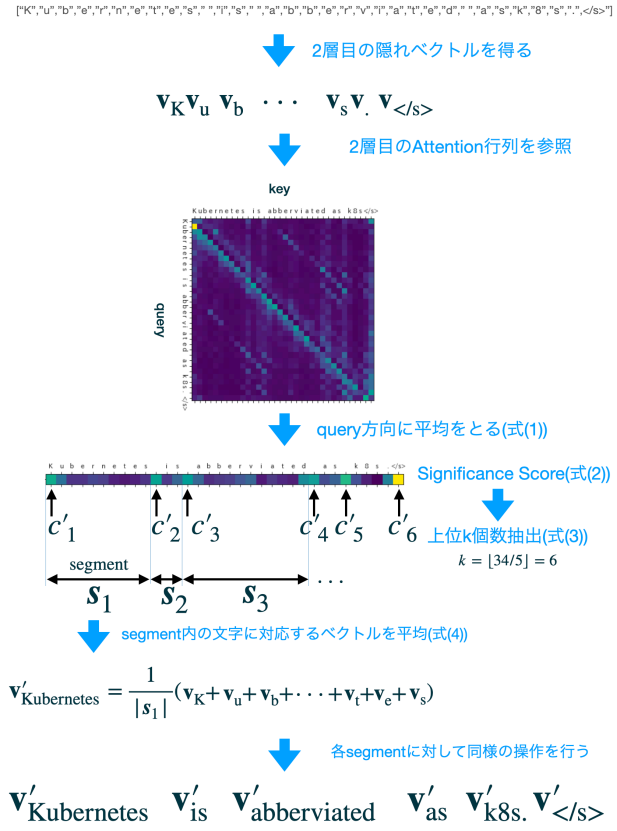

図 1 文字トークン結合の流れ

4 実験

ByT5 の Attention 情報を利用したモデル内部での文字トークン結合が，トークンの意味の獲得による精度の向上と速度の改善に効果があるかを実験により検証した．提案手法は，”e”と”ee”といった連続文字列を式 (4) で平均化後に再度生成することは難しいことから，分類タスクで実験を行った．

4.1 実験タスク

英語のテキスト分類タスクとして SST-2 [15] を用いた．これはテキストに含まれる感情を 2 値分類するタスクである．実験はランダムに k 個のトークンを指定して結合する手法をベースラインとし，強い Attention を先頭に結合する提案手法に意味があるの

かを調べた．加えて，空白で区切って結合する単語レベルトークン化，Goyal ら [13] の手法に基づいて実装した Attention の強いトークンのみを抽出する手法，ByT5 の fine-tuning により実験した．ByT5 では，全ての Encoder 層で文字トークンに対して同じ相対位置埋め込みのパラメータを共有している．一方提案手法では，文字レベルの位置情報と，結合後のトークン位置情報が異なると考えられることから，トークン獲得後の層である 3 層目以降の Attention で 1 層目とは異なる位置埋め込みのパラメータを用いた．なお，予備実験でこれらの手法で大きな差は確認できなかったため，その比較は省略する．

いずれの実験も、バッチサイズ 12，学習率 4e-5 で 40 エポック学習させたのちに、validation データに対して最も accurary が高いものを用いて、test データで accuracy を算出した．シーケンス長は、mT5 が ByT5 の平均トークン長の 1/4.1 倍であることから、それよりも平均で短くなる 1/5 倍の長さになるように式 (3) の k を設定した．実験は異なるランダムシードで 5 回行い，その平均により算出した．

処理速度は，test データの単位時間あたりのイテレーション数を比較した．実験は CPU AMD EPYC 7742 64-Core，メモリ 1TB，GPU NVIDIA A100-SXM 40GB，バッチサイズ 12 で行った．なお，提案手法は文字トークンの結合処理を CPU 上で行うことから，処理速度は GPU 以外の条件にも依存する．

4.2 実験結果

表 1 は，文字トークンの選択手法を変更した際の実験結果である．提案手法はベースラインであるランダムに文字トークンを選択し結合した場合に比べ，1.8 ポイント高い結果となった．また，Attention の強いトークンを抽出する手法に比べ，3.2 ポイント高く，空白で区切る単語レベルトークン化を行う場合よりも 0.8 ポイント高くなった．提案手法は ByT5-base と比較して 6%ほど精度を落とすが，表 2 に示すように，単位時間あたり 1.78 倍のイテレーションを実行する事が可能になった．

5 回の実験結果における混同行列の平均を表 4 に示す．学習データには Positive/Negative いずれも同数含まれている．ByT5-base では若干 Positive 判定が大きかったが，提案手法では Positive と Negative いずれもほぼ同数の判定となった．

表 1 SST-2 における accuracy(%)

ランダムに選択し結合 (ベースライン)	83.5
Attention から選択し結合 (提案手法)	85.3
空白を選択し結合	84.5
Attention から選択し抽出	82.1
ByT5-base	90.6

表 2 各モデルの処理速度の比較

モデル名	処理速度 (it/s)	比率
ByT5-base(ベースライン)	4.88	1
提案手法	8.67	1.78
4 層目で結合	7.12	1.47
12 層目で結合	5.12	1.15

5 考察

5.1 トークンの獲得

表 1 より，ランダムに文字トークンを結合する手法と比較して，Attention の上位 k 個を用いて文字トークンを結合する方法は accuracy が 1.8 ポイント高い．これより Attention のまとまりは，意味のまとまりを示す情報を持っていると考えられる．

また，表 1 より提案手法は accuracy では ByT5 に及ばなかった．Transformer アーキテクチャは，全体から自身のベクトルを再構成しているため，周りの文字トークンの結合による意味を持ったトークン獲得は，Attention の計算で既に行われていたと考えられる．accuracy が ByT5 より下回った原因として，提案手法は式 (4) で文字トークンを平均化していることから，分類タスクを選択したが，本手法は分類タスクでも必要な情報が消えたと考えられる．表 3 は，結合する層を変更した際の実験結果である．より上位の層で結合するほど，accuracy は高い結果となっており，処理速度と精度はトレードオフの関係にある．提案手法は低い層で結合したことから，事前学習と大きく異なる情報を結合以降の層に伝えたことも精度低下の原因であると考えられる．より事前学習を再利用できるトークンの結合方法が存在すれば，この低下は緩和できると考えられる．

図 1 中のスペース区切りの実験結果と比較して，提案手法の方が精度が高い．ニューラルネットによる自然言語処理では，単語モデルよりサブワードモデルの方がタスクを解くのに適していることは，Sennrich ら [16] や Kudo ら [17] も示しており，本実験結果もニューラルネットワークで分類タスクを解

表 3　異なる層で結合した実験結果

2 層目 (提案手法)	85.2
6 層目	87.1
12 層目	89.6

表 4　5 回の実験における混同行列の平均

(a)ByT5

		Predicted	
		Neg	pos
Actual	Neg	382.8	45.2
	Pos	35.2	408.8

(b) 提案手法

		Predicted	
		Neg	Pos
Actual	Neg	362.8	65.2
	Pos	65.4	378.6

く際には，必ずしも単語レベルのトークン化が適しているわけではなく，単語とは異なるより適した最小単位が存在することを示しているとわかる．

5.2　トークンの文字列としての意味

validation データに対するトークン化の例を表 5 に示す．入力のプレフィックスである，”sst2 sentence:” がいずれのパターンも先頭トークンとして文字列に結合されていることがわかる．これは，fine-tuning の際に全てのタスクに付属していることから，細分化の必要がないと学習したため獲得できたと考えられる．表 5 の 1 例目で”affecting”という単語が確認できるように，一部は単語としての意味を持ったトークンの結合ができている．一方で，サブワードにトークン化する際，一般的に 1 つのトークンとされる”and”や”ing”のような，複数文で共通の部分文字列が結合されている様子は確認できない．本研究では fine-tuning では，2 値分類タスクを行ったため，スペースを区切りとした簡易的な結合は獲得できたものの，サブワードのように，共通部分の結合を獲得するにはタスクが簡素であったと考えられる．

5.3　Attention 行列の差異

表 2 は，validation データのうちの一つである，”sst2 sentence: i had to look away - this was god awful . ”を ByT5-base と提案手法の両者に入力した画像である．それぞれ，2，3，12 層目の Attention 行列をそれぞれ Head 方向に平均をとった．提案手法の Attention 行列は，ByT5-base と同様に，Encoder 最終層では query 方向の強い値が連続していることが見てとれる．従って，文字トークン結合後も，各トークンに対して Attention が機能していると考えられる．一方で，ByT5 の Attention 行列の対角成分は，残差ネットワークにより，自身を再構成が容易なことから，小さい値となっているが，提案手法である結合後の Attention 対角成分にそのような様子は見られない．これは，トークンの結合を残差ネットワークの後に設定したため学習されなかったと考えられる．

表 5　validation データに対する結合例

入力文
sst2 sentence: it ’s a charming and often affecting journey .
結合結果
sst2_sentence:_it’s_ / a_ / cha / r / ming_and_o / fte / n_ / affecting_ / jo / urne / y_ / . / <s>
入力文
sst2 sentence: unflinchingly bleak and desperate
結合結果
sst2_sentence:_ / unf / linc / hingly_ble / a / k_and_ / / des / pe / r / ate / <s>

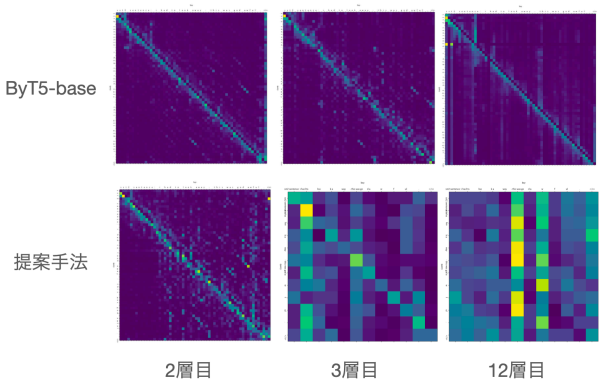

図 2　Attention 行列の様子

6　おわりに

本研究では，バイトレベルにトークン化するモデルである ByT5 に，語彙に依存しない利点を維持したままシーケンス長を削減する手法を提案した．テキスト分類実験において，ランダムに文字トークンを選択し結合する手法と比較して精度の向上を確認し，ByT5-base と比較して精度の下落を小さく抑え，処理速度の向上を確認できた．表現能力の担保を調べるためには，今後生成タスクでも検証する必要がある．文字トークン結合時に情報を脱落させないためには，文字レベルのトークンがサブワードトークンと比較して，具体的にどのような情報を保持しているかを解明し，それらを維持したまま事前学習済みの層へ伝播させる方法の研究が必要である．

7 謝辞

本研究の一部は科研費 21H03500 と 21H05054 の助成を受けたものである．

参考文献

[1] Ilya Sutskever, James Martens, and Geoffrey Hinton. Generating text with recurrent neural networks. In **Proceedings of the 28th International Conference on International Conference on Machine Learning**, ICML'11, p. 1017–1024, Madison, WI, USA, 2011. Omnipress.

[2] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In **Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)**, pp. 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[3] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[4] Yi Tay, Vinh Q. Tran, Sebastian Ruder, Jai Gupta, Hyung Won Chung, Dara Bahri, Zhen Qin, Simon Baumgartner, Cong Yu, and Donald Metzler. Charformer: Fast character transformers via gradient-based subword tokenization, 2021.

[5] Jonathan H. Clark, Dan Garrette, Iulia Turc, and John Wieting. Canine: Pre-training an efficient tokenization-free encoder for language representation. **Transactions of the Association for Computational Linguistics**, Vol. 10, pp. 73–91, 2022.

[6] Hicham El Boukkouri, Olivier Ferret, Thomas Lavergne, Hiroshi Noji, Pierre Zweigenbaum, and Jun'ichi Tsujii. CharacterBERT: Reconciling ELMo and BERT for word-level open-vocabulary representations from characters. In **Proceedings of the 28th International Conference on Computational Linguistics**, pp. 6903–6915, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics.

[7] Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. ByT5: Towards a Token-Free future with pre-trained Byte-to-Byte models. **Transactions of the Association for Computational Linguistics**, Vol. 10, pp. 291–306, 2022.

[8] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. **Journal of Machine Learning Research**, Vol. 21, No. 140, pp. 1–67, 2020.

[9] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mT5: A massively multilingual pre-trained text-to-text transformer. In **Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, pp. 483–498, Online, June 2021. Association for Computational Linguistics.

[10] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. **Advances in Neural Information Processing Systems**, Vol. 33, , 2020.

[11] Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. January 2020.

[12] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The Long-Document transformer. April 2020.

[13] Saurabh Goyal, Anamitra R Choudhury, Saurabh Raje, Venkatesan T Chakaravarthy, Yogish Sabharwal, and Ashish Verma. PoWER-BERT: Accelerating BERT inference via progressive word-vector elimination. In **International Conference on Machine Learning**. International Machine Learning Society (IMLS), February 2021.

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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)**, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[15] Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. https://nlp.stanford.edu/~socherr/EMNLP2013_RNTN.pdf. Accessed: 2022-12-12.

[16] 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.

[17] 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.