BERT を用いた原文と要約文の分散表現の最適な統合手法の検討

1　はじめに

近年, 深層学習をはじめとする機械学習手法は目覚ましい発展を遂げており, 自然言語処理 や画像処理などの分野で高い精度を達成している.

より高精度なモデルを作るための深層学習の要素技術の 1 つにプーリングと呼ばれる手法がある. プーリングとは, 入力されたデータから得た特徴量の次元を縮小し, 計算量を削減するとともに, 抽出された特徴量のロバスト性を高めるための手法である. しかし, 自然言語処理分野におけるプーリング手法は画像処理分野と比べて数少なく, その効果に関する理解は不十分であるという背景がある. このような背景を踏まえ, 大和 [1] は大規模言語モデル (Large Language Models, LLM) の 1 つである BERT において一般的に用いられる [CLS] トークンの埋め込み表現を用いたプーリング手法と, 平均プーリング手法を組み合わせた CLS-Average Pooling (CAP 層) を提案し, テキスト分類タスクにおける各プーリング手法のみを用いた場合と比較して, その有効性を示した.

本研究では, 大和による手法を基に, LLM を用いて原文から生成した要約文の分散表現を組み込んだ新たなプーリング手法を検討し, テキスト分類タスクにおける性能向上を目的とした.

2　要素技術

2.1　BERT

Bidirectional Encoder Representations from Transformers (BERT) [2] は, 2018 年に Google が発表した複数層の双方向 Transformer エンコーダに基づく汎用言語モデルである. BERT は, 入力された単語列全体に対応する分散表現と, 入力された単語列の各単語に対応する分散表現を出力する. 事前学習済みモデルを他のタスクに転移学習することが容易で, さまざまなタスクに対応することができる.

本研究では, 東北大学の乾研究室から公開されている日本語版 Wikipedia と CC-100 データセットの日本語部分で事前学習されたモデル[*1]を用いた.

2.2　PLaMo

PLaMo[*2] は Preferred Elements 社が開発している大規模言語モデルである. 多言語に広く対応している Generative Pre-trained Transformer (GPT)[3] とは異なり, 日本語に特化した独自のテキストデータを中心に事前学習されており, 他社モデルをベースとしていないため, 社外ライセンスの縛りや開発上における不明瞭点が取り除かれている国産の基盤モデルである.

3　先行研究

図 1 に大和 [1] による CAP 層の概略を示す. 図 1 のように, BERT におけるテキスト分類で一般的に用いられる [CLS] トークンの埋め込み表現 $E_{[\mathrm{CLS}]}$ および, [CLS] トークン以外の入力トークンすべてに対する平均プーリングによって得られた埋め込み表現 E_{Avg} に対して, 学習可能な和が 1 となるパラメータ $p, q(\geq 0)$ を用いて $E_{[\mathrm{CLS}]}$ および, E_{Avg} の重み付き和

$$pE_{[\mathrm{CLS}]} + qE_{\mathrm{Avg}} \tag{1}$$

を CAP 層の出力とし, 入力データの分散表現とする. また, 重みパラメータ p, q の更新式は更新後の値を p', q' とすると以下の式で表される.

$$p' = \frac{p^2}{p^2 + q^2} \tag{2}$$

$$q' = \frac{q^2}{p^2 + q^2} \tag{3}$$

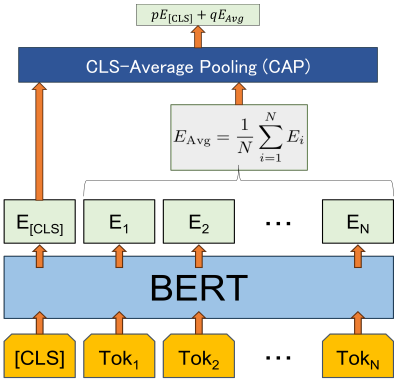

図 1: CLS-Average Pooling (CAP 層) の概要

[*1] https://huggingface.co/cl-tohoku/bert-base-japanese-v3

[*2] Preferred Elements, https://plamo.preferredai.jp/, 2024.

4　データセット

本研究では, 先行研究に倣い livedoor ニュースコーパス[*3] を用いた. これは, 株式会社ロンウィットが公開しているデータセットであり, オンラインニュースサイト「livedoor ニュース」の記事を収集し, 可能な限り HTML タグを取り除いて作成されたコーパスである. 9 つのカテゴリ, 計 7367 記事から構成され, 各記事データ D はカテゴリ名 $\mathrm{D_{label}}$, 記事タイトル $\mathrm{D_{title}}$, 記事本文 $\mathrm{D_{body}}$ などのメタデータから構成されている. 表 1 に実験で用いたデータセットの内訳を示す. 本研究では, 各カテゴリの記事を訓練データ, 検証データ, テストデータとして可能な限りラベル比が均等になるように 8:1:1 に分割した. 表 1 におけるラベルとカテゴリの対応は以下の通りである.

- label 0 : 独女通信 (870 件)
- label 1 : IT ライフハック (870 件)
- label 2 : 家電チャンネル (864 件)
- label 3 : livedoor HOMME (511 件)
- label 4 : MOVIE ENTER (870 件)
- label 5 : Peachy (842 件)
- label 6 : エスマックス (870 件)
- label 7 : Sports Watch (900 件)
- label 8 : トピックニュース (770 件)

5　提案手法

本研究では, PLaMo API を用いてデータセットに含まれるメタデータから各記事データ D の要約文を事前に生成し, 要約文から得た分散表現を CAP 層に組み込むプーリング手法について提案する. 図 2 にモデルの概略を示す.

まず, 要約文の生成手法について説明する. 各記事データ D に対して, プロンプト $\mathrm{D_{prompt}}$ を作成し, PLaMo からの回答 $\mathcal{L}(\mathrm{D_{prompt}})$ を記事データ D の要約文 $\mathrm{D_{summary}}$ とした. 図 3 に要約文の生成プロンプトの概要を示す.

そして, 図 2 のように 2 つの独立した事前学習済 BERT モデルを用いて, BERT (1) では原文データにおける [CLS] トークンの埋め込み表現 $E_{[\mathrm{CLS}]}$ および, [CLS] トークン以外の入力トークンすべてに対する平均プーリングによって得られた埋め込み表現 E_{Avg} を

[*3] http://www.rondhuit.com/download.html

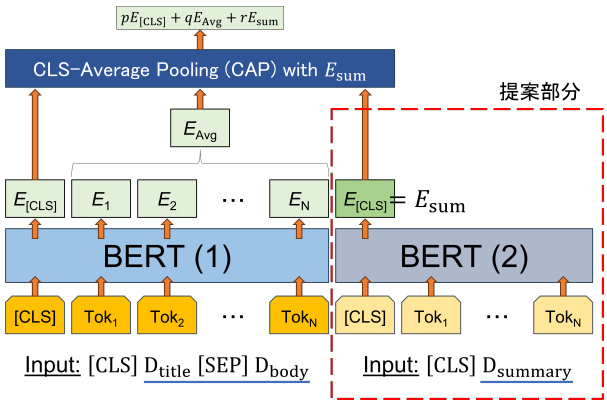

図 2: 提案手法モデルの概要

$$D_{summary} = \mathcal{L}(D_{prompt}) \qquad \mathcal{L}: \text{大規模言語モデル API (e.g., ChatGPT API, PLaMo API)}$$

$$D_{prompt} = \begin{bmatrix} \text{“}role\text{”}: \text{“system”}, \text{“}content\text{”}: \text{“あなたは優秀な AI アシスタントです。”} \\ \text{“}role\text{”}: \text{“user”}, \text{“}content\text{”}: \text{“以下は「}D_{title}\text{」というタイトルの} \\ \qquad\qquad \text{記事の本文です。内容を要約し、} \\ \qquad\qquad \text{結果のみを出力してください。”} \\ \text{“}role\text{”}: \text{“user”}, \text{“}content\text{”}: \text{“}D_{body}\text{”} \end{bmatrix}$$

図 3: 要約文の生成プロンプト

学習し, BERT (2) では要約文データにおける [CLS] トークンの埋め込み表現 E_{sum} を並行して学習して, 総和が 1 となる学習可能なパラメータ $p, q, r(\geq 0)$ を用いて $E_{[\mathrm{CLS}]}$, E_{Avg}, E_{sum} の重み付き和

$$pE_{[\mathrm{CLS}]} + qE_{\mathrm{Avg}} + rE_{\mathrm{sum}} \tag{4}$$

を入力データに対する分散表現とし, これを分類器への入力とする. また, 重みパラメータ p, q, r の更新式は先行研究に倣い, 更新後の値を p', q', r' とすると以下の式で表される.

$$p' = \frac{p^2}{p^2 + q^2 + r^2} \tag{5}$$

$$q' = \frac{q^2}{p^2 + q^2 + r^2} \tag{6}$$

$$r' = \frac{r^2}{p^2 + q^2 + r^2} \tag{7}$$

本研究では, 原文データにおいて記事タイトルと記事本文のペアを 1 文書として扱い, 各 BERT モデルへの入力形式は, 原文データを扱う BERT (1) では先行研究に倣って「 [CLS] $\mathrm{D_{title}}$ [SEP] $\mathrm{D_{body}}$」とし, 要約文データを扱う BERT (2) では「 [CLS] $\mathrm{D_{summary}}$」の形式とした.

6　数値実験

本研究では, 提案手法の有効性を示すために, 先行研究と同様に livedoor ニュースコーパスの記事ラベルを推定するテキスト分類タスクを解き, 検証データ

表 1: データセットの内訳 (livedoor ニュースコーパス)

	label 0	label 1	label 2	label 3	label 4	label 5	label 6	label 7	label 8	Total
訓練	684	695	690	421	700	645	713	716	631	5895
検証	88	80	88	42	93	100	77	105	63	736
テスト	98	95	86	48	77	97	80	79	76	736

表 2: 実験パラメータ

パラメータ	値
最適化手法	Adam
学習率	3.0×10^{-5}
損失関数	Cross Entropy Loss
エポック数	20
バッチサイズ	16
最大トークン長	512
重み初期値 (提案手法)	$(p, q, r) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$
重み初期値 (従来手法)	$(p, q) = (0.5, 0.5)$

表 3: 実験結果

	評価指標	
	Accuracy	F1 値 (weighted)
提案手法	<u>0.9687</u>	<u>0.9683</u>
従来手法 (CAP のみ)	0.9565	0.9560
E_{sum} のみ	0.8369	0.8340
ベースライン	0.1111	-

における F1 値が最も高いエポックにおけるモデルを用いてテストデータに対する精度を評価した. また, 精度比較のため従来手法として CAP 層のみを用いた場合と, 原文データを入力として使わずに, 要約文データにおける [CLS] トークンの埋め込み表現 E_{sum} のみを分類器への入力とした場合についても評価した. 表 2 に実験パラメータを示す. 分類器は 1 層からなる全結合層とし, BERT (1) および BERT (2) についてはそれぞれの最終層のみをファインチューニングした.

表 3 に実験結果を示す. ベースラインにはランダムでラベルを推定した場合を設定した. 表 3 より, 提案手法における Accuracy は従来手法と比較して 0.0122 ポイント向上しており, F1 値に関しても従来手法より 0.0123 ポイント上回っている. E_{sum} のみを用いて推定した場合, 訓練データに対しては他の手法と同様に上手く学習が進んだが, 汎化性能が得られず高い評価指標を示さなかった.

次に, 図 4 に提案手法におけるパラメータ (p, q, r) の学習推移, 図 5 に従来手法におけるパラメータ (p, q)

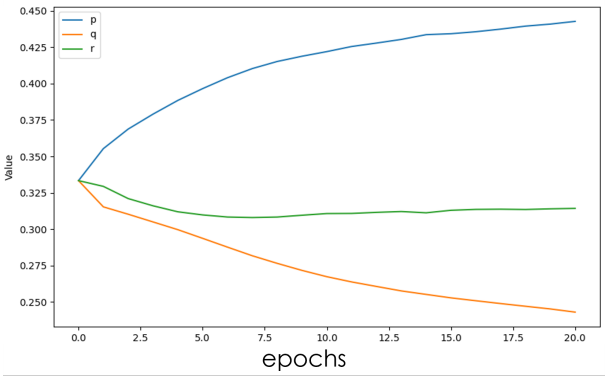

図 4: パラメータ (p, q, r) の学習推移

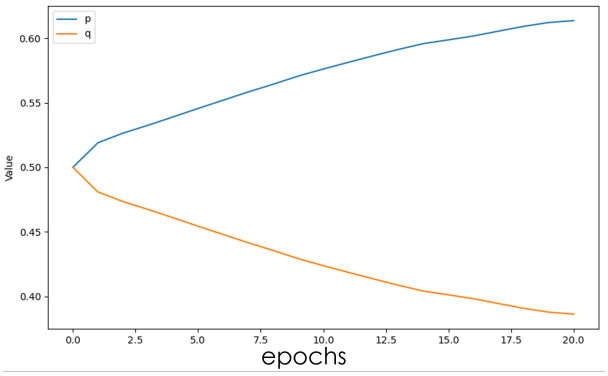

図 5: パラメータ (p, q) の学習推移

の学習推移を示す. 縦軸はパラメータの値, 横軸はエポック数を示している. 最終エポックでの学習後におけるパラメータの値は, 提案手法では $(p, q, r) = (0.441, 0.245, 0.314)$ となり, 従来手法では $(p, q) = (0.613, 0.387)$ となった. 図 4 および図 5 より, 原文データにおける [CLS] トークンの埋め込み表現 $E_{[\text{CLS}]}$ の重み p は単調増加し, [CLS] トークン以外の入力トークンすべてに対する平均プーリングによって得られた埋め込み表現 E_{Avg} の重み q は単調減少していることが分かる. このことから, テキスト分類タスクにおいて, $E_{[\text{CLS}]}$ は, E_{Avg} に比べて, 学習における重要度が高く, その重みが相対的に大きいことが分かる. そして, このタスクが $E_{[\text{CLS}]}$ に強く依存しているのであれば, 提案手法において要約文データにおける [CLS] トークンの埋め込み表現 E_{sum} の重み r に関しても

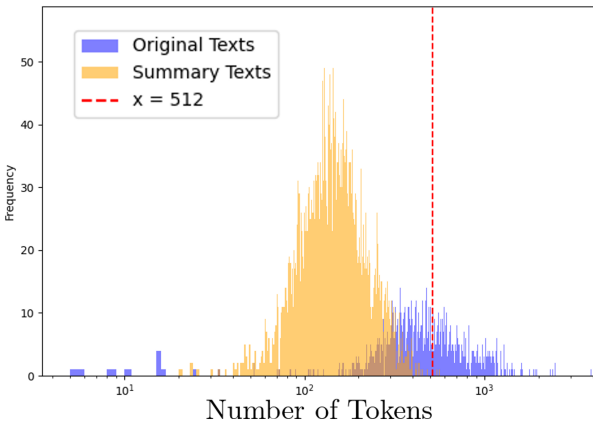

図 6: 訓練データにおけるトークン数分布

表 4: トークン数が閾値より大きいかで分割した訓練データ数

	トークン数	
	< 512	≥ 512
原文データ	2148	3747
要約文データ	5884	11

単調に減少していくと考えられるが, 図 4 より, r の値は一定の値まで減少した後, わずかに増加する傾向が見られた. この結果より, E_{sum} が学習過程においてポジティブな寄与をしていることが分かった.

さらに, 図 6 に訓練データにおけるトークン数の分布を示す. 縦軸は度数, 横軸は対数スケールでトークン数を示している. また, 紫色の分布が原文データ, 黄色の分布が要約文データのトークン数を表しており, 赤色の点線は本研究で使用している事前学習済 BERT モデルにおける最大入力トークン長である 512 を示している. また, 表 4 は図 6 をもとに 512 トークンを閾値としてデータ数を算出したものである. 表 4 より, 原文データにおいて 512 トークンを超えるデータは 3747 件あり, これは全体の約 63% を占めているのに対して, 要約文において 512 トークンを超えるデータは 11 件しかなく, これは全体の約 0.18% と大幅に減少していることが分かった. このことから, 要約文データにおける [CLS] トークンの埋め込み表現 E_{sum} を学習に組み込むことで, 従来手法では学習時に欠損してしまっていた原文データの重要な文脈情報を補完し, 分類精度の向上に貢献したと考えられる.

以上の結果より, 提案手法の有効性を確認することができた.

7 まとめと今後の課題

本研究では, 大和による [CLS] トークンの埋め込み表現を用いたプーリング手法と, 平均プーリング手法を組み合わせた CLS-Average Pooling (CAP 層) を基に, LLM を用いて生成した要約文の分散表現を組み込んだ新たなプーリング手法を提案し, テキスト分類タスクにおいて分類精度を検証した. その結果, Accuracy, F1 値ともに提案手法が高い評価指標を示し, その有効性を確認することができた.

今後の課題としては, 以下のものが挙げられる.

- 要約文に対する妥当性の検討, および生成手法の改善

  本研究では PLaMo API のみを用いて原文データから要約文を生成したが, その妥当性については考慮していない. また, プロンプト内で回答の出力形式を自由にしていたため, 要約文データの一貫性を担保できていない. よって, 異なる LLM の活用や, 回答の出力形式をルールベースで厳格化することでより高品質な要約文が得られると考えられる.

- 他のタスクやデータセットにおける提案手法の有効性の検証

  本研究では livedoor ニュースコーパスデータセットを用いたテキスト分類において, 提案手法の有効性を示した. 一方で他のタスクやデータセットにおける提案手法の有効性については不明瞭であるため, より複雑なデータセットや実験設定での検証が必要である.

参考文献

[1] 大和秀徳. BERT の分散表現に対する学習可能な重みを持つプーリング手法の提案. 大阪公立大学 大学院情報学研究科 基幹情報学専攻 知能情報分野 修士論文, 2024.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. 2019.

[3] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *arXiv preprint arXiv:1801.06146*, 2018.