BERT を用いた原文と要約文の分散表現の統合手法

第 1 グループ　高山 裕成

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 Model, 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) 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 API を用いて原文から生成した要約文の情報を組み込んだプーリング手法を提案し, より適切な文章の分散表現を獲得することを目標とする.

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. 本研究では, 東北大学が公開している日本語 BERT-base モデル[1]を用いた.

2.2. PLaMo

PLaMo[3] とは 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)[4] 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国産の基盤モデルである. そのため, 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.

3. 先行研究

大和 [1] による CAP について述べる. 大和は, 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

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

を CAP の出力とし, C を入力文の分散表現とした. 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.

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

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

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

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[2] を用いた. これは, ニュースサイト「livedoor ニュース」の記事を収集し, 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 $\mathrm{D}_{\mathrm{label}}$, 記事タイトル $\mathrm{D}_{\mathrm{title}}$, 記事本文 $\mathrm{D}_{\mathrm{body}}$ などのメタデータから構成されている. 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.

5. 提案手法

図 1 に提案モデルの概要を示す. 提案手法ではまず, PLaMo API を用いてデータセットに含まれる各記事データ D に対して, メタデータを基にプロンプト $\mathrm{D}_{\mathrm{prompt}}$ を作成し, PLaMo API からの回答 $\mathcal{L}(\mathrm{D}_{\mathrm{prompt}})$ を記事データ D の要約文 $\mathrm{D}_{\mathrm{summary}}$ とした. 図 2 に要約文の生成プロンプトの概要を示す.

次に, 図 1 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, 平均プーリングによって得られた埋め込み表現 E_{Avg} を学習し, 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 $E'_{[\mathrm{CLS}]}$ および, 平均プーリングによって得られた埋め込み表現 E'_{Avg} を並行して学習する. 各 BERT モデルへの入力形式は, 原文データを扱う BERT (1) では大和による手法に倣って「 [CLS] $\mathrm{D}_{\mathrm{title}}$ [SEP] $\mathrm{D}_{\mathrm{body}}$」とし, 要約文データを扱う BERT (2) では「 [CLS] $\mathrm{D}_{\mathrm{summary}}$」の形式とする. そして, 得られた 4 つの分散表現 $E_{[\mathrm{CLS}]}$, E_{Avg}, $E'_{[\mathrm{CLS}]}$, E'_{Avg} に対し, 学習可能な和が 1 となるパラメータ $p, q, r, s(\geq 0)$ を用いて, これらの重み付き和

$$C = pE_{[\mathrm{CLS}]} + qE_{\mathrm{Avg}} + rE'_{[\mathrm{CLS}]} + sE'_{\mathrm{Avg}} \tag{4}$$

を入力文の分散表現とする. また, 重みパラメータ p, q, r, s の更新式は各パラメータの二乗和を $\|\mathbf{W}\|^2 = p^2+q^2+r^2+s^2$, 更新後の値を p', q', r', s' とすると以下の式で表される.

$$(p', q', r', s') = \left(\frac{p^2}{\|\mathbf{W}\|^2}, \frac{q^2}{\|\mathbf{W}\|^2}, \frac{r^2}{\|\mathbf{W}\|^2}, \frac{s^2}{\|\mathbf{W}\|^2}\right) \tag{5}$$

6. 数値実験

提案手法の有効性を示すために, 先行研究と同様にニュース記事のラベルを推定する 9 クラスのテキスト分類タスクを解き, 検証データにおける 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分類精度を評価した. ベースラインとしては大和による手法 (CAP) を設定し, 比較手法としては PLaMo API を用いた Zero-Shot, Few-Shot における分類手法を設定した.

[2] http://www.rondhuit.com/download.html

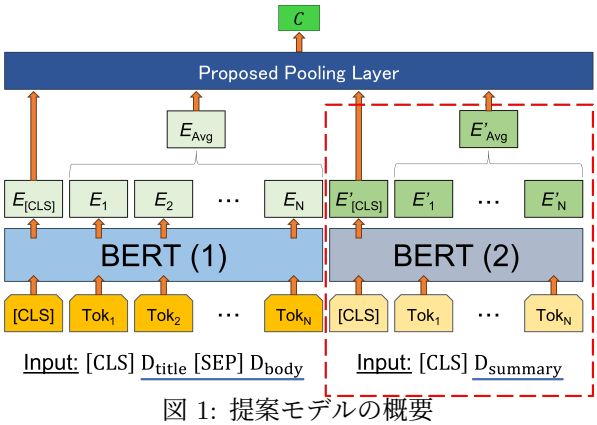

図 1: 提案モデルの概要

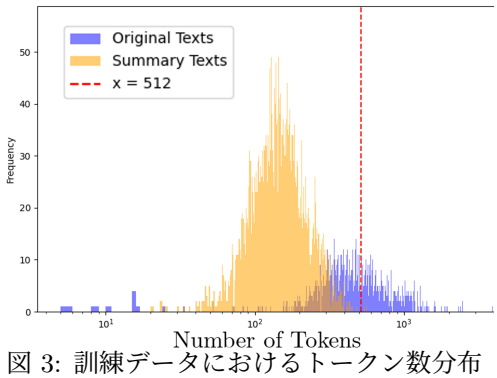

図 3: 訓練データにおけるトークン数分布

$$
\begin{cases}
D_{summary} = \mathcal{L}(D_{prompt}) & \mathcal{L}: \text{大規模言語モデル API (e.g., ChatGPT API, PLaMo API)} \\
D_{prompt} = \left[\begin{array}{l} \text{“}role\text{”}: \text{“system”}, \text{“}content\text{”}: \text{“あなたは優秀な AI アシスタントです。”} \\ \text{“}role\text{”}: \text{“user”}, \text{“}content\text{”}: \text{“以下は「}D_{title}\text{」というタイトルの} \\ \qquad\qquad\qquad\qquad \text{記事の本文です。内容を要約し、} \\ \qquad\qquad\qquad\qquad \text{結果のみを出力してください。”} \\ \text{“}role\text{”}: \text{“user”}, \text{“}content\text{”}: \text{“}D_{body}\text{”} \end{array} \right]
\end{cases}
$$

図 2: 要約文の生成プロンプト

表 1: 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

表 2: 実験結果

手法		精度	手法		精度
提案手法	best	96.7	Zero-Shot	**Top**$_1$	30.2
	mean	96.5		**Top**$_3$	49.5
ベースライン	best	96.1	Few-Shot	**Top**$_1$	33.5
	mean	95.8		**Top**$_3$	59.6

6.1. PLaMo API への入力プロンプト

実験に用いたデータセットに含まれるラベルの詳細と分類対象である記事データのメタデータ $\mathrm{D_{title}}$, $\mathrm{D_{body}}$ から正解だと考えられるラベルを上位 3 位まで抽出し, JSON 形式で厳格に回答を求めるプロンプトを作成した. この際, PLaMo API の温度パラメータは 0.0 とした. また, Few-Shot における入出力例としては, 入力トークン数を削減するために各ラベルにつき 1 件のダミー記事データを人手で用意した. 入力トークン数を抑えるため, タイトルおよび本文は 2 行以下の短文とし, 想定される出力に関しても恣意的な値を設定した.

6.2. 実験設定

表 1 に実験パラメータを示す. 提案手法およびベースライン手法では入力文に対する分散表現 C を 1 層の線形層からなる分類器への入力とし, ラベルを推定する. それぞれの手法で用いている BERT モデルについては最終層のみをファインチューニングした. また, 重みパラメータの初期値は提案手法では $(p, q, r, s) = (0.25, 0.25, 0.25, 0.25)$, ベースライン手法では $(p, q) = (0.5, 0.5)$ とした.

6.3. 結果と考察

表 2 に実験結果を示す. 提案手法とベースライン手法は 3 回試行における分類精度の最大値および平均値, PLaMo API を用いた Zero-Shot および Few-Shot は Top – k 精度 $(k = 1, 3)$ を示している. 実験結果より, 提案手法はベースラインおよび比較手法より高い分類精度を達成し, その有効性が確認された. 一方, Few-Shot の精度が低かったのは扱ったデータセットの特性上, 意味的に似たニュースを多く含むラベルの組が存在し, 与えられた入出力例だけではカテゴリ間の曖昧な境界が明確に捉えられなかったことが原因として考えられる. また, LLM API のみの分類精度が提案手法に匹敵していた場合, その実用的意義が薄れるが, 実験結果より本研究で扱ったテキスト分類タスクは現状の LLM API のみを用いる手法では困難であり, 提案手法の意義は認められると考えられる.

さらに, 図 3 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, 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, 要約文データの分散表現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たか, 原文データ内に含まれるキーワードが強調されるように学習が進んだことによって分類精度の向上に貢献したと考えられる.

7. まとめと今後の展望

本研究では, 大和による CLS-Average Pooling (CAP) を基に, LLM API を用いて原文から生成した要約文の情報を組み込んだモデルを提案し, テキスト分類タスクによる検証でその有効性を示した.

今後の課題として, 以下が挙げられる.

- 要約文に対する妥当性の検討, および生成手法の改善
- 最適な学習パラメータやアーキテクチャの探索

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. In **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] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. 2019.

[3] Preferred Elements. PLaMo-100B: A ground-up language model designed for japanese proficiency. *arXiv preprint arXiv:2410.07563*, 2024.

[4] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al. Improving language understanding by generative pre-training. 2018.