修 士 学 位 論 文

題　　目

BERT を用いた原文と要約文の分散表現の統合手法

主査　　森 直樹 教授

副査　　黄瀬 浩一 教授

副査　　吉岡 理文 教授

令和　6　年（　2024　年）度修了

（No.　2210104043　）　　高山 裕成

大阪府立大学大学院工学研究科
電気・情報系専攻 知能情報工学分野

目 次

図 目 次

表目次

1　はじめに

近年, 深層学習の発展により, 自然言語処理（Natural Language Processing, NLP）や画像処理（Computer Vision, CV）をはじめとする多くの分野で高精度なモデルが開発されている. 特に, Transformer[2] に基づく大規模言語モデル（Large Language Models, LLM）は, 様々な NLP タスクにおいて顕著な成果を上げており, Bidirectional Encoder Representations from Transformers (BERT)[3] や XLNet[4], Text-To-Text Transfer Transformer (T5)[5], Generative Pre-trained Transformer (GPT)[6],[7] などが広く利用されている.

NLP において重要なタスクの一つにテキスト分類がある. テキスト分類は, 情報検索や感情分析, ニュース記事の分類など, 多様な応用が可能である. LLM を活用したテキスト分類の性能を向上させるためには, 適切な文章の分散表現を獲得することが不可欠である. そのため, BERT をはじめとする LLM におけるプーリング手法の選択が, テキスト分類タスクの精度に大きく影響を及ぼす. プーリングとは, 入力データの特徴量を適切に集約し, 計算量の削減とともに, ロバストな特徴表現を得るための技術である. CV の分野では様々なプーリング手法が提案されているが, NLP におけるプーリング手法の種類は未だに数少なく, その効果に関する理解も十分ではない. このような背景のもと, 大和 [1] は 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. CAP 層では, 学習可能な重みパラメータを用いて [CLS] トークンの分散表現と平均プーリング手法で得られた分散表現の重み付き和を取ることで 2 つの重みパラメータを適切に学習しながら文章の分散表現を獲得する.

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で原文に含まれる文脈的に重要な情報をより効果的に強調することを目標とする. また, 提案手法の有効性を確認するために, 大和による先行研究に倣ってテキスト分類タスクにおける分類精度を比較する.

本研究の主な貢献は以下の通りである.

- NLP のテキスト分類タスクに対して, 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, その有効性を示した.
- 提案手法を適用した上で, 評価指標の変化や学習曲線の結果から, 要約文の情報を組み込むことで学習に一定の寄与を与えていることを確認した.

以下に本論文の構成を示す.

まず, 2 章で本研究に関連する要素技術について述べ, 3 章では本研究に関連する研究および大和による先行研究について述べる. 次に, 4 章で実験に用いたデータセットの説明について述べ, 5 章では本研究の提案手法について述べる. そして, 6 章で数値実験およびその結果と考察について述べ, 最後に 7 章でまとめと今後の課題について述べる.

2 要素技術

2.1 形態素解析

形態素とは日本語などの自然言語において意味を持つ最小の単位である. 形態素解析は文を形態素に分割し, 各形態素の品詞などを判定する技術である. 日本語の文は各単語が明示的に分離されていないため, 英語のような単語で分割される言語と異なり, 計算機による自然言語処理では形態素解析が必要となる. 形態素解析のツールとしては, JUMAN++ [8], MeCab [9], Sudachi [10] などが挙げられる. 本研究では BERT への入力において日本語文を単語トークンに分割する際に MeCab を用いている.

2.2 局所表現

自然言語処理における単語の表現手法として, 最も基本的なものの一つに局所表現がある. 局所表現の代表的な手法として One-hot 表現が挙げられる. One-hot 表現は, 各単語を語彙集合内の特定の次元に対応させ, 該当する次元のみが 1 で, それ以外は 0 で構成されるベクトルとして表現する手法である. この手法は単純で実装が容易であるという利点を持つが, 各単語間の意味的な関係は表現できておらず, また単語数が増えるとベクトルのサイズが増大してしまいベクトル空間が疎になってしまうという問題点がある.

2.3 分散表現

分散表現は各単語を高次元の実数ベクトルで表現する手法である. 分散表現は局所表現の問題点であった単語間の意味的な関係を捉えられるようになり, 似た意味の単語はベクトル上で近くなるといった特徴や, 分散表現の加算減算が単語の意味上の加算減算に対応する加法構成性を備えている. 単語の分散表現の獲得手法として, Word2Vec [11] がある. この手法は, 同じ文脈で出現する単語は類似した意味を持つと予想されることに基づいており, 写像されたベクトルは, One-hot 表現のような局所表現と異なり, 単語間の意味を考慮した類似度測定や,「王様」–「男」+「女」=「女王」のような単語間の意味に

おける演算などができるようになる. Word2Vec には周辺の単語から, 単語を予測する Contiuous Bag-of-Words (CBOW) や, ある単語から周辺の単語を予測する Skip-gram の手法が挙げられる. また, 文の分散表現を得る手法として, 単語の出現回数に基づく Bag of Words や Word2Vec を拡張した Doc2Vec[12] や Skip-thought[13] がある. 近年では, Word2Vec を改良した GloVe[14] や双方向 Long Short-Term Memory (LSTM) を用いた ELMo [15] や複数層の双方向 Transformer のエンコーダ層を用いた Bidirectional Encoder Representations from Transformers (BERT) [3] も分散表現の獲得手法として挙げられる. 特に ELMo や BERT により獲得した単語の分散表現は文脈上の意味も表現できるとされている. 本研究では分散表現の獲得に BERT を用いている.

2.4 Attention 機構

機械翻訳のタスクに対して考案されたモデルとして, Recurrent Neural Network (RNN) に基づく Encoder-Decoder モデルが存在する. しかし, Encoder-Decoder モデルでは可変長の文を固定長のベクトルにエンコードするため, 長い入力文になるほど隠れ層のノード数が不足してしまい, 学習が困難となる. そこで Bahdanau らはエンコーダ側で入力文の各単語の荷重を決定してエンコードすべき場所を制御する Attention 機構 [16] を提案した. また, Luong らはこの Bahdanau らによるモデルをより単純化したモデルとして, 入力されたすべての単語を使用する Global attentional model [17] を提案した. 図 2.1 にその概略図を示す. 図 2.1 の $\bar{h}_s$ はエンコーダで出力される入力文のすべての単語の隠れ状態ベクトルを表す. また, h_t は デコーダから出力されるターゲットとなる単語の隠れ状態ベクトルである. Attention 機構は各単語に対して文脈上の重要度を学習して, それに応じて各単語の重みを変える. これによって, 長い文章であっても重要な情報には重みが付く. そのため, Attention 機構を持たない通常の RNN モデルに比べて柔軟に対応できる.

(2.1), (2.2), (2.3) 式に Attention 機構の計算式を示す. 例として score を $\bar{h}_s$ と h_t のドット積を h_t の単語の隠れ状態ベクトルの次元数の平方根 $\sqrt{d_h}$ で割ったものとする. Attention 機構では入力文の各単語 x_i に対する荷重 α_i を計算する. α_i の計算には, $\bar{h}_s$ および h_t を用いて得られる score を正規化することに

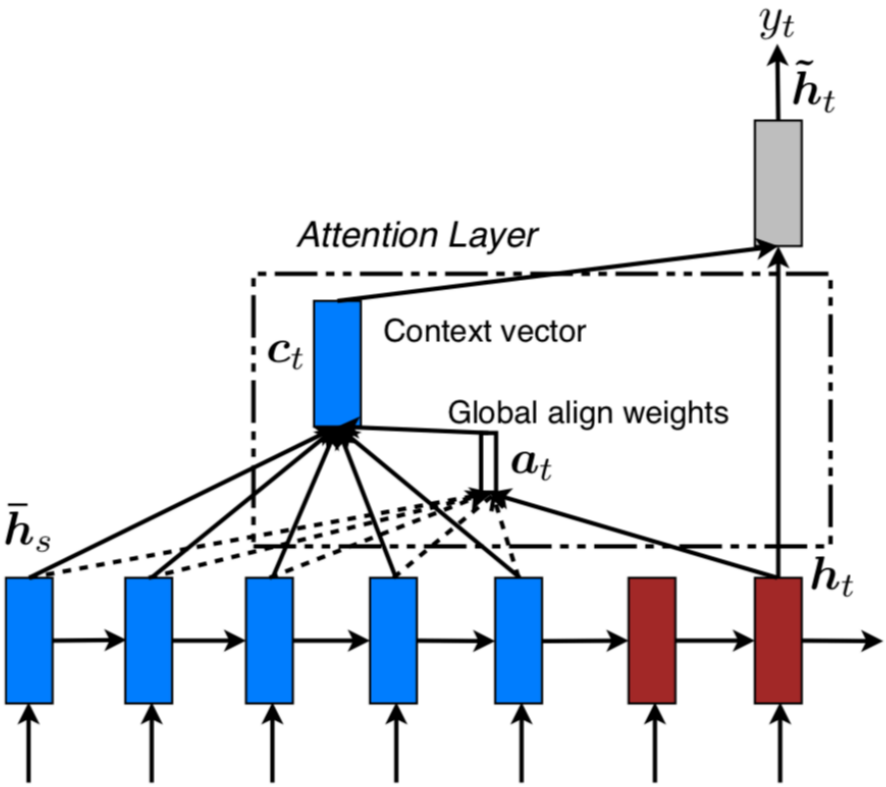

図 2.1: Attention 機構の概略図 (文献 [17] の図 2 より引用)

より得られる. 次に, この比重 α_i と エンコーダから得られた $\bar{h_s}$ を用いてターゲットとなる単語の文脈ベクトル c_t を得る. そして文脈ベクトル c_t とデコーダから得られる h_t によって最終出力が得られる.

$$\mathrm{score}(h_t, \bar{h_s}) = \frac{h_t^{\mathrm{T}} \bar{h_s}}{\sqrt{d_h}} \tag{2.1}$$

$$\alpha_t(i) = \frac{\exp(\mathrm{score}(h_t, \bar{h_s}))}{\sum_{j=1}^{n} \exp(\mathrm{score}(h_t, \bar{h_s}))} \tag{2.2}$$

$$c_t = \sum_{i=1}^{n} \alpha_t(i) \bar{h_s} \tag{2.3}$$

2.5 Transformer

Transformer [18] は 従来の言語モデルで用いられてきた RNN を用いずに Attention 機構のみを基本構造とする Encoder-Decoder モデルである. RNN は時系列データに対して有効であるものの, 前状態の入力を必要とするため文章の先頭の単語から順番に計算する必要があることから, 計算の並列化が難しい. そのため, 多くの計算時間を要するという欠点がある. 一方で Attention 機構のみを用いたモデルは, 行列計算の組み合わせのみで表現できるため, 計算の並列化が可能である. 図 2.2 にその概略図を示す. ここで, 入力文における全ての単語の隠れベクトルを Key K, Value V とし, ある単語の隠れベクトルを Query Q とし, d_k を Key, Value, Query の次元数とする. このとき, (2.1), (2.2), (2.3) 式は (2.4) 式のように書き換えられる.

$$\text{Attention}(Q, V, K) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.4}$$

このとき, Transformer のエンコーダおよびデコーダは Key, Value, Query がすべて同一である Self-Attention が用いられている. そのため, 異なるデータ間の対応関係を獲得するのではなく, 入力データ内の単語同士での類似度や重要度を獲得できる. これによって, 文章内での単語の依存関係を獲得できる.

Transformer のその他の工夫として, Multi-Head Attention が挙げられる. 図 2.3 に, Multi-Head Attention の概要図を示す. これは異なる表現空間の情報を複数組み合わせて用いる手法である. いくつかの線形写像を組み合わせることで d_k 次元の Key と Value と Query を持つ 1 つの Attention 関数を使用するよりも, 効果的に学習ができる.

2.6 Bidirectional Encoder Representations from Transformers

Bidirectional Encoder Representations from Transformers (BERT) [3] 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. これまでの言語モデルは特定の学習タスクに対して 1 つのモデルを用いてきたが, BERT は

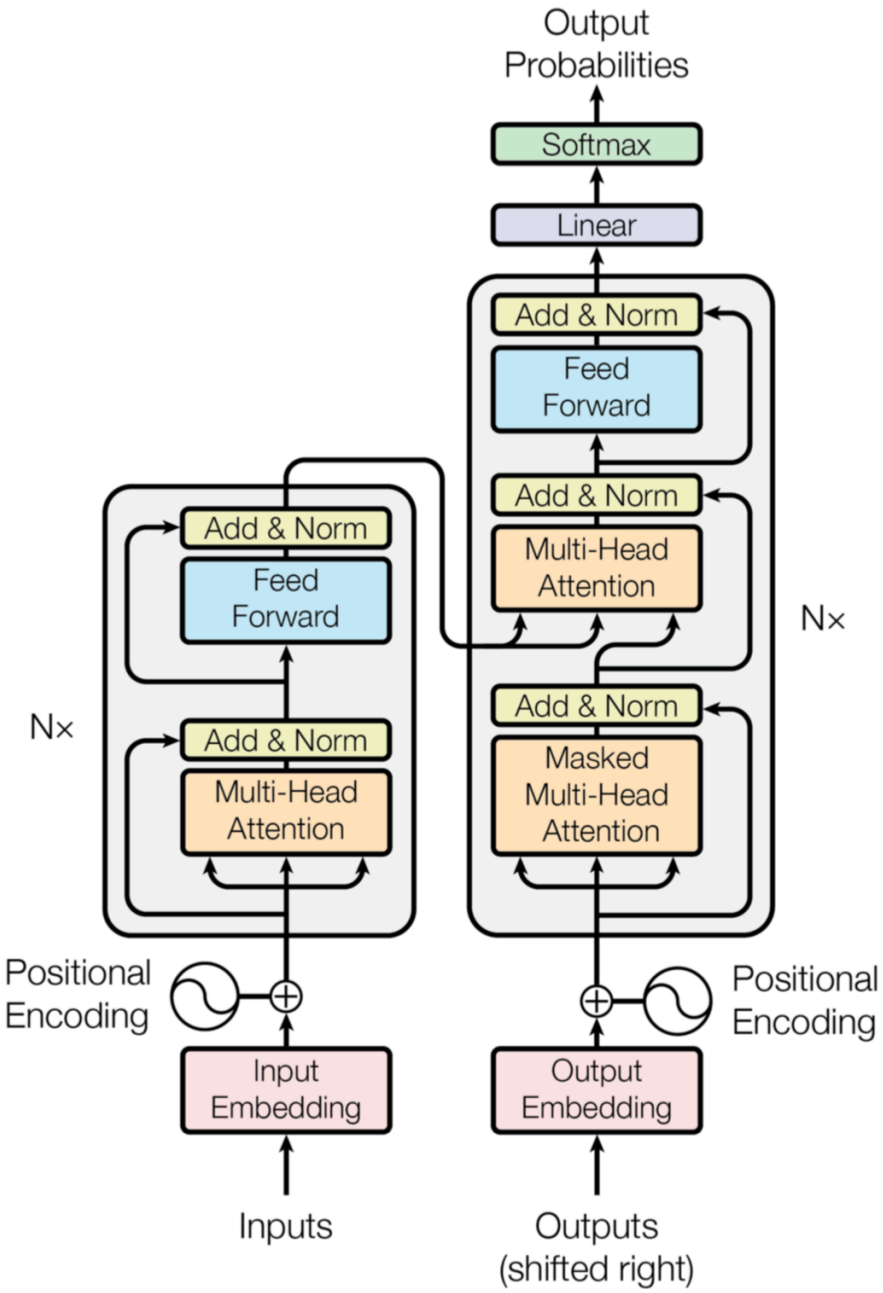

図 2.2: Transformer の概略図 (文献 [18] の図 1 より引用)

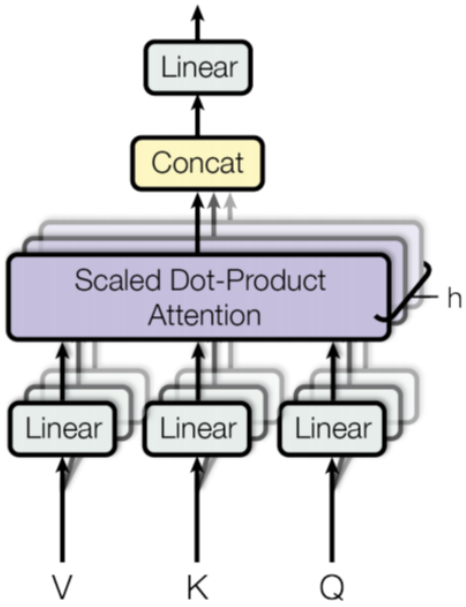

図 2.3: Multi-Head Attention の概略図 (文献 [18] の図 2 より引用)

Wikipedia のような大規模な教師なしコーパスに対して事前学習して, 各タスクに対して転移学習およびファインチューニングをすることで, さまざまなタスクに対応することができる. BERT の事前学習では, 周囲の単語からある単語を予測する Masked Language Model (MLM) と 2 つ目の文章が 1 つ目の文章の次の文章であるかを予測する Next Sentence Prediction (NSP) によりモデルを学習する.

図 2.4 に BERT の事前学習と転移学習の概略図を示す. BERT に文章を入力する際には, 文章の先頭に [CLS] トークンを付与し, 文章の末尾に [SEP] トークンを付与する. BERT は単語ごとの分散表現を出力するが, [CLS] トークンに対する分散表現を文章全体の分散表現として扱い, 分類問題を解くことができる. また事前学習の際, 2 文を扱う際には文章の間に [SEP] トークンを付与する. BERT から得られる単語の分散表現は, 学習済みモデルを基に転移学習し解決すべきタスクに適用させることが可能である. 本研究では テキスト分類に用い文章や単語の分散表現を獲得するために用いた.

2.7 PLaMo

PLaMo[19] 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)[6] 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. 本研究ではベータ版として公開されていたバージョンを用いた.

2.8 Term Frequency-Inverse Document Frequency

Term Frequency-Inverse Document Frequency (TF-IDF) は, 文書集合内での単語の重要度を評価するための統計的手法であり, 情報検索や自然言語処理の分野で広く利用されている [20],[21]. TF-IDF は, ある単語が特定の文書内でどれだけ頻繁に出現するか (Term Frequency), およびコーパス全体におけるその単語の分布 (Inverse Document Frequency) を組み合わせることで, 単語の相対的な重要性を測ることができる. また, TF-IDF は頻出するが情報価値の低い単語 (“the”, “is”, “and” など) の影響を抑え, 文書ごとに特徴的な単語に高い値を与えることから単語の重要度を数値化できるため, 様々なタスクで活用されている.

TF-IDF の計算は以下の (2.5) 式で定義される.

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \text{IDF}(t) \tag{2.5}$$

ここで,

- t は対象の単語（Term）
- d は対象の文書（Document）
- $\text{TF}(t, d)$ は文書 d 内における単語 t の出現頻度
- $\text{IDF}(t)$ は単語 t の逆文書頻度（Inverse Document Frequency）

である.

2.8.1 Term Frequency (TF)

単語 t の文書 d における出現頻度 $\mathrm{TF}(t, d)$ は, 文書 d における単語 t の出現回数 $f_{t,d}$ を用いて, (2.6) 式で定義される.

$$\mathrm{TF}(t, d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \tag{2.6}$$

2.8.2 Inverse Document Frequency (IDF)

単語 t がコーパス内でどれだけ珍しいかを示す尺度であり, (2.7) 式で定義される.

$$\mathrm{IDF}(t) = \log\left(\frac{N}{1 + |\{d \in D : t \in d\}|}\right) \tag{2.7}$$

ここで,

- N はコーパス内の総文書数
- $|\{d \in D : t \in d\}|$ は単語 t を含む文書の数

である.

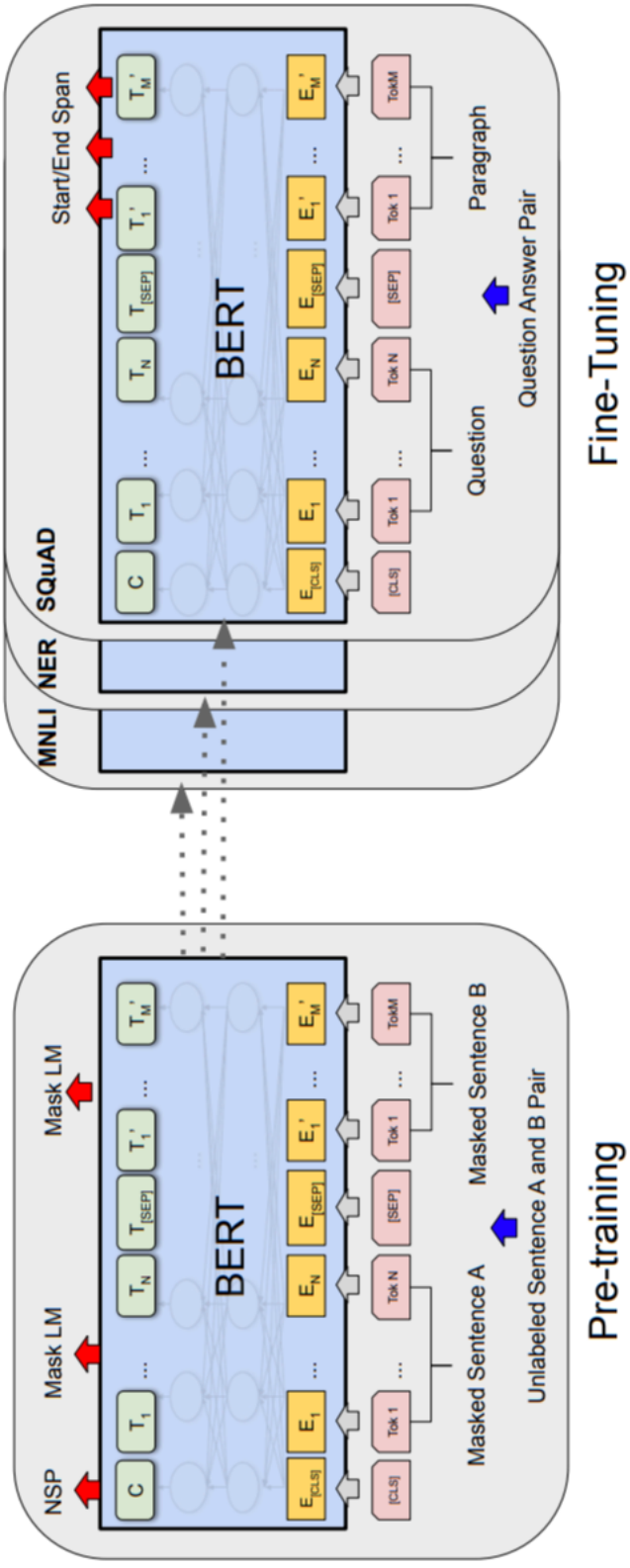

図 2.4: BERT の事前学習と転移学習の概略図 (文献 [3] の図 1 より引用)

3 関連研究

3.1 テキスト分類

テキスト分類とは, 与えられた文章をあらかじめ定義されたカテゴリに分類するタスクである. カテゴリの定義は多岐にわたり, 例えば, 情報検索においては複数の情報の中から共通するトピックを識別し, 感情分析では文章に含まれるポジティブやネガティブといった感情を特定し, 推薦システムではユーザの選好度に基づいてアイテムを分類するなどの応用がある.

テキスト分類は大きく 2 つのステップに分けられる. まず, 特徴量の獲得ステップである. このステップでは One-hot Vector や Word2Vec [11] のような単語の出現頻度に基づくアプローチや Recurrent Neural Network (RNN) [22],[23], Convolutional Neural Network (CNN) [24],[25], Graph Convolutional Network (GCN) [26] といったニューラルネットワーク (Neural Network, NN) の技術を用いたアプローチ, LLM を用いたアプローチ [3],[27],[28] 等が提案されている. 次に, 獲得された特徴量を用いて分類するステップである. このステップでは, サポートベクターマシン (Support Vector Machine, SVM)[29] や線形層を用いたアプローチ [30] 等が提案されている.

3.2 プーリング

プーリングは CV の領域で活発に研究されている分野である. CV の領域では, 画像の特徴量を強調するために用いられる. NLP の領域では, 単語や文章全体の特徴量を強調するために用いられる. プーリングの利点として, 有用な特徴量を強調し, 関係のない情報を破棄するという点や単語の位置に関係なく特徴を捉えられるという点が挙げられる [31].

3.2.1 CV におけるプーリング手法

CV におけるプーリング手法について述べる. Average pooling は要素の平均を取る手法である [32] . 入力ベクトルを $[\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_N]$ とし, N をベクトルの

長さ, $\boldsymbol{r}$ の j 番目の値を $\boldsymbol{r}_j$ とすると, $\boldsymbol{r}_j$ は (3.1) 式で表すことができる.

$$\boldsymbol{r}_j = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{h}_{ij} \tag{3.1}$$

Max pooling は要素の最大値を取る手法である [33]. Max pooling は (3.2) 式で表すことができる.

$$\boldsymbol{r}_j = \max(\boldsymbol{h}_{1j}, \boldsymbol{h}_{2j} \ldots \boldsymbol{h}_{Nj}) \tag{3.2}$$

ここで $\boldsymbol{h}_{ij}$ は $\boldsymbol{h}_i$ の j 番目のベクトルである.

Yu らは, Average pooling と Max Pooling を重み付き和で組み合わせた Mixed pooling を提案した [34]. Mixed pooling は (3.3) 式で表すことができる.

$$\boldsymbol{r}_j = \lambda \max(\boldsymbol{h}_{1j}, \boldsymbol{h}_{2j} \ldots \boldsymbol{h}_{Nj}) + (1-\lambda)\frac{1}{N}\sum_{i=1}^{N} \boldsymbol{h}_{ij} \tag{3.3}$$

ここで、λ は Average pooling または Max pooling の重みを制御するハイパーパラメータである. $\lambda = 0$ の場合, Average pooling のように動作し, $\lambda = 1$ の場合, Max pooling のように動作する.

Sermanet らは L_p ノルムを用いて Average pooling と Max Pooling の重みを調整する L_p Pooling を提案した [35]. L_p Pooling は (3.4) 式で表すことができる.

$$\boldsymbol{r}_j = \left(\frac{1}{N}\sum_{i=1}^{N} {\boldsymbol{h}_{ij}}^p\right)^{\frac{1}{p}} \tag{3.4}$$

ここで、p は Average pooling と Max pooling の重みを制御するハイパーパラメータである. $p = 1$ の場合, Average pooling のように動作し, $p \to \infty$ の場合, Max pooling のように動作する.

3.2.2 NLP におけるプーリング手法

NLP におけるプーリング手法について述べる. NLP におけるプーリング手法はあまり提案されておらず, CV で用いられるプーリング手法がそのまま適用されている場合が多い. 一方で NLP において, 単語は数値的に連続したデータに直接変換することが出来ず, 言語モデルを介して数値表現に変換しても, 単語の分散表現の各次元の間には数値的な大小が存在しないという特徴があ

る. そのため, 単純にプーリングをしても意味のある特徴を獲得できるとは限らない. また, 言語モデルから得られる単語の分散表現は, それ全体で単語の意味を表現しているため, CV で用いられるプーリング手法をそのまま適用してしまうと元の情報が失われてしまい, かえってノイズになってしまう可能性がある. したがって, NLP におけるプーリング手法については, こうした問題に対して慎重に検討する必要がある. Devlin らは BERT において特殊トークンである [CLS] トークンの分散表現を文章全体の特徴として分類に用いた[3] . また, 文章全体の特徴を得るために [CLS] トークンを除いた, 単語トークン全体の平均をとる平均プーリング手法 (Average pooling) もよく用いられている.

3.3 CLS-Average Pooling

本研究の先行研究にあたる大和[1] による CLS-Average Pooling (CAP 層) について述べる. 大和は BERT の [CLS] トークンの分散表現と平均プーリング手法 (Average pooling) で獲得した分散表現を学習可能なパラメータを用いてこれらの重み付き和を文章全体を表現する分散表現とした.

モデル全体の概要を図 3.1 に示し, 図 3.2 に CAP 層の詳細を示す. 提案手法では CAP を実行する層を追加し, [CLS] トークンの分散表現と平均プーリング手法で得られた分散表現の重み付き和を計算する.

ここで, $\mathrm{Tok_i(i = \{1, \ldots, N\})}$ は入力文章をトークナイザによって単語トークンに分割した際の i 番目の単語トークンを示しており, N は文章内の単語トークンの総数を示している. また, BERT の最終層から得られる [CLS] トークンの分散表現を $E_{[\mathrm{CLS}]} \in \mathbb{R}^H$, 各単語トークンの分散表現を $E_i \in \mathbb{R}^H$, 平均プーリング手法によって獲得する分散表現 $E_{\mathrm{Avg}} \in \mathbb{R}^H$ とする. H は BERT の隠れ層の次元数である. 例として, BERT-base モデルの場合 $H = 768$ である. (3.5) 式に平均プーリング手法の計算式を示す.

$$E_{\mathrm{Avg}} = \frac{1}{N} \sum_{i=1}^{N} E_i \tag{3.5}$$

次に, p_1 を [CLS] トークンの分散表現 $E_{[\mathrm{CLS}]}$ に対する重みパラメータ, p_2 を平均プーリング手法で得られた分散表現 E_{Avg} に対する重みパラメータと

する. ここで, $p_1, p_2 \in \mathbb{R}$ は学習可能なパラメータで, 学習を通じて変化する. 実際に分類器へ入力するために, p_1, p_2 を和が 1 になるように調整する. (3.6), (3.7) 式に計算式を示す. ここで, p'_1 は p_1 の調整後の値を, p'_2 は p_2 の調整後の値を示す.

$$p'_1 = \frac{p_1^2}{p_1^2 + p_2^2} \tag{3.6}$$

$$p'_2 = \frac{p_2^2}{p_1^2 + p_2^2} \tag{3.7}$$

こうして得られた p'_1, p'_2 を用いて, [CLS] トークンの分散表現と平均プーリング手法によって獲得した分散表現 の重み付き和である CAP を計算する. CAP の計算式は以下の通りである.

$$C = p'_1 \boldsymbol{E}_{[\mathrm{CLS}]} + p'_2 \boldsymbol{E}_{\mathrm{Avg}} \tag{3.8}$$

ここで $C \in \mathbb{R}^H$ は CAP の出力によるベクトル表現である. この C を分類器への入力とし, 最終的な出力である文章の分類をする.

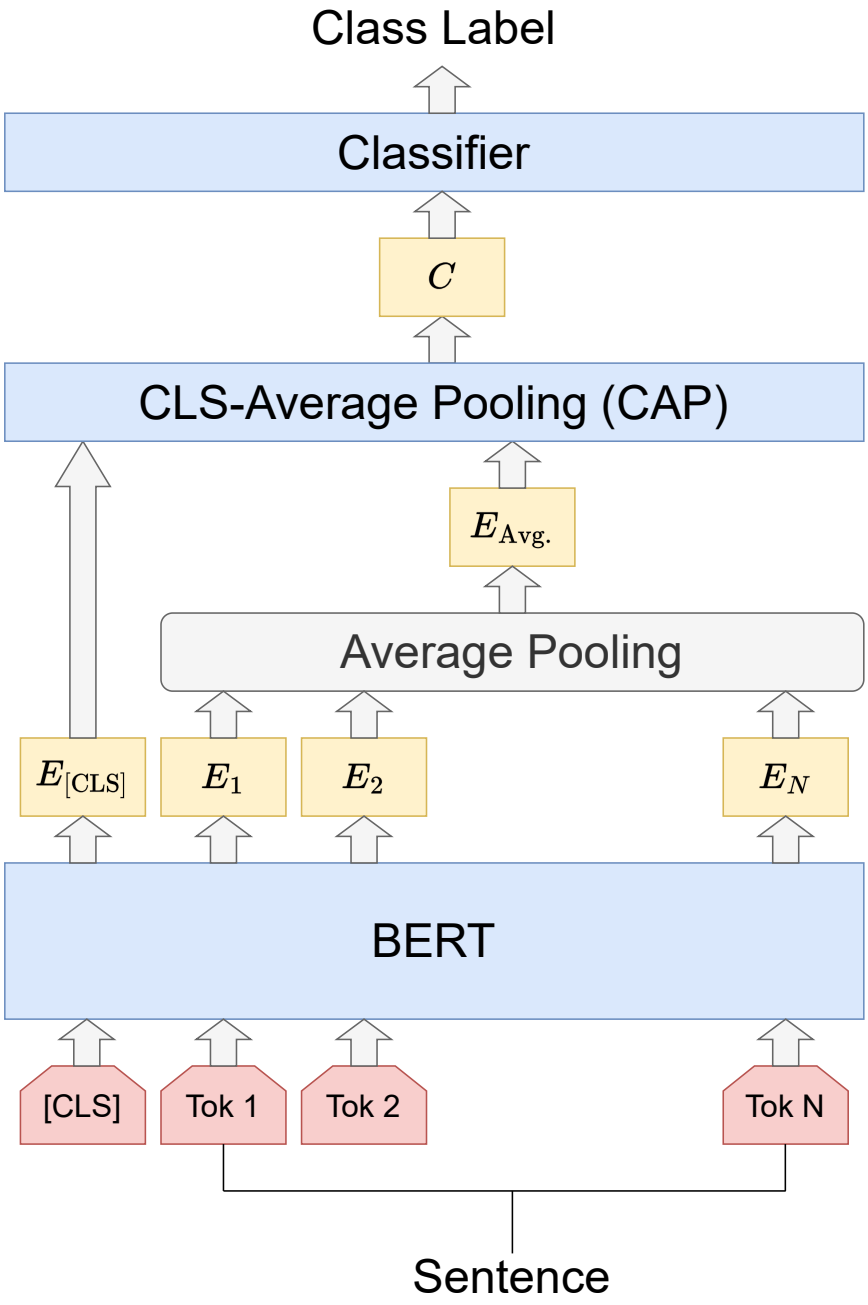

図 3.1: CAP 層を導入したモデル全体の概要 (文献 [1] より引用)

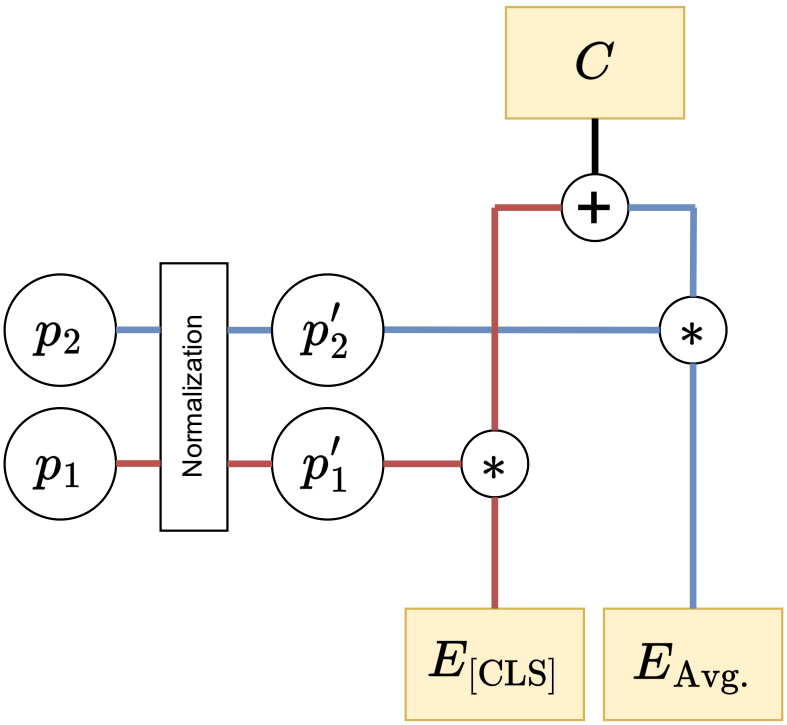

図 3.2: CAP 層の概要 (文献 [1] より引用)

4　データセット

本章では本研究の数値実験で用いたデータセットについて説明する. 本研究では大和 [1] に倣い, livedoor ニュースコーパス[1] 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 D_{label}, 記事タイトル D_{title}, 記事本文 D_{body} などのメタデータから構成されている. 表 4.1 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 4.1 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)

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

表 4.1: 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

5 提案手法

本章では, 本研究における提案モデルおよび手法についての説明をする. 本研究における提案手法は, 主に要約文の生成手法の提案, モデルおよびプーリング手法の提案に分けられる.

5.1 要約文の生成手法

本研究では, 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 5.1 に要約文の生成プロンプトの概要を示す. また, 図 5.3 および図 5.4 に生成された要約文の例を示す.

5.2 提案モデル

図 5.2 に提案モデルの概要を示す. $\mathrm{Tok_i(i = \{1, \ldots, N\})}$ は入力文章をトークナイザによってサブワードに分割した際の i 番目の単語トークンを示しており, N は文章内の単語トークンの総数を示している.

提案手法では, 図 5.2 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, [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} を並行して学習する. 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) では大和による先行研究に倣って「 [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] $\mathrm{D}_{\mathrm{summary}}$」の形式とした.

そして, 得られた 4 つの分散表現 ($E_{[\mathrm{CLS}]}$, E_{Avg}, $E'_{[\mathrm{CLS}]}$, E'_{Avg}) に対して, 後述する学習可能な重みパラメータを複数個用いたプーリング手法によって, 入力された文章データに対する分散表現 $C \in \mathbb{R}^H$ を得る.

提案手法を用いて入力された文章データの分散表現を獲得する利点としては, 大和による手法において使用していた事前学習済 BERT モデルにおける最大入力長の制限による

5.3　提案プーリング手法

本研究では, 2 つのプーリング手法によって, 入力されたデータに対する分散表現 $C \in \mathbb{R}^H$ を得る.

5.3.1　提案手法 (A)

学習可能な重みパラメータベクトルを $\mathbf{W} = (p, q, r, s), \quad (p, q, r, s \geq 0)$ とし, 入力された文章データに対する分散表現 C は (5.1) 式で与えられる.

$$C = \mathbf{W} \begin{pmatrix} E_{[\mathrm{CLS}]} \\ E_{\mathrm{Avg}} \\ E'_{[\mathrm{CLS}]} \\ E'_{\mathrm{Avg}} \end{pmatrix} = pE_{[\mathrm{CLS}]} + qE_{\mathrm{Avg}} + rE'_{[\mathrm{CLS}]} + sE'_{\mathrm{Avg}} \tag{5.1}$$

また, 重みパラメータの更新式は大和の手法に倣い, 更新後の重みベクトルを $\mathbf{W}' = (p', q', r', s')$ とすると, (5.2) 式で表される. (5.2) 式より, 重みパラメータは総和が 1 となるように学習される.

$$\mathbf{W}' = \frac{\mathbf{W} \bigodot \mathbf{W}}{\|\mathbf{W}\|^2} \tag{5.2}$$

5.3.2　提案手法 (B)

学習可能な重みパラメータベクトルを $\mathbf{W} = (p, q), \quad (p, q \geq 0)$ をとし, 入力された文章データに対する分散表現 C は (5.3) 式で与えられる. これは, それぞれの [CLS] トークンの埋め込み表現の平均ベクトルと平均プーリングによって得られた埋め込み表現の平均ベクトルの重み付き和を表している. また, 重みパラメータの更新式は (5.2) 式に準ずる.

$$C = \mathbf{W}\begin{pmatrix} \frac{E_{[\mathrm{CLS}]}+E'_{[\mathrm{CLS}]}}{2} \\ \frac{E_{\mathrm{Avg}}+E'_{\mathrm{Avg}}}{2} \end{pmatrix} = p\frac{E_{[\mathrm{CLS}]}+E'_{[\mathrm{CLS}]}}{2} + q\frac{E_{\mathrm{Avg}}+E'_{\mathrm{Avg}}}{2} \tag{5.3}$$

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

図 5.1: 提案手法における要約文の生成プロンプト

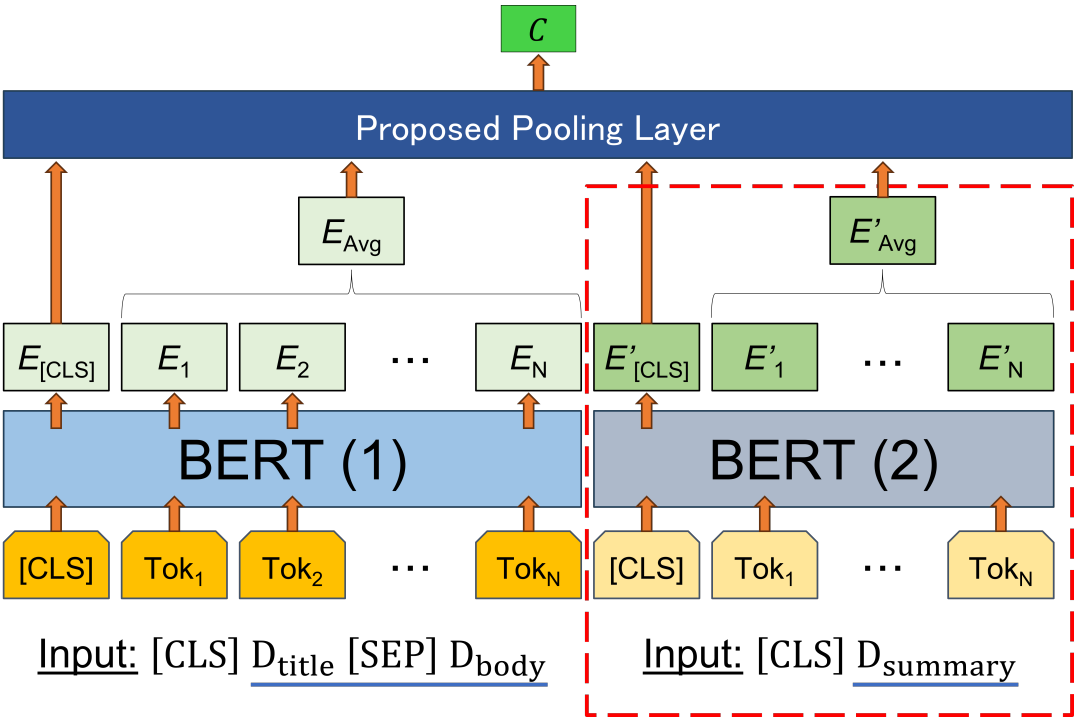

図 5.2: 提案手法モデルの概要

<原文>

【健康のために体張ります! Presented by ゆるっとcafé】

みなさま、こんにちは!
私は以前、「アラフォーだけど結婚したい!」という
コラムを書かせていただいた漫画家の尾形未紀です。
40代半ばで結婚した私たち夫婦。子供はできるかどうかまだわからないし、親ももうかなりいい年だから、あまり頼ることもできない!
だから夫婦2人でしっかり支えあって生きていかなくてはいけないと思う今日この頃...。でも、そんなときに気になるのが「健康」。
長生きしても健康じゃなきゃ、楽しくないもんね?
独女のみなさんも、年々老いてくるにしたがって、増えてくる悩みに頭を抱えてるのでは!? 疲れがとれにくくなったり、シミやしわが増えてきたり、二日酔いがなかなか治らなかったり...
さらに、病気で入院したりする友達や知り合いも増えているのでは?
体にいろいろと不具合が出始めてる頃ではないでしょうか...!?
私の担当編集のアラサー女子も、
「最近、髪が薄くなってきた...」と嘆いてます(笑)。
健康に関する悩みって、周りからは「気にしすぎ」と
思うことでも、本人にとっては真剣だものね。
そこで、みなさんの身近な健康や美容に関する悩みを
私たち夫婦が解決していく「健康コミックエッセイ」を
はじめることになりました!
少しでも健康でいて、長生きを!
それを目標にがんばりたいと思いますので、
独女のみなさんもぜひ読んでみてくださいね♪

<要約文>

この文章では、尾形未紀の新シリーズ「健康コミックエッセイ」の紹介がされています。
このエッセイでは、健康や美容に関する身近な悩みについて取り上げ、解決策を探っていきます。
尾形未紀と夫の夫婦二人三脚で、充実した健康的な生活を送るためのサポートを読者に提供する目的があるようです。具体的には、疲れやすさ、老化による見た目の変化、アルコールの影響、病気や入院に関する不安、髪の健康など、幅広いトピックが取り上げられる予定です。このエッセイは、主に独女（独身女性）の健康や美容に関する悩みに焦点を当てていますが、健康に関心のある読者にとって有益な情報を提供することを目指しています。

図 5.3: 要約文の生成例 (1), 記事ラベル : 独女通信

<原文>

【C-3POが名場面を紹介する『スター・ウォーズinコンサート』日本上陸】

2009年4月にロンドンにて世界初初演し、同年より2010年にかけて北米、欧州ツアーを行い、175万人以上を動員した『スター・ウォーズinコンサート』の、日本公演が遂にアジア初として決定した。
通常は、1万5千人規模のアリーナ級の会場で行われていたが、今回は演出と音響の質を最大限に引き出すため、約4500人規模のホール級会場での公演が実現。更なる臨場感と興奮が味わえるショーとなり、目と耳の肥えた日本のファンに向けたプレミアムなコンサートとなる。
『スター・ウォーズinコンサート』は特別に編集された映画の名場面を、巨大LEDスクリーンに映し、ジョン・ウィリアムズのスリリングな楽曲の演奏にライブでシンクロする。また、全6作でC-3PO役を演じたアンソニー・ダニエルズが、名場面のナレーションを行うことにより、スター・ウォーズに新しい生命を吹き込み、観客を興奮の世界と誘う。
東京公演は、8月9日から8月12日に東京国際フォーラム ホールAで、大阪公演は、8月15日にグランキューブ大阪で行われる。詳細は、公式サイトで確認できる。
ジョージ・ルーカスが全精力を傾注し、最新テクノロジーを駆使した3D映画として、再び大スクリーンに帰ってくることで注目される『STAR WARS エピソード1/ファントム・メナス 3D』の公開も3月16日と迫っている。今年は「スターウォーズ」の熱が全世界を包むことだろう。

<要約文>

『スター・ウォーズinコンサート』アジア初の日本公演が東京と大阪で開催されます。このコンサートは、映画の名場面を巨大LEDスクリーンで映し、ジョン・ウィリアムズの楽曲の演奏に合わせて楽しむものです。特別ゲストとして、全6部作でC-3POを演じたアンソニー・ダニエルズが名場面のナレーションを行います。東京公演は8月9日から12日まで東京国際フォーラム ホールA、大阪公演は8月15日にグランキューブ大阪で行われます。ジョージ・ルーカスが監督した『スター・ウォーズ エピソード1/ファントム・メナス 3D』の公開も3月16日に予定されており、今年はスター・ウォーズの熱が世界中を包み込むでしょう。

図 5.4: 要約文の生成例 (2), 記事ラベル : MOVIE ENTER

6　数値実験

本章では, 提案手法の有効性を確認するために実施した実験について説明する.

6.1　実験 (1): livedoor ニュースデータコーパスを用いたテキスト分類

6.2　実験 (2): TF-IDF スコアを用いた要約文情報の統合実験

7　まとめと今後の課題

謝辞

本研究を進めるにあたり, 多大なるご指導とご助言を賜りました森直樹教授に心より深く感謝申し上げます. また, 本研究において直接ご指導いただいた岡田真助教には, 研究のアイデアや方針に関する助言のみならず, 論文の執筆や発表の作法に至るまで, 日々多岐にわたる貴重なご指導を頂戴いたしました. この場を借りて厚く御礼申し上げます.

最後に, 研究活動を通じて建設的な意見を寄せてくださった諸先輩方, 後輩の皆様, そして共に励まし合いながら研究に取り組んできた同期の皆さんにも, 心からの感謝を申し上げます.

2025 年 2 月 10 日

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] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need, 2017.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. 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.

[4] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le. *XLNet: generalized autoregressive pretraining for language understanding*. Curran Associates Inc., Red Hook, NY, USA, 2019.

[5] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1), jan 2020.

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

[7] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin eds., *Advances in Neural Information Processing Systems*, Vol. 33, pp. 1877–1901. Curran Associates, Inc., 2020.

[8] 森田 一, 黒橋 禎夫. RNN 言語モデルを用いた日本語形態素解析の実用化. 第 78 回全国大会講演論文集, 2016(1):13–14, mar 2016.

[9] 工藤, 山本, 松本. Conditional random fields を用いた日本語形態素解析. 情報処理学会研究報告. NL, 自然言語処理研究会報告, 第 161 巻, pp. 89–96. 一般社団法人情報処理学会, may 2004.

[10] K. Takaoka, S. Hisamoto, N. Kawahara, M. Sakamoto, Y. Uchida, and Y. Matsumoto. Sudachi: a japanese tokenizer for business. In N. C. C. chair), K. Choukri, C. Cieri, T. Declerck, S. Goggi, K. Hasida, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis, and T. Tokunaga eds., *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France, may 2018. European Language Resources Association (ELRA).

[11] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger eds., *Advances in Neural Information Processing Systems*, Vol. 26. Curran Associates, Inc., 2013.

[12] Q. Le and T. Mikolov. Distributed representations of sentences and documents. In *International conference on machine learning*, pp. 1188–1196. PMLR, 2014.

[13] R. Kiros, Y. Zhu, R. Salakhutdinov, R. S. Zemel, A. Torralba, R. Urtasun, and S. Fidler. Skip-thought vectors. *arXiv preprint arXiv:1506.06726*, 2015.

[14] J. Pennington, R. Socher, and C. Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, Doha, Qatar, Oct. 2014. Association for Computational Linguistics.

[15] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations, 2018.

[16] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate, 2016.

[17] M.-T. Luong, H. Pham, and C. D. Manning. Effective approaches to attention-based neural machine translation, 2015.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc.

[19] P. E. K. Abe, K. Chubachi, Y. Fujita, Y. Hirokawa, K. Imajo, T. Kataoka, H. Komatsu, H. Mikami, T. Mogami, S. Murai, K. Nakago, D. Nishino, T. Ogawa, D. Okanohara, Y. Ozaki, S. Sano, S. Suzuki, T. Xu, and T. Yanase. PLaMo-100B: A ground-up language model designed for japanese proficiency. *arXiv preprint arXiv:2410.07563*, 2024.

[20] K. Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21, 1972.

[21] K. S. Jones and P. Willett. *Readings in Information Retrieval*. Morgan Kaufmann, 2004.

[22] O. undefinedrsoy and C. Cardie. Deep recursive neural networks for compositionality in language. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pp. 2096–2104, Cambridge, MA, USA, 2014. MIT Press.

[23] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1480–1489, San Diego, California, June 2016. Association for Computational Linguistics.

[24] Y. Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1746–1751, Doha, Qatar, Oct. 2014. Association for Computational Linguistics.

[25] J. Wei and K. Zou. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. 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. 6382–6388, Hong Kong, China, Nov. 2019. Association for Computational Linguistics.

[26] L. Yao, C. Mao, and Y. Luo. Graph convolutional networks for text classification. *CoRR*, abs/1809.05679, 2018.

[27] J. Howard and S. Ruder. Fine-tuned language models for text classification. *CoRR*, abs/1801.06146, 2018.

[28] X. Sun, X. Li, J. Li, F. Wu, S. Guo, T. Zhang, and G. Wang. Text classification via large language models, 2023.

[29] Y. Arslan, K. Allix, L. Veiber, C. Lothritz, T. F. Bissyandé, J. Klein, and A. Goujon. A comparison of pre-trained language models for multi-class text classification in the financial domain. In *Companion Proceedings of the Web Conference 2021*, WWW '21, pp. 260–268, New York, NY, USA, 2021. Association for Computing Machinery.

[30] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pp. 427–431, Valencia, Spain, Apr. 2017. Association for Computational Linguistics.

[31] H. Gholamalinezhad and H. Khosravi. Pooling methods in deep neural networks, a review. *CoRR*, abs/2009.07485, 2020.

[32] Y. Le Cun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Handwritten digit recognition with a back-propagation network. In *Proceedings of the 2nd International Conference on Neural Information Processing Systems*, NIPS'89, pp. 396–404, Cambridge, MA, USA, 1989. MIT Press.

[33] M. A. Ranzato, Y.-L. Boureau, and Y. LeCun. Sparse feature learning for deep belief networks. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, NIPS'07, pp. 1185–1192, Red Hook, NY, USA, 2007. Curran Associates Inc.

[34] D. Yu, H. Wang, P. Chen, and Z. Wei. Mixed pooling for convolutional neural networks. In *Rough Sets and Knowledge Technology*, 2014.

[35] P. Sermanet, S. Chintala, and Y. LeCun. Convolutional neural networks applied to house numbers digit classification. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pp. 3288–3291, 2012.