事前学習済みBERTモデル検索タスクのための評価データセット

ファムフーロン† 三林 亮太†† 荘司 慶行††† 加藤 誠†††,††††† 山本 岳洋†
山本 祐輔†††††† 大島 裕明†

† 兵庫県立大学 情報科学研究科 〒651–2197 兵庫県神戸市西区学園西町 8–2–1
†† 神戸大学 国際文化学研究科 〒651–8501 兵庫県神戸市灘区鶴甲 1–2–1
††† 静岡大学 情報学部 〒432–8011 静岡県浜松市中央区城北 3–5–1
†††† 筑波大学 図書館情報メディア系 〒305 – 8550 茨城県つくば市春日 1–2
††††† 国立情報学研究所 情報社会相関研究系 〒101–8430 東京都千代田区一ツ橋 2–1–2
†††††† 名古屋市立大学 データサイエンス研究科 〒467–8501 愛知県名古屋市瑞穂区瑞穂町字山の畑 1
E-mail: †{af23a009@guh.u-hyogo.ac.jp, t.yamamoto@sis.u-hyogo.ac.jp, ohshima@ai.u-hyogo.ac.jp},
††mibayashi@people.kobe-u.ac.jp, †††shojiy@inf.shizuoka.ac.jp, {††††,†††††}mpkato@acm.org,
†††††yusuke_yamamoto@acm.org

あらまし 本研究では，事前学習済みBERTモデル検索タスクのための評価データセットを構築する．自然言語処理では，事前学習済みモデルをタスクに合わせてファインチューニングして利用することが一般的であるが，多数の候補から目的タスクに適したモデルを選択することは容易ではない．モデルの適性は実際にファインチューニングを行うまで判別しにくく，すべての候補を試行するには多大な時間と計算資源を要するため，タスクに適したモデルを効率的に探索できる仕組みが不可欠である．そこで本研究では，48件の文書分類タスクと20種類のBERTモデルの組合せからなる評価データセットを作成した．さらに，既存手法をベースラインとして実装するとともに，タスクと検索対象モデルの特徴を用いた検索手法を提案し，構築したデータセットを用いてその有効性を検証した．

キーワード BERTモデル検索，言語モデル，検索モデル，テキスト分類

1 はじめに

事前学習済みBERTモデルは，文書分類や質問応答など，多岐にわたる自然言語処理タスクで用いられている．新たなタスクに取り組む際，ゼロからモデルを学習する代わりに，既存の事前学習済みモデルをファインチューニングして用いることが一般的である．しかし，特定のタスクに最も適した事前学習済みモデルを選択することは容易ではない．

通常，特定のタスクに対する事前学習済みモデルの適合性は，ファインチューニングを通じて評価される．一般的なモデル選定のアプローチは以下の通りである．

1. 多数の事前学習済みモデルの中からいくつかの候補を選択する
2. 1で選択したモデルをタスクの一部のデータ（訓練データ等）を用いてファインチューニングを行う
3. タスクの残りのデータ（テストデータ等）を用いて，2で学習したモデルの性能を評価する
4. 最も性能が良いモデルを採用する

このプロセスにおける最大の問題は，ファインチューニングに要する計算コストの高さである．Hugging Face Hubでは2025年12月24日時点で46,337の事前学習済みBERTモデルが公開されており，これら全てのモデルに対してファインチューニングを行い比較することは不可能である．そのため，実際に学習を行うことなく，事前学習済みBERTモデルを効率的に検索・選択する手法が必要とされている．

事前学習済みモデルの検索あるいは選択に関する問題は，近年活発に研究されており，*Transferability Estimation*などの名称で呼ばれている．しかし，現時点では研究分野として十分に体系化されておらず，問題に対する統一的な定義もなされていない．また，手法の性能を定量的に比較するための標準的なベンチマークデータセットも公開されていないのが現状である．

そこで本研究では，事前学習済みBERTモデル検索という問題を体系的に定義する．また，本タスクの評価基盤として，48件のテキスト分類タスクと20種類の事前学習済みBERTモデルからなる新たなデータセットを構築した．さらに，既存の主要なモデル検索手法をベースラインとして評価するとともに，Prompt Tuningによるタスク表現とモデルの埋め込みベクトルをTransformer Encoderで統合する新たなモデル検索手法を提案する．

なお，本研究で定義している事前学習済みBERTモデル検索タスクは，国際会議NTCIR-19 [1]において，*ModelRetrieval*タスクのサブタスクとして実施している [2]．

1：https://research.nii.ac.jp/ntcir/ntcir-19/index-ja.html
2：https://modeltreival.github.io/modelretrieval-1/

2 関連研究

2.1 事前学習済みモデルとドメイン特化型モデル

現代の機械学習において，下流タスクに適応させた事前学習済みモデルの利用は不可欠となっている．自然言語処理分野では，BERT [9] が Transformer アーキテクチャ [24] を用いた大規模事前学習の有効性を確立した．その後継モデルは，さらなる効率と精度の向上を実現している．例えば，RoBERTa は Next Sentence Prediction（NSP）タスクを排除し，データ規模を拡大した [17]．ALBERT は，埋め込み行列の分解と層間でのパラメータ共有によりモデルパラメータを削減した [14]．また，DeBERTa は Disentangled Attention と強化されたデコーディング機構を導入している [12]．さらに，DistilBERT [23] のように，知識蒸留を用いて性能を維持しつつモデルを軽量化する手法も提案されている．

ドメイン特化は，転移学習の効果をさらに高める．具体例として，科学文献向けの SciBERT [3]，金融ドメイン向けの FinBERT [18]，攻撃的な表現を扱う HateBERT [5]，そして法務コーパス向けの LEGAL-BERT [7] などが挙げられる．生物医学分野では，BioBERT [15]，ClinicalBERT [13]，MedBERT [21] などがドメイン固有のコーパスを活用している．また，インターネット上のテキスト分布に適応させたモデルとして IMHO [6] や BERTweet [19] がある．ドメイン適応事前学習（Domain-adaptive pre-training）は，ラベルなしデータを用いて一般コーパスから新規ドメインへの橋渡しを行う有効な手段である [11]．

2.2 機械学習モデル検索 (Machine Learning Model Retrieval)

新たなタスクに対して適切なモデルを選択する研究は，「モデル選択（Model selection）」，「転移性推定（Transferability estimation）」，「ニューラルネットワーク検索（Neural network retrieval）」など様々な名称で呼ばれており，多岐にわたるタスク，分野，モダリティにまたがっている．

例えば自然言語処理分野では，Safikhani ら [22] がテキスト分類のためのドメインを考慮した事前学習済みモデル検索を研究しており，Dai ら [8] は検索タスク（Retrieval tasks）における転移性を評価している．コンピュータビジョン分野では，Bolya ら [4] や Nermeen ら [1] が画像分類モデルの検索を対象としているほか，Fouquet ら [10] は物体検出，Yang [25] はセグメンテーションタスクを扱っている．

また，情報検索の国際会議である TREC 2025 においても，Million LLMs Track [3]が実施されている．同トラックは，ユーザの任意の質問（プロンプト）に対して最適な回答を生成する大規模言語モデル（LLM）を検索するタスクであり，生成 AI 時代におけるモデル選択の重要性が広く認識されつつあることがわかる．

先行研究ではタスクやモダリティによって異なる用語が用いられているが，これらは「与えられたタスクと制約条件の下で，候補モデルを期待される性能順にランク付けする」という共通の目的を持っている．本研究では，この視点を「機械学習モデル検索（Machine Learning Model Retrieval）」と定義する．

3：https://trec-mllm.github.io/

3 問題定義

事前学習済み BERT モデル検索タスクを定義する．まず，検索対象となる K 個の事前学習済みモデルの集合を $\mathcal{M}$ とする．

$$\mathcal{M} = \{m_1, m_2, \ldots, m_K\}$$

ここで，各モデル m_k は特定のアーキテクチャと事前学習済みのパラメータを持つ．

次に，検索クエリとなるターゲットタスク $\mathcal{T}$ を定義する．本研究において，タスク $\mathcal{T}$ は訓練データ，検証データ，テストデータの 3 つの集合から構成されるとする．

$$\mathcal{T} = \{D_{\text{train}}, D_{\text{val}}, D_{\text{test}}\}$$

ここで，各データセットは入力文書 x とラベル y のペアの集合 $\{(x_i, y_i)\}_{i=1}^{N}$ である．

モデル検索タスクの目的は，ターゲットタスク $\mathcal{T}$ に対して，最も高い性能を発揮するモデル $m \in \mathcal{M}$ を特定することである．

あるモデル m_k を $\mathcal{T}$ の訓練データ D_{train} および検証データ D_{val} を用いてファインチューニングして得られるモデルを m'_k とする．このとき，テストデータ D_{test} におけるモデルの真の性能を $S(m'_k, D_{\text{test}})$ と定義する．本研究では評価指標として Macro F1 値を用いる．

検索システム f は，テストデータ D_{test} を参照することなく，以下の推定スコア $\hat{s}_k$ を算出するものと定義する．

$$\hat{s}_k = f(m_k, D_{\text{train}}, D_{\text{val}})$$

算出されたスコア $\hat{s}_k$ に基づくモデルのランク付けが，真の性能 $S(m'_k, D_{\text{test}})$ に基づくランク付けに近いほど検索性能が良い．

4 事前学習済み BERT モデル検索タスクのための評価データセット

本研究では，事前学習済み BERT モデル検索タスクの評価用データセットとして，48 件の文書分類タスクと 20 種類の異なる事前学習済み BERT モデルからなるベンチマークを構築した．各タスクに対して全モデルを適用し，正解となるランクリストを作成した．この正解リストと検索結果を比較することで，検索手法の定量的な評価が可能となる．

4.1 文書分類タスク

ベンチマークデータセットに含まれる文書分類タスクの一部を表 1 に示す．これらのタスクは Hugging Face Datasets で公開されており，`https://huggingface.co/datasets/<dataset_name>` からアクセス可能である．各タスクはラベル数やデータ規模が

異なっている．

各タスクのデータセットは，**訓練データ**，**検証データ**，**テストデータ**の 3 つに分割して使用する．各タスクのデータ分割は以下のルールに基づいて行った．

- Hugging Face 上ですでに訓練・検証・テストデータに分割されている場合：その分割をそのまま利用する．
- 訓練セットとテストセットのみに分割されている場合：元の訓練データの 10%を検証データとして切り出し，残りを訓練データとする．
- 訓練データのみが提供されている場合：全データの 10%を検証用，10%をテスト用として切り出し，残りの 80%を訓練データとする．

さらに，実験における計算コストを抑制するため，各データ分割のサイズに上限を設定した．具体的には，上記の手順で分割された訓練，検証，およびテストデータのサンプル数がそれぞれ 5,000 件を超える場合，ランダムサンプリングによって各 5,000 件にした．

4.2 検索対象 BERT モデル

検索対象となる事前学習済み BERT モデルの一部を表 2 に示す．これらのモデルも Hugging Face で公開されており，`https://huggingface.co/<model_name>` から利用可能である．各モデルは，アーキテクチャ（BERT，RoBERTa など），パラメータ数，および事前学習やファインチューニングに使用されたデータセットが異なっており，これらが特定の文書分類タスクに対する適合性に影響を与えると考えられる．例えば，ツイートデータで学習されたモデルは，一般的な Web テキストで訓練されたモデルと比較して，ツイート関連タスクにおいてより高い性能を発揮する可能性がある．

4.3 正解ランクリストの作成

各タスクに対する事前学習済み BERT モデルの正解ランクリスト $\mathbf{m} = (m_1, m_2, \ldots, m_k)$ は，以下の手順で作成した．

1. タスクの訓練データと検証データを用いて，すべての事前学習済みモデルに対してファインチューニングを行う．
2. タスクのテストデータを用いて，ファインチューニング済みモデルの性能を Macro F1 値で評価する．
3. テストデータにおける Macro F1 値に基づいて，モデルを降順にランク付けする．

実験環境として，4 基の NVIDIA RTX 3090 GPU (24GB) を使用した．ハイパーパラメータは，バッチサイズを 16，学習率を 2×10^{-5}，最適化手法に AdamW，Weight Decay を 0.01 とした．損失関数にはクロスエントロピー誤差を用い，検証データの損失に基づいた Early Stopping（patience=10）を適用した．

4.4 タスクの重要度と分類

モデル選択の難易度や重要性はタスクごとに異なる．そこで，各タスクにおけるモデル検索の重要度を定量化するため，**期待リグレット（Expected Regret）**を導入する．

まず, タスク $\mathcal{T}$ におけるモデル m の Macro F1 値を $S(m, \mathcal{T})$ とし, タスク内の最大性能で正規化した値を正規化性能 $\bar{S}(m, \mathcal{T})$ と定義する．

$$\bar{S}(m, \mathcal{T}) = \frac{S(m, \mathcal{T})}{\max_{m' \in \mathcal{M}} S(m', \mathcal{T})} \tag{1}$$

この正規化性能を用いて，候補モデル集合 $\mathcal{M}$ からランダムにモデルを選択した場合に，最適なモデルと比較して平均的にどの程度の性能損失が生じるかを表す期待リグレット $R_{\text{exp}}(\mathcal{T})$ を以下のように定義する．

$$R_{\text{exp}}(\mathcal{T}) = 1 - \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} \bar{S}(m, \mathcal{T}) \tag{2}$$

R_{exp} が大きいタスクは，モデルによる性能差が大きく，適切なモデルを選択できなかった場合の損失が大きいため，モデル検索の重要度（Criticalness）が高いと言える．本研究では，このスコアに基づいてタスクを以下の 3 つのカテゴリーに分類した．

- **Low criticalness**: $R_{\text{exp}} < 0.03$
- **Medium criticalness**: $0.03 \leqq R_{\text{exp}} < 0.10$
- **High criticalness**: $0.10 \leqq R_{\text{exp}}$

4.5 実験設定と評価指標

構築した 50 件のタスクを，Low, Medium, High criticalness の各カテゴリーの割合が均等になるように，検索モデル学習用の訓練タスクセット（25 件）と評価用のテストタスクセット（25 件）に分割した．

検索手法の評価指標として，nDCG@k（Normalized Discounted Cumulative Gain）を用いる．上位 k 件の検索結果に対する DCG@k は以下のように計算する．

$$\text{DCG@}k = \sum_{i=1}^{k} \frac{\text{rel}_i}{\log_2(i+1)} \tag{3}$$

ここで，順位 i のモデルの適合スコア rel_i は，そのモデルの正規化性能 $\bar{S}$ に基づいて以下のように定義する．

$$\text{rel}_i = \begin{cases} 4 & (0.975 \leqq \bar{S} \leqq 1.000) \\ 3 & (0.950 \leqq \bar{S} < 0.975) \\ 2 & (0.925 \leqq \bar{S} < 0.950) \\ 1 & (0.900 \leqq \bar{S} < 0.925) \\ 0 & (0.000 \leqq \bar{S} < 0.900) \end{cases} \tag{4}$$

最終的な nDCG@k は，理想的なランキングの IDCG@k を用いて計算する．

表 1 本研究で構築したベンチマークデータセットに含まれる一部の文書分類タスク．データ数は前処理（上限 5,000 件）後の最終的な事例数を示す．タスク名の右肩の記号はデータ分割方法の差異を表す（無印：公式分割，*：訓練データから検証データを分割，**：訓練データから検証・テストデータを分割）．

ID	タスク名	サブセット	ラベル数	データ数		
				訓練	検証	テスト
1	cardiffnlp/tweet_eval	emoji	20	5,000	5,000	5,000
17	stanfordnlp/sst2**	–	2	5,000	5,000	5,000
50	toxigen/toxigen-data*	annotated	5	5,000	896	940
...	...	...	...	...	...	...

表 2 検索対象となる事前学習済み BERT モデルの一部．各モデルのアーキテクチャおよび事前学習に使用されたデータセット情報を示す．

ID	モデル名	事前学習データ	事前ファインチューニング
	BERT 系モデル		
4	GroNLP/hateBERT	BookCorpus, English Wikipedia, RAL-E	–
...	...	...	...
	DistilBERT 系モデル		
11	distilbert-base-uncased-distilled-squad	BookCorpus, English Wikipedia	SQuAD
...	...	...	...
	RoBERTa 系モデル		
15	zhayunduo/roberta-base-stocktwits-finetuned	BookCorpus, English Wikipedia, CC-News, OpenWebText, Stories	StockTwits
...	...	...	...
	DistilRoBERTa 系モデル		
18	j-hartmann/emotion-english-distilroberta-base	BookCorpus, English Wikipedia, CC-News, OpenWebText, Stories, Tweets	Crowdflower, Emotion, GoEmotions, ...
...	...	...	...
	その他のモデル (ALBERT, DeBERTa)		
20	microsoft/deberta-base	BookCorpus, English Wikipedia, OpenWebText, Stories	–
...	...	...	...

$$\mathrm{nDCG@}k = \frac{\mathrm{DCG@}k}{\mathrm{IDCG@}k} \tag{5}$$

本研究では，$k \in \{1, 3\}$ で評価を行う．

5 検 索 手 法

本節では，事前学習済みモデルの検索手法を分類し，実験に用いる既存手法および提案手法ついて説明する．

5.1 検索手法の分類

既存のモデル検索手法は，検索時に候補モデルへのデータ入力（推論）を必要とするか否かによって，主に**推論あり（Inference-based）**と**推論なし（Inference-free）**の 2 種類に分類することができる．

a） 推論ありの手法

クエリタスクのデータ $\mathcal{T}$（訓練データ D_{train} 等）を候補モデル $m \in \mathcal{M}$ に入力し，得られた特徴量 $f(x; m)$ を用いてスコアを算出する手法である．候補モデル数 K に比例して計算コストが増大するが，タスクとモデルの適合性をデータに基づいて直接的に評価できるため，推定精度が高い傾向にある．

b） 推論なしの手法

検索時に，クエリタスクのデータを候補モデルに入力することなく検索を行う手法である．モデルの特性を表すベクトルを事前に計算しておきタスク特性との類似度を計算するアプローチや，過去のタスクにおける性能統計のみを利用するアプローチなどがある．個々のモデルに対する推論コストが発生しないため，高速な検索が可能である．

5.2 既存手法手法

5.2.1 LogME

LogME (Logarithm of Maximum Evidence) [26] は，**推論あり**の代表的な手法である．本手法は，事前学習済みモデルを固定的な特徴抽出器と見なし，その出力特徴量に対する線形回

帰モデルの周辺尤度（Evidence）をスコアとして用いる．

具体的には，モデル m を用いて訓練データ D_{train} および検証データ D_{val} から抽出した特徴行列を $\mathbf{F}$，ラベルに対応するターゲット（分類タスクの場合は One-hot ベクトル等）を $\mathbf{y}$ とする．線形回帰の重みを $\mathbf{w}$，観測ノイズの精度を β，重みの事前分布の精度を α としたとき，LogME スコア $S_{\text{LogME}}(m)$ は，以下の周辺尤度 $p(\mathbf{y}|\mathbf{F},\alpha,\beta)$ の最大値として定義される．

$$S_{\text{LogME}}(m) = \max_{\alpha,\beta} \log \int p(\mathbf{y}|\mathbf{F},\mathbf{w},\beta)p(\mathbf{w}|\alpha)d\mathbf{w} \tag{6}$$

この値が大きいほど，モデルの特徴空間がタスクのラベル予測に適しており，ファイチューニング後の性能が高くなるとされる．

5.2.2 H-Score

H-Score [2] は，特徴空間における情報の分離度合いを測定する**推論あり**の手法である．この手法は，特徴量の冗長性を考慮しつつ，クラス間の分散が全分散に対してどれだけの割合を占めるかを評価する．

モデル m によって抽出されたデータ x_i の特徴量を $\mathbf{f}_i = f(x_i;m)$ とする．全データの平均ベクトルを $\boldsymbol{\mu}$，全共分散行列を $\boldsymbol{\Sigma}_{\text{tot}}$ とする．また，クラス c に属するデータの平均ベクトルを $\boldsymbol{\mu}_c$，N_c をクラス c のデータ数としたとき，クラス間共分散行列 $\boldsymbol{\Sigma}_B$ は以下のように定義される．

$$\begin{aligned}\boldsymbol{\Sigma}_{\text{tot}} &= \frac{1}{N}\sum_i (\mathbf{f}_i - \boldsymbol{\mu})(\mathbf{f}_i - \boldsymbol{\mu})^\top \\ \boldsymbol{\Sigma}_B &= \sum_c \frac{N_c}{N}(\boldsymbol{\mu}_c - \boldsymbol{\mu})(\boldsymbol{\mu}_c - \boldsymbol{\mu})^\top\end{aligned} \tag{7}$$

H-Score $S_{\text{H}}(m)$ は，全共分散行列の逆行列とクラス間共分散行列の積のトレースとして定義される．

$$S_{\text{H}}(m) = \text{Tr}(\boldsymbol{\Sigma}_{\text{tot}}^{-1}\boldsymbol{\Sigma}_B) \tag{8}$$

値がほどタスクに適していることを示す．

5.2.3 k 近傍法

k 近傍法（k-NN）を用いた手法 [20] は，モデルの特徴空間上での近傍探索により性能を推定する**推論あり**の手法である．

まず, モデル m を用いて, 訓練データ $(x_i^{\text{train}}, y_i^{\text{train}}) \in D_{\text{train}}$ および検証データ $(x_j^{\text{val}}, y_j^{\text{val}}) \in D_{\text{val}}$ の特徴ベクトル $\mathbf{d}_i^{\text{train}}, \mathbf{d}_j^{\text{val}}$ をそれぞれ計算する．次に，訓練データの特徴ベクトルを参照集合として，検証データの各事例に対して k-NN による分類を行う．

検証事例 j に対する予測ラベルを $\hat{y}_j$ としたとき，スコア $S_{\text{kNN}}(m)$ は検証データセット全体に対する予測の Macro F1 値として定義される．

$$S_{\text{kNN}}(m) = \frac{1}{|D_{\text{val}}|}\sum_{j=1}^{|D_{\text{val}}|} \mathbb{I}(\hat{y}_j = y_j^{\text{val}}) \tag{9}$$

ここで，$\mathbb{I}(\cdot)$ は条件が真の場合に 1，偽の場合に 0 を返す指示関数である．

5.2.4 ModelSpider

ModelSpider [27] は，検索時に対象モデルへの推論を必要としない**推論なし**の手法である．本手法では，各事前学習済みモデル $m \in \mathcal{M}$ の特性を表すモデルベクトル $\mathbf{v}_m$ を事前に抽出・保存しておく．

検索時には，まずクエリタスクの訓練データ D_{train} からタスクの特性を表す**タスク行列** $\mathbf{V}_{\text{task}}$ を構築する．これは，各クラスラベル c に属するサンプルの特徴量の平均ベクトル（重心）$\mathbf{c}_c \in \mathbb{R}^d$ を行ベクトルとして積み重ねた行列として定義される．

$$\mathbf{V}_{\text{task}} = [\mathbf{c}_1, \ldots, \mathbf{c}_C]^\top \in \mathbb{R}^{C\times d} \tag{10}$$

ここで C はクラス数，d は特徴量の次元数である．

タスクに対するモデルのスコア $S_{\text{Spider}}(m)$ は，タスク行列 $\mathbf{V}_{\text{task}}$ とモデルベクトル $\mathbf{v}_m$ を入力とし，学習済みのスコアリング関数 ϕ を用いて以下のように算出される．

$$S_{\text{Spider}}(m) = \phi(\mathbf{v}_m, \mathbf{V}_{\text{task}}) \tag{11}$$

ここで関数 ϕ は，1 層の Transformer Encoder および全結合層からなるニューラルネットワークである．

5.2.5 Average Rank

Average Rank は，訓練タスクにおけるモデルの性能統計のみを利用する**推論なし**のベースライン手法である．具体的には，訓練タスク $\mathcal{T}_{\text{train}}$ を用いて，各モデル $m \in \mathcal{M}$ の平均ランクをスコア $S_{\text{Avg}}(m)$ として算出する．

$$S_{\text{Avg}}(m) = \frac{1}{|\mathcal{T}_{\text{train}}|}\sum_{\mathcal{T}\in\mathcal{T}_{\text{train}}} \text{Rank}(m,\mathcal{T}) \tag{12}$$

ここで $\text{Rank}(m,\mathcal{T})$ はタスク $\mathcal{T}$ におけるモデル m の正解順位である．検索時には，ターゲットタスクの内容（テキストやラベル）に依存せず，常にこの $S_{\text{Avg}}(m)$ の昇順（平均順位が良い順）にソートされた固定のランクリストを出力する．

5.3 提案手法

本研究では，新たな**推論なし**の手法を提案する．図 1 に示すように，本手法では，クエリとなるターゲットタスクの特徴量と，検索候補となる各モデルの埋め込みベクトルを統合し，その適合度スコアを推定するものである．以下，学習フェーズと推論フェーズについて詳細を述べる．

5.3.1 学習フェーズ

学習フェーズでは，タスクとモデルのペアを入力として，そのタスクに対するモデルの性能を精度よく予測するように最適化を行う．

a） タスクエンベディング

クエリタスク $\mathcal{T}$ の特徴量を抽出するため，Zhou らによって提案された TuPaTE [28] に基づく手法を用いる．具体的には，

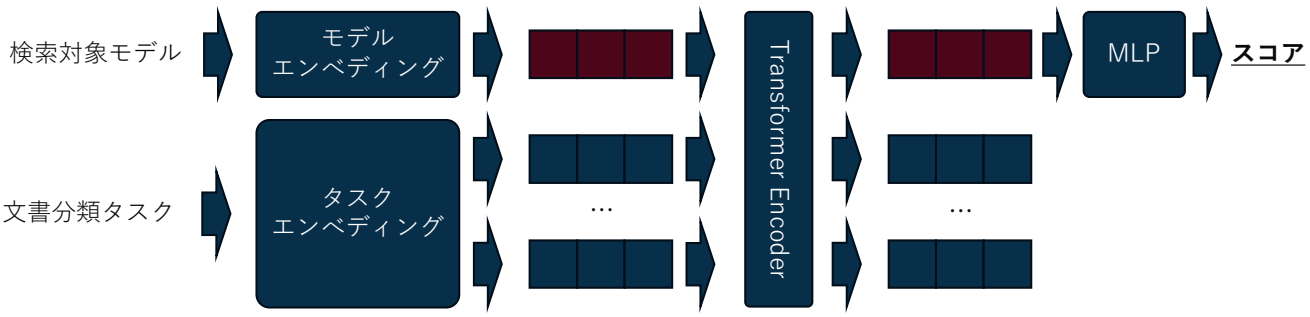

図 1　提案手法の概要：クエリタスクから Prompt Tuning により抽出されたタスクエンベディングと，各候補モデルの学習可能な埋め込みベクトルを Transformer Encoder で統合し，MLP を介して適合度スコアを予測する．

任意の言語モデルに対し，Prompt Tuning [16] を適用する．

まず，タスクの入力テキストの先頭に $P = 20$ 個の学習可能な特殊トークン（Soft Prompts）を付加し，拡張された入力系列を構成する．この系列をモデルに入力し，得られた `[CLS]` トークンのベクトルを分類器に通してラベルを予測する．この際，Transformer のパラメータは固定し，特殊トークンの埋め込み行列と分類器のみを更新対象とする．学習後，得られた P 個の特殊トークンの埋め込みベクトル集合を，タスク $\mathcal{T}$ の特徴量行列 $\mathbf{V}_{\mathcal{T}}^{(\mathrm{tsk})} \in \mathbb{R}^{P\times 768}$ とする．

本研究では, 事前学習済みの `bert-base-multilingual-uncased` に対しクエリタスクの訓練データを用いて Prompt Tuning を行った．エポック数を 20、最適化を AdamW、学習率を 1e-3 とした．Prompt Tuning の学習時間は一つのタスクにおいて平均として約 2 分間程度であった．

b） モデルエンベディング

検索対象となる各モデル $m_k \in \mathcal{M}$ に対して，ランダムに初期化された学習可能な埋め込みベクトル $\mathbf{v}_k^{(\mathrm{mdl})} \in \mathbb{R}^{768}$ を割り当てる．

c） スコア予測

あるタスク $\mathcal{T}$ とモデル m_k の適合度を算出するため，タスクエンベディングとモデルエンベディングを結合した入力を構成する：

$$\mathbf{Z}_{k,\mathcal{T}} = \left[\mathbf{v}_k^{(\mathrm{mdl})}; \mathbf{V}_{\mathcal{T}}^{(\mathrm{tsk})}\right] \in \mathbb{R}^{(P+1)\times 768} \tag{13}$$

この入力に対し，位置エンコーディングを付加した後，2 層の Transformer Encoder に入力する．Encoder の出力のうち，モデルベクトルに対応する位置のベクトルを $\mathbf{u}_{k,\mathcal{T}} \in \mathbb{R}^{768}$ とする．最後に，$\mathbf{u}_{k,\mathcal{T}}$ を MLP に入力し，推定スコア $\hat{s}_k$ を算出する：

$$\hat{s}_k = \mathrm{MLP}(\mathbf{u}_{k,\mathcal{T}}) \tag{14}$$

d） 最　適　化

予測スコア $\hat{s}_k$ と，実際のファインチューニングによって得られた真の Macro F1 値 s_k との間の平均二乗誤差（MSE）を損失関数として定義する：

$$L = \frac{1}{|\mathcal{M}|}\sum_{k=1}^{K}(\hat{s}_k - s_k)^2 \tag{15}$$

この損失を最小化するように，モデルエンベディング $\{\mathbf{v}_k^{(\mathrm{mdl})}\}$，Transformer Encoder，および MLP の各パラメータを誤差逆伝播法により更新する．

5.4　推論フェーズ

推論フェーズでは，未知のターゲットタスクに対し，以下の手順でモデルのランキングを生成する．

1. **タスクエンベディングの抽出**: 新たなタスク $\mathcal{T}$ の訓練データを用い，Prompt Tuning によって P 個の特殊トークンベクトルを取得する．
2. **スコア推定**: 学習済みの各モデルエンベディング $\mathbf{v}_k^{(\mathrm{mdl})}$ とタスクエンベディングを順次結合し，学習済みの Transformer Encoder および MLP を通じて各モデルの予測スコア $\hat{s}_k$ を算出する．
3. **ランキングの生成**: 算出された $\hat{s}_k$ の降順にモデルをソートし，ターゲットタスクに適合する可能性が高い順にモデルリストを提示する．

6　評 価 結 果

本節では，構築したデータセットにおける提案手法および既存の検索手法（KNN，LogME，H-Score，Model Spider，Avg. Rank）の検索性能を評価する．表 3 に評価結果の詳細を示す．

検索重要度（Criticalness）が High なタスク群においては，各手法間で性能の差が顕著に見られた．全体的な傾向として，推論ありの手法の方が，推論なしに比べて高い検索精度を示す傾向にある．一方で，推論ありの手法は検索時の計算コストが高いことから，実用上は検索速度との間にトレードオフが存在する点に留意が必要である．

推論なしの手法群に着目すると，検索重要度が高いタスクにおいて，提案手法が NDCG@1 (0.750) および NDCG@3 (0.719) の双方で最も高い平均スコアを記録した．今後，学習データがさらに増加する環境においては，提案手法を含む推論なし手法のさらなる性能向上が期待できる．

表 3　全タスクにおける各検索手法の定量評価結果（NDCG スコア）．タスク重要度（Criticalness）ごとの平均値および標準偏差を併記する．各項目における最高値を**太字**，次点を下線で示す．

Group	ID	NDCG@1						NDCG@3					
		推論なし			推論あり			推論なし			推論あり		
		提案手法	M.Sp	AvR	KNN	LogME	H-Sc	提案手法	M.Sp	AvR	KNN	LogME	H-Sc
High	3	**1.000**	0.000	0.000	**1.000**	0.500	**1.000**	**0.765**	0.000	0.235	**0.685**	0.656	0.564
	6	**1.000**	0.500	0.500	**1.000**	**1.000**	**1.000**	**0.725**	0.435	0.672	0.712	0.541	**0.779**
	9	**0.750**	0.500	0.500	**1.000**	**1.000**	**1.000**	**0.712**	0.491	0.644	**1.000**	0.899	0.920
	13	0.500	**1.000**	**1.000**	**1.000**	0.750	0.500	0.626	0.865	**0.932**	**0.921**	0.875	0.672
	28	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.852	0.676	**0.926**	**0.883**	**0.883**	**0.883**
	37	**0.750**	**0.750**	**0.750**	**0.750**	**0.750**	**0.750**	**0.824**	0.352	**0.824**	**0.809**	0.528	0.528
	48	0.250	**1.000**	**1.000**	0.000	0.000	**0.750**	0.531	0.499	**0.561**	0.457	0.398	**0.824**
	Mean	**0.750**	0.679	0.679	0.821	0.679	**0.893**	**0.719**	0.474	0.685	**0.781**	**0.781**	0.641
	STD	0.289	0.374	0.374	0.374	0.134	0.374	0.112	0.269	0.244	0.182	0.142	0.189
Medium	10	**1.000**	**1.000**	**1.000**	**0.750**	**0.750**	**0.750**	0.926	0.661	**1.000**	0.809	**0.824**	**0.824**
	14	**0.500**	0.250	0.250	**1.000**	0.750	**1.000**	**0.515**	0.500	0.457	**0.867**	0.778	0.633
	17	0.750	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.883	0.735	**0.941**	**0.883**	0.765	0.765
	24	**0.750**	0.500	0.500	**1.000**	0.000	**1.000**	0.691	0.574	**0.707**	**1.000**	0.531	**1.000**
	43	**1.000**	**1.000**	**1.000**	**1.000**	0.750	**1.000**	**1.000**	0.941	**1.000**	**1.000**	0.809	0.867
	47	**1.000**	0.750	0.750	**1.000**	**1.000**	0.750	0.645	0.661	**0.883**	**0.941**	0.587	0.926
	Mean	**0.833**	0.750	0.750	**0.958**	0.667	**0.958**	0.777	0.679	**0.831**	**0.917**	0.860	0.691
	STD	0.204	0.316	0.316	0.102	0.342	0.102	0.188	0.152	0.213	0.077	0.091	0.124
Low	11	**1.000**	**1.000**	**1.000**	**1.000**	0.750	**1.000**	**1.000**	0.926	0.926	**1.000**	0.883	**1.000**
	16	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
	20	**1.000**	**1.000**	**1.000**	0.500	**0.750**	**0.750**	**1.000**	0.867	**1.000**	0.691	**0.735**	**0.735**
	26	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
	29	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.867	**1.000**	**1.000**	**1.000**	**1.000**
	32	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
	34	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
	35	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
	39	0.750	**1.000**	**1.000**	**0.750**	**0.750**	**0.750**	0.883	**1.000**	0.941	**0.883**	**0.883**	**0.883**
	44	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
	45	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.941	**1.000**	**0.941**	**0.941**	**0.941**
	46	0.750	**1.000**	**1.000**	0.500	0.500	**1.000**	0.707	**1.000**	0.941	0.765	0.765	**0.883**
	Mean	0.958	**1.000**	**1.000**	**0.896**	**0.896**	**0.958**	0.966	0.967	**0.984**	0.940	0.934	**0.953**
	STD	0.097	0.000	0.000	0.198	0.097	0.167	0.088	0.053	0.029	0.106	0.083	0.097

7 結　　論

本研究では，事前学習済み BERT モデル検索という問題を体系的に定義し，その評価基盤として 48 件の文書分類タスクと 20 種類のモデルからなるベンチマークデータセットを構築した．本データセットを用い，既存の 5 つの主要な検索手法に加え，タスク表現とモデル埋め込みをを用いた新たなモデル検索手法を提案し，比較評価を行った．

実験の結果，検索の重要度が高い **High Criticalness** なタスクにおいては，推論を伴う手法（LogME および k 近傍法）が高い検索精度を示す一方，推論を必要としない **Inference-free** な手法の中では，提案手法が最も優れた性能を達成することを確認した．これにより，検索時の計算コストを抑えつつ高精度なモデル選定を実現する手法としての有効性が示された．

今後の展望としては，まず検索対象となるモデルのバリエーションをさらに拡大し，より大規模なモデル群への対応を進める．また，クエリとなるタスクの拡張として，LLM を用いたデータ生成手法などを活用し，評価タスクのさらなる拡充を目指す．本研究で構築した評価基盤が，今後のモデル検索研究のさらなる発展に貢献することを期待する．

謝　　辞

本研究は，JSPS 科研費 JP24K03228，JP25K03229，JP25K03228，ならびに，2025 年度国立情報学研究所公募型共同研究（251S4-22794）の助成を受けたものです．ここに記して謝意を表します．

文　献

[1] Nermeen Abou Baker and Uwe Handmann. One size does not fit all in evaluating model selection scores for image classification. *Scientific Reports*, Vol. 14, , 2024.

[2] Yajie Bao, Yang Li, Shao-Lun Huang, Lin Zhang, Lizhong Zheng, Amir Zamir, and Leonidas Guibas. An Information-Theoretic Approach to Transferability in Task Transfer Learning. In *Proceedings of 2019 IEEE International Conference on Image Processing (ICIP 2019)*, pp. 2309–2313, 2019.

[3] Iz Beltagy, Kyle Lo, and Arman Cohan. SciBERT: A Pretrained Language Model for Scientific Text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP 2019)*, pp. 3615–3620, 2019.

[4] Daniel Bolya, Rohit Mittapalli, and Judy Hoffman. Scalable Diverse Model Selection for Accessible Transfer Learning. In *Proceedings of the 35th International Conference on Neural Information Processing Systems (NeurIPS 2021*, pp. 19301–19312, 2021.

[5] Tommaso Caselli, Valerio Basile, Jelena Mitrović, and Michael Granitzer. HateBERT: Retraining BERT for Abusive Language Detection in English. In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, pp. 17–25, 2021.

[6] Tuhin Chakrabarty, Christopher Hidey, and Kathy McKeown. IMHO Fine-Tuning Improves Claim Detection. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2019)*, pp. 558–563, 2019.

[7] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. LEGAL-BERT: The Muppets straight out of Law School. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*, pp. 2898–2904, 2020.

[8] Mengyu Dai, Amir Hossein Raffiee, Aashish Jain, and Joshua Correa. Evaluating Transferability in Retrieval Tasks: An Approach Using MMD and Kernel Methods. In *Proceedings of the 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2024)*, pp. 22390–22400, 2024.

[9] 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 (NAACL 2019)*, pp. 4171–4186, 2019.

[10] Louis Fouquet, Simona Maggio, and Léo Dreyfus-Schmidt. Transferability Metrics for Object Detection. *arXiv preprint arXiv:2306.15306*, 2023.

[11] Xiaochuang Han and Jacob Eisenstein. Unsupervised Domain Adaptation of Contextualized Embeddings for Sequence Labeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP 2019)*, pp. 4238–4248, November 2019.

[12] He, Pengcheng and Liu, Xiaodong and Gao, Jianfeng and Chen, Wei. DeBERTa: Decoding-Enhanced BERT with Disentangled Attention. In *Proceedings of the 9th International Conference on Learning Representations (ICLR 2021)*, pp. 1–21, 2021.

[13] Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. ClinicalBERT: Modeling Clinical Notes and Predicting Hospital Readmission. *arXiv preprint arXiv:1904.05342*, 2019.

[14] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. *arXiv preprint arXiv:1909.11942*, 2019.

[15] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan So, and Jaewoo Kang. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, Vol. 36, pp. 1234–1240, 2019.

[16] Brian Lester, Rami Al-Rfou, and Noah Constant. The Power of Scale for Parameter-Efficient Prompt Tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP 2021)*, pp. 3045–3059, November 2021.

[17] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly optimized BERT Pretraining Approach. *arXiv:1907.11692*, 2019.

[18] Zhuang Liu, Degen Huang, Kaiyu Huang, Zhuang Li, and Jun Zhao. FinBERT: a pre-trained financial language representation model for financial text mining. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI 2020)*, pp. 4513–4519, 2021.

[19] Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. BERTweet: A pre-trained language model for English tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*, pp. 9–14, 2020.

[20] Huu-Long Pham, Ryota Mibayashi, Takehiro Yamamoto, Makoto P. Kato, Yusuke Yamamoto, Yoshiyuki Shoji, and Hiroaki Ohshima. Inference-based no-learning approach on pre-trained BERT model retrieval. In *Proceddings of the 2024 IEEE International Conference on Big Data and Smart Computing (BigComp 2024)*, pp. 234–241, 2024.

[21] Laila Rasmy, Yang Xiang, Ziqian Xie, Cui Tao, and Degui Zhi. Med-BERT: pre-trained contextualized embeddings on large-scale structured electronic health records for disease prediction. *npj Digital Medicine*, Vol. 4, , 2021.

[22] Parisa Safikhani and David Broneske. AutoML Meets Hugging Face: Domain-Aware Pretrained Model Selection for Text Classification. In *Proceedings of the 2025 Annual Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics (NAACL 2025)*, pp. 466–473, 2025.

[23] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All You Need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS 2017)*, pp. 6000–6010, 2017.

[25] Yuncheng Yang, Meng Wei, Junjun He, Jie Yang, Jin Ye, and Yun Gu. Pick the Best Pre-trained Model: Towards Transferability Estimation for Medical Image Segmentation. In *Proceedings of the 26th Medical Image Computing and Computer Assisted Intervention (MICCAI 202)*, pp. 674–683, 2023.

[26] Kaichao You, Yong Liu, Jianmin Wang, and Mingsheng Long. LogME: Practical Assessment of Pre-trained Models for Transfer Learning. In *Proceedings of the 38th International Conference on Machine Learning (PMLR 2021)*, Vol. 139, pp. 12133–12143, 2021.

[27] Yi-Kai Zhang, Ting-Ji Huang, Yao-Xiang Ding, De-Chuan Zhan, and Han-Jia Ye. Model Spider: Learning to Rank Pre-trained Models Efficiently. In *Proceedings of the 37th International Conference on Neural Information Processing Systems (NIPS 2023)*, pp. 13692–13719, 2023.

[28] Wangchunshu Zhou, Canwen Xu, and Julian McAuley. Ef-

ficiently Tuned Parameters Are Task Embeddings. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP 2022)*, pp. 5007–5014, December 2022.