文書分類タスクにおける事前学習済みBERTモデルの検索

三林　亮太† 　日置　淳也† 　ファムフーロン†† 　加藤　誠††† 　山本　祐輔††††
荘司　慶行††††† 　山本　岳洋† 　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
†† コベルコシステム株式会社　〒 657-0845 兵庫県神戸市灘区岩屋中町 4–2–7
††† 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
†††† 静岡大学 情報学部　〒 432–8011 静岡県浜松市中区城北 3–5–1
††††† 青山学院大学 理工学部　〒 252–5258 神奈川県相模原市中央区淵野辺 5–10–1
E-mail: †{threeforest8,junya.hioki.j,huulongpham28}@gmail.com, ††mpkato@slis.tsukuba.ac.jp,
†††yusuke_yamamoto@acm.org, ††††shoji@it.aoyama.ac.jp, †††††t.yamamoto@sis.u-hyogo.ac.jp,
††††††ohshima@ai.u-hyogo.ac.jp

あらまし　本論文では，文書分類タスクにおける事前学習済み BERT モデルの検索をおこなう．近年，機械学習モデルを用いて特定のタスクを解く際には，事前学習済みモデルをベースにファインチューニングすることが一般的である．このベースとなる事前学習済みモデルは，ユーザがタスクに応じて適切なモデルを選択する必要がある．しかし，事前学習済みモデルは多く存在し，タスクにおいてどれが適切なモデルであるかはわからない．そこで，本研究では解きたいタスクに適した事前学習済みモデルを検索する手法を提案する．本研究は，BERT モデルを対象とし，タスクは文書分類を対象とする．まず，Kaggle 上に存在する文書分類タスク 4 件を対象に，5 つの事前学習済み BERT モデルで実際にファインチューニングをおこなった．ファインチューニングによって，そのタスクに対するモデルの性能を得る．その結果を正解データとしてランキングの推定をおこなう．推定の際には，タスクの入力データを事前学習モデルに入力し，そこから得られた特徴量をもとに，実際の性能順にランキングする方法をおこなう．

キーワード　モデル検索，ランキング学習，事前学習済みモデル，BERT

1　はじめに

本論文では，文書分類タスクにおける事前学習済み BERT モデルの検索手法を提案する．たとえば，図 1 に示すように，医療文書を分類するタスクを解きたい場合，事前学習済み BERT モデルの中から，一番性能が高くなる Med-BERT を検索結果の上位に示す．

文書分類タスクにおいて効果的な手法は，BERT モデルを用いた分類手法である．BERT モデル [5] は Transformer [16] ベースの機械学習モデルであり，分類性能が高いことが知られている．関連研究で詳しく述べるが，BERT モデルは大量の文書データで事前学習されており，その事前学習済みモデルによってタスクの性能は異なる．事前学習済みモデルは多く存在し [1]，ユーザは事前学習済み BERT モデルから解きたい文書分類タスクに適したモデルを選択する必要がある．

しかし，このような大量の事前学習済み BERT モデルから適した BERT モデルを選択するのは容易でない．モデルがタスクに適しているかは，ユーザがモデルの情報を調査する必要がありコストが高く，正確ではない．実際にいくつかの BERT モデルをファインチューニングして確かめる方法もあるが，ファインチューニングには計算資源や計算時間がかかるためコストが高い．

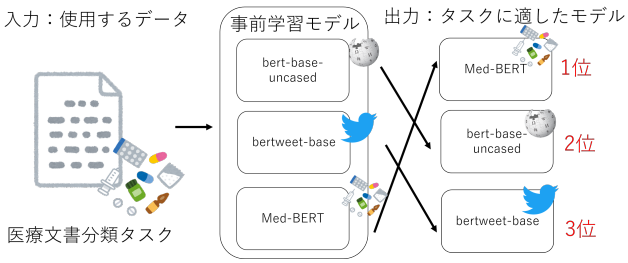

図 1　事前学習済み BERT モデルの検索

そこで，本研究では，解きたい文書分類タスクに適した事前学習済み BERT モデルの検索手法を提案する．本手法では，解きたいタスクのデータを入力に，適した事前学習モデルを検索する手法を提案する．まず，4 件の文書分類タスクを対象に，5 件の事前学習済み BERT モデルをファインチューニングする．ファインチューニング結果を正解のランキングとし，そのランキングを再現するような学習をおこなう．

2　関連研究

文書分類タスクにおける事前学習済み BERT モデルを検索する手法は，筆者らの知る限り提案されていない．これまでに，

1：モデル共有サイトの huggingface では 106,932 件の事前学習済みモデルが公開されている（2022/12/27 現在）

顔画像認識モデルの検索や音声分離のモデル検索など関連した研究は存在するが，文書分類タスクにおいては存在しない．本研究と関連が深い研究として，機械学習モデルの検索手法とBERT 事前学習済みモデルとランキング学習について述べる．

2.1 タスク内における適切な機械学習モデルの検索

タスク内のデータによってモデルを切り替える研究がある．星野ら [18] は顔画像認識において，入力画像に対して適した機械学習モデルを検索し推論をおこなっている．これは顔画像を推論する API をもとに，タスク内のデータがどの API で解くとよいかを推論をしている．

2.2 事前学習済みモデル

事前学習済みモデルとは，大量の学習データであらかじめ学習した状態のモデルである．近年，事前学習済みモデルのパラメータをもとに，解きたいタスクにおけるファインチューニングをすることで，タスクに対する性能が高くなることがわかっている．たとえば，画像分類タスクにおいては CIFAR-10 という大量の画像データセットで学習したモデルをもとにファンチューニングをすると性能が向上することが知られている．

ドメインに応じた事前学習済みモデルも提案されており，これらについては 2.4 節で詳しく述べる．これらのモデルは，Wikipedia などの一般的なデータで学習した事前学習モデルよりも，ドメインに関連するタスクにおいて性能が高くなることが報告されている．

2.3 BERT モデル

BERT [5] は，Transformer [16] の Encoder 部分に特化した汎用言語モデルである．一般的に，BERT は大量のテキストで事前学習と呼ばれる学習をおこなう．この事前学習によって，特定のタスクでの性能が違うことが知られている．事前学習済みモデルとは，Wikipedia などの大量の文書データであらかじめ学習したモデルのことである．

BERT の改良モデルとして RoBERTa [11] がある．RoBERTa は事前学習で Next Sentence Prediction をおこなわず，事前学習に用いるテキストデータを増量させたモデルである．

2.4 追加学習済み BERT モデル

近年，特定のドメインで追加の学習をおこなった，追加学習モデルが提案されている．ファインチューニングをおこなう前に，解きたいタスクのドメインで追加の学習をすることでモデルの性能が向上することが知られている [6], [13]．たとえば，Med-BERT [14] では，医療に関するコーパスで追加学習をおこない，疾患の予測タスクについてファインチューニングした結果，事前学習モデルより，予測性能が向上していることを示した．また，LEGAL-BERT [4] においても，法律に関するコーパスで追加学習をおこなうことで，タスクの性能が向上するほか，学習の収束速度の向上やロス値が小さくなる傾向を示した．Han らは，固有表現抽出タスクにおいて，追加学習で良い性能を示している [7]．以上のように，特定のドメインで追加学習をおこなうことで，タスクに対するモデルの性能が向上することが知られている．

追加学習済み BERT モデルはいくつか提案されており，その追加学習のドメインは多岐に渡る．たとえば，beltagy らは化学に関するドメインに特化した追加学習モデルである，SciBERT [1] を提案している．また，LEGAL-BERT [4] では，法律関係のコーパスで追加学習をおこなっており，多様なドメインに特化している．

特に，医療に関する追加学習モデルは多く提案されており，Lee らは医学論文を用いて追加学習をおこなった BioBERT [10] を提案している．また，Laila らは，Med-BERT [14] と呼ばれる，医療に関するドメインに特化した追加学習モデルを提案している．加えて，ClinicalBERT [8] では，匿名の医療診断書情報を用いて追加学習をおこなっている．さらに，Yao ら [17] は生物医学とコンピュータ科学を対象に追加学習をしたモデルを提案している．

近年では，SNS 等のインターネットに関するドメインに特化した追加学習モデルも提案されている．Tuhin らは，インターネット上の意見に関するドメインに特化した追加学習モデルである IMHO [3] を提案している．特定の SNS サービスでは，Twitter のコーパスを用いて追加学習した BERTweet [12] を Dat らは提案している．本モデルは RoBERTa [11] を用いて学習している．以上のように，多用なドメインのコーパスにおいて，追加学習がおこなわれている．

3 汎用言語モデル検索問題の定義

本節では，汎用言語モデル検索問題の必要性について述べ，汎用言語モデル検索問題の問題定義について述べる．

3.1 汎用言語モデル検索の必要性

近年，BERT などの汎用言語モデルは事前学習済みのモデルに対してファインチューニングをおこなうことが一般的である．事前学習済みの汎用言語モデルをベースにすることで大規模な訓練データを作らずに済むなど多くの利点がある．

ただし，事前学習済みモデルは適切に選ぶ必要がある．たとえば，huggingface 上では多くの事前学習済みモデルが公開されているが，解きたいタスクにおいて，どれが良い事前学習済みモデルであるかはわからない．公開されている事前学習済みモデルには，名前だけでは性能を判断できないものも多く，事前学習で用いたコーパスなどの情報が記載されていないモデルもある．そこで，解きたいタスクに応じて，適切な汎用言語モデルを自動で検索してくれる手法が必要である．

適切であろうモデルをいくつかユーザが選択し，実際にファインチューニングすれば性能はわかるが，ファインチューニングは大きいメモリの GPU が必要かつ計算にかかる時間が長いため計算コストが高い．そのため，ファインチューニングより速く，事前学習済みモデルの性能が検索できれば，ユーザにとって無駄な時間を使わずに済む利点がある．以上のように，汎用言語モデル検索問題には実際に性能を確かめるまでの計算コストがかかるという問題点を解決する必要がある．

3.2 問題定義

本タスクは以下のように定義する．

- **入力：テキストとラベルのペア集合**
- **出力：事前学習済み汎用言語モデル列**

入力はテキストとラベルのペアの集合：$D = \{(x_i, y_i)\}_{i=1}^{N}$．ここで，$x_i$ はテキストであり，y_i はラベルである．すべてのテキストとラベルのペアの集合を $\mathcal{D} = D_1, D_2, \cdots, D_n$ としたとき，ある $D \in \mathcal{D}$ に対する出力は，あるスコア関数 $f: \quad \mathcal{D} \times M \to R$ に対して，$i < j \Rightarrow f(D, m_i) \leqq f(D, m_j)$ であって $\forall m(f(D, m_1) \leqq f(D, m))$ を満たすようなモデル列 $\mathbf{m} = (m_1, m_2, \cdots, m_k)$ である．

3.3 汎用言語モデル検索のクラス

本節では，汎用言語モデル検索問題のクラスを定義する．本問題の解決方法はいくつか考えられ，本研究ではそれらのアプローチを大きく 3 クラスに分ける．

まず，汎用言語モデルに対して，多少のファインチューニングをしていい場合である．一般的に，タスクに応じたファインチューニングをおこなう場合，検証データにおける Loss が最小となるエポックまで学習を続ける．しかしこれでは計算時間が長くかかってしまう．本アプローチでは最小の Loss とはいかないが，性能が推定できる段階までに学習を留めることで，計算時間を短くモデルの性能を推定する．本研究ではこの方法をファインチューニングあり手法と呼ぶ．たとえば，正例集合 P と負例集合 N を入力として与える場合，$score(BERT_i) = eval(finetuning(BERT_i, (P, N)))$ となる．

次に，汎用言語モデルから得た事例をもとにモデルの順位を推定する方法である．本研究ではこの方法を，Infer あり手法と呼ぶ．たとえば，文書集合 D が入力として与えられる場合，各 BERT から得られた事例が十分に分離されていれば適切なモデルであると判断できる．つまり，$score(BERT_i) = \Sigma_{d,d' \ in D} ||BERT_i(d) - BERT_i(d')||$ である．正例 p と負例 n を入力として与えられる場合は，各 BERT から得られた事例が，十分に遠いモデルが良いモデルである．つまり，$score(BERT_i) = ||BERT_i(p) - BERT_i(n)||$ などの方法が考えられる．以上のように，汎用言語モデルに文書を入力して得られた事例を用いて推論することを Infer あり手法と呼ぶ．

最後に，汎用言語モデルに対しては推論時に何も入力せずモデルの順位を推定する方法である．本研究ではこの方法を，Infer なし手法と呼ぶ．これは事前に特徴量を計算しておき，計算時には汎用言語モデルで infer しない．基本的には，正例 p と負例 n を入力として与えられる場合，

たとえば，代表文書集合 R を用意して，すべての文書間の類似度 $||BERT_i(r) - BERT_i(r')||(r, r' \in R)$ を事前に計算しておくなどの方法が考えられる．

4 文書分類タスクに用いた学習データと BERT モデル

本研究では，infer あり手法においてタスクを実際に解く．今回は 4 件の文書分類タスクを対象とした．文書分類タスクは，データ分析コンペティションのサイトである Kaggle [2]から 4 件選択した．本タスクの対象とする言語は英語である．

4.1 文書分類タスク

対象とする文書分類タスクは以下の 4 件である．

- **Jigsaw Multilingual Toxic Comment Classification** [3]
- **Natural Language Processing with Disaster Tweets** [4]
- **Quora Insincere Questions Classification** [5]
- **What's Cooking?** [6]

Kaggle から取得できるデータは訓練データとテストの 2 つに分かれている．しかし，テストデータはコンペ用に作成されたものであるため，正解ラベルが付与されていない．そのため，学習データを 8:1:1 に分割し，訓練データ，検証データ，テストデータとして使用した．4 件のタスクの詳細を述べる．

Jigsaw Multilingual Toxic Comment Classification は，会話内容のテキストから失礼なコメントや無礼なコメントを分類するタスクである．学習データは 223,549 件あり，訓練データは 178,839 件，検証データは 22,355 件，テストデータは 22,355 件に分割し使用した．

Natural Language Processing with Disaster Tweets は，Twitter のツイート内容が災害に関するものであるか，そうでないかを分類するタスクである．ラベルは災害に関係があるか，ないかの 2 種類である．学習データは 7,613 件あり，これを訓練データ 6,090 件，検証データ 761 件，テストデータ 762 件に分割し使用した．

Quora Insincere Questions Classification は，質問応答サービスである Quora において，不誠実な質問を分類するタスクである．不誠実な質問とは，役に立つ答えを探す質問ではなく，主張することを目的とした質問や，誤った前提に基づいた質問を指す．ラベルは不誠実な質問であるか，ないかの 2 種類である．学習データは 1,306,122 件あり，これを訓練データ 1,044,897 件，検証データ 130,612 件，テストデータ 130,613 件に分割し使用した．

What's Cooking?は，食材の名称からどの地域の郷土料理の材料かを推定するタスクである．たとえば,「water, vegetable oil, wheat, salt」といった食材の名称から,「indian」の郷土料理であると予測する．特徴量は「baking powder」,「eggs」といった食材名が複数個与えられる．推定する郷土料理の種類は 20 種類ある．学習データは 39,774 件あり，これを訓練データ 31,819 件，検証データ 3,977 件，テストデータは 3,978 件に分割し使用した．

2：https://www.kaggle.com

3：https://www.kaggle.com/competitions/jigsaw-multilingual-toxic-comment-classification

4：https://www.kaggle.com/competitions/nlp-getting-started

5：https://www.kaggle.com/competitions/quora-insincere-questions-classification

6：https://www.kaggle.com/competitions/whats-cooking-kernels-only

表 1　各タスクのデータ数

タスク名	全体のデータ	訓練データ	検証データ	テストデータ
Jigsaw Multilingual Toxic Comment Classification	223,549	178,839	22,355	22,355
Natural Language Processing with Disaster Tweets	7,613	6,090	761	762
Quora Insincere Questions Classification	1,306,122	1,044,897	130,612	130,613
What's Cooking?	39,774	31,819	3,977	3,978

表 2　各タスクの正解率

タスク名	bert-base	bert-large	bert-multilingual	TweetBERT	HateBERT
Jigsaw Multilingual Toxic Comment Classification	0.9599	0.9595	0.9586	0.9295	0.9596
Natural Language Processing with Disaster Tweets	0.8281	0.8202	0.8425	0.7165	0.8412
Quora Insincere Questions Classification	0.9644	0.9653	0.9633	0.9561	0.9646
What's Cooking?	0.7562	0.7720	0.7564	0.7391	0.7416

表 3　各タスクにおけるモデルのランキング

タスク名	1 位	2 位	3 位	4 位	5 位
Jigsaw Multilingual Toxic Comment Classification	bert-base	HateBERT	bert-large	bert-multilingual	TweetBERT
Natural Language Processing with Disaster Tweets	bert-multilingual	HateBERT	bert-base	bert-large	TweetBERT
Quora Insincere Questions Classification	bert-large	HateBERT	bert-base	bert-multilingual	TweetBERT
What's Cooking?	bert-large	bert-multilingual	bert-base	HateBERT	TweetBERT

4.2　検索対象の事前学習済み BERT モデル

検索対象とする事前学習モデルは，huggingface [7]から 5 件のモデルを対象とした．対象のモデルは以下の通りである．

- **bert-base-uncased** [8]
- **bert-large-uncased** [9]
- **bert-base-multilingual-uncased** [10]
- **TweetBERT** [12] [11]
- **HateBERT** [2] [12]

5　事前学習済み BERT モデルの検索

5.1　ランキング評価データセットの作成

まず，本研究ではランキングを評価するためのデータセットを作成する．ランキング評価データセットは，4 件の文書分類タスクにおいて，5 つの事前学習済み BERT モデルで実際にファインチューニングした結果の性能である．本研究における性能は，タスクにおける正解率とし，正解率が高い順に事前学習 BERT モデルをランキングする．

5.2　事前学習済み BERT モデルの検索

提案手法では，検索対象の事前学習済み BERT モデルから得られた特徴量を入力とし，図 2 に示すように，性能のランキングを予測する．今回，ランキングに用いる特徴量として，Masked Language Model Scoring（MLMS）[15] を用いて特徴量を得る．MLMS は単方向モデルにおける Perplexity を擬似的に求める手法である．事前学習済み BERT モデルにおいて，特定の文書分類タスクのテキストデータの Perplexity が低い場合，その事前学習済み BERT モデルはテキストデータをよく理解しているため性能が高くなる可能性が高いと仮説を立てた．Perplexity は一般的に生成モデルでしか計算ができないが，salazar ら [15] が提案した手法で，BERT モデルから Perplexity を擬似的に計算できる．

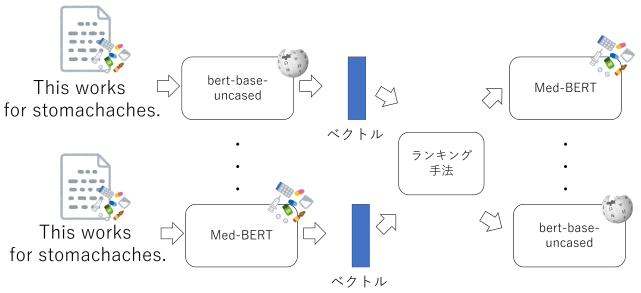

図 2　事前学習済み BERT モデルの検索手法

6　実　　験

6.1　ランキング評価データセットの作成

ランキング評価データセットを作成する際の訓練データの前処理は以下の通りである．

Jigsaw Multilingual Toxic Comment Classification
モデルに入力するデータは “comment_text” 列にあるコメントのテキストを入力とした．テキストの先頭に `[CLS]` トークンを付与し，末尾には `[SEP]` トークンを付与した．正解ラベルは “toxic” 列を正解ラベルとした．

Natural Language Processing with Disaster Tweets
モデルに入力するデータは “text” 列にあるツイートのテキス

7：https://huggingface.co/models
8：https://huggingface.co/bert-base-uncased
9：https://huggingface.co/bert-large-uncased
10：https://huggingface.co/bert-base-multilingual-uncased
11：https://huggingface.co/vinai/bertweet-base
12：https://huggingface.co/GroNLP/hateBERT

表 4　Masked Language Model Scoring によるスコア

タスク名	bert-base	bert-large	bert-multilingual	HateBERT	TweetBERT
Jigsaw Multilingual Toxic Comment Classification	-2.716	-2.561	-2.827	-3.074	N/A
Natural Language Processing with Disaster Tweets	-4.183	-3.983	-4.044	-4.817	N/A
Quora Insincere Questions Classification	-2.399	-2.251	-2.536	-3.100	N/A
What's Cooking?	-3.421	-3.117	-3.344	-4.223	N/A

表 5　Masked Language Model Scoring によるランキング

タスク名	1 位	2 位	3 位	4 位	5 位
Jigsaw Multilingual Toxic Comment Classification	bert-large	bert-base	bert-multilingual	HateBERT	TweetBERT
Natural Language Processing with Disaster Tweets	bert-large	bert-multilingual	bert-base	HateBERT	TweetBERT
Quora Insincere Questions Classification	bert-large	bert-base	bert-multilingual	HateBERT	TweetBERT
What's Cooking?	bert-large	bert-multilingual	bert-base	HateBERT	TweetBERT

トを入力とした．テキストの先頭に [CLS] トークンを付与し，末尾には [SEP] トークンを付与した．正解ラベルは “target” 列を正解ラベルとした．

Quora Insincere Questions Classification モデルに入力するデータは “question_text” 列にあるツイートのテキストを入力とした．テキストの先頭に [CLS] トークンを付与し，末尾には [SEP] トークンを付与した．正解ラベルは “target” 列を正解ラベルとした．

What's Cooking? モデルに入力するデータは “ingredients” 列にある材料名のテキストを入力とした．材料名は複数個あるため，それぞれの材料名の間を半角スペースで結合して一つのテキストとした．テキストの先頭に [CLS] トークンを付与し，末尾には [SEP] トークンを付与した．正解ラベルは “cuisine” 列を正解ラベルとした．

4 件の文書分類タスクに対する，ファインチューニングのパラメータは以下のように設定した．

- **batch size：64**
- **optimizer：Adam** [9]
- **ロス関数：Cross-entropy Loss**
- **学習率：2e-5**
- **Dropout：0.1**
- **max length： 512**

4 件の文書分類タスクに対する学習結果の正解率とモデルのランキングを表 2 と表 3 に示す．4 件の文書分類タスクの結果として，タスクごとにモデルの性能差が見られた．また，図 3 に示すように，各モデル間で，学習にかかるエポック数にも差が見られた．本ランキングを元に，ランキング評価用データを作成した．

6.2　事前学習済み BERT モデルの検索

事前学習済み BERT モデルの検索は，Masked Language Model Scoring（MLMS）[15] を使用した特徴量を用いる．MLMS は BERT モデルにおいて入力文書に対する pseudo-log-likelihood scores（PPLs）を求める手法である．PPLs スコアが低いとその文書を予測する確率が高く，そのモデルにおいて，よく知った文書であることがわかる．本手法では，事前学習済みモデルにおいてスコアが低い場合，そのモデルはその文書をよく知っておりファインチューニングの際に性能が良くなるという仮定のもと MLMS を使用した．

スコアを計算する際は，[CLS] トークンと [SEP] トークンを除外して計算し，1 文のスコアの合計を文のトークン数で割った．今回，スコアを求めるにあたって，計算時間の問題から，Jigsaw Multilingual Toxic Comment Classification タスクと Quora Insincere Questions Classification タスクについては訓練データからランダムに 500 件サンプリングし，スコアを求めた．Quora Insincere Questions Classification タスクと What's Cooking?タスクについては，訓練データ全件に対してスコアを求めた．スコアは合計をデータ数で割り平均した．

結果として，MLMS によるスコアとランキングを表 4 と表 5 に示す．What's Cooking?タスクにおいては，実際にファインチューニングしたランキングと予測したランキングは一致した．Natural Language Processing with Disaster Tweets タスクにおいては，ランキングが一致せず，正解データではランキング一位であった bert-multilingual より bert-large が良い結果となっている．Jigsaw Multilingual Toxic Comment Classification タスクにおいても，ランキングが一致せず，特に HateBERT の順位が予想より低くなった．Quora Insincere Questions Classification タスクにおいても HateBERT の順位が低い傾向にあった．

7　まとめと今後の課題

本研究では，文書分類タスクにおける事前学習済み BERT モデルの検索手法を提案した．まず，汎用言語モデル検索問題の問題定義をおこなった．次に，infer あり手法について取り組み，提案手法を評価するためのランキング評価用データセットを作成した．今回，5 件の文書分類タスクを対象に，4 件の事前学習済み BERT モデルのランキング評価用データを作成した．結果として，各タスクにおいてモデルごとに差が見られた．得られた評価用データをもとに Masked Language Model Scoring を用いたランキングをおこなった．結果として，実際にファインチューニングした際のランキングと予測したランキングはいくつかのタスクにおいて不一致であったが，Infer ありの方法としての一例を示せた．今後の課題としては，文書分類

以外のタスクにおいても，本手法を試し，指定のタスクとデータを入力に，適切なモデルの検索を試したいと考えている．

謝　　辞

本研究は JSPS 科研費 JP21H03775，JP21H03774，JP21H03554，JP18H03244，JP22H03905，2022 年度国立情報学研究所公募型共同研究 21S1002 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] Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*, 2019.

[2] Tommaso Caselli, Valerio Basile, Jelena Mitrović, and Michael Granitzer. HateBERT: Retraining BERT for abusive language detection in English. In *Proceedings of WOAH '21*, pp. 17–25, 2021.

[3] Tuhin Chakrabarty, Christopher Hidey, and Kathy McKeown. IMHO fine-tuning improves claim detection. In *Proceedings of NAACL'19*, pp. 558–563, 2019.

[4] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. Legal-bert: The muppets straight out of law school. *arXiv preprint arXiv:2010.02559*, 2020.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL'19*, pp. 4171–4186, 2019.

[6] Suchin Gururangan, Ana Marasovic, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*, 2020.

[7] Xiaochuang Han and Jacob Eisenstein. Unsupervised domain adaptation of contextualized embeddings: A case study in early modern english. *arXiv preprint arXiv:1904.02817*, 2019.

[8] Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. Clinicalbert: Modeling clinical notes and predicting hospital readmission. *arXiv preprint arXiv:1904.05342*, 2019.

[9] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of ICLR'15*, 2015.

[10] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *arXiv preprint arXiv:1901.08746*, 2019.

[11] 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 preprint arXiv:1907.11692*, 2019.

[12] Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. BERTweet: A pre-trained language model for English tweets. In *Proceedings of EMNLP'20*, pp. 9–14, 2020.

[13] Alan Ramponi and Barbara Plank. Neural unsupervised domain adaptation in NLP - A survey. *arXiv preprint arXiv:2006.00632*, 2020.

[14] 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. *arXiv preprint arXiv:2005.12833*, 2020.

[15] Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. Masked language model scoring. In *Proceedings of ACL'20*, pp. 2699–2712, 2020.

[16] 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 NIPS'17*, pp. 5998–6008, 2017.

[17] Yunzhi Yao, Shaohan Huang, Wenhui Wang, Li Dong, and Furu Wei. Adapt-and-distill: Developing small, fast and effective pretrained language models for domains. In *Proceedings of ACL-IJCNLP'21*, pp. 460–470, 2021.

[18] 星野厚, 齊藤拓己, 岡瑞起. クラウド AI サービス推薦システムの提案. 人工知能学会論文誌, Vol. 37, No. 2, pp. 1–8, 2022.

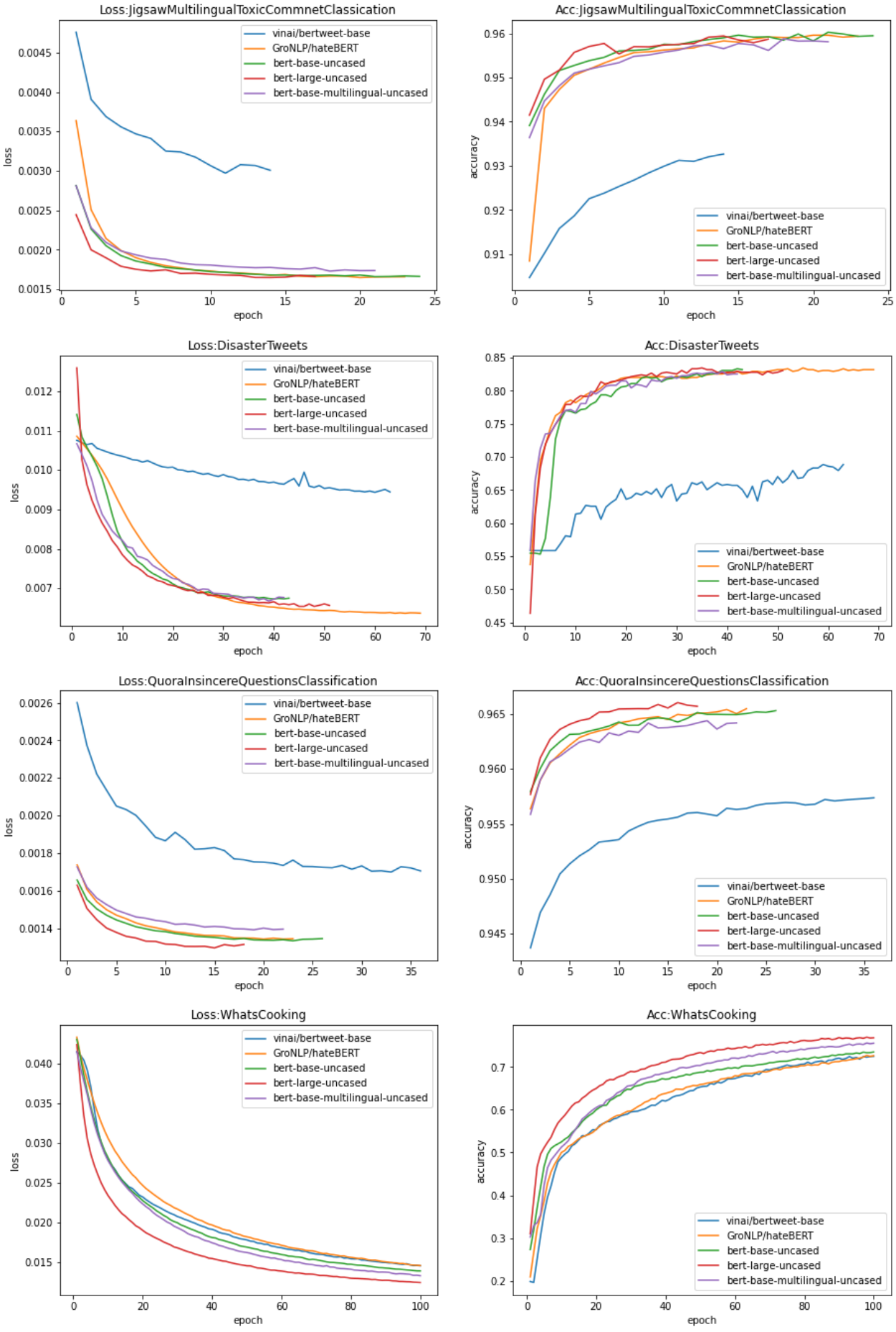

図 3　文書分類タスク 4 件に対する学習曲線