個人情報を活用する小規模言語モデルによる実装の検討

川村　碧葵† 丸　千尋†,†† 中野美由紀†††,†††† 小口　正人†

† お茶の水女子大学　〒 112–8610 東京都文京区大塚 2–1–1
†† 中央大学　〒 112–8551 東京都文京区春日 1–13–27
††† 津田塾大学　〒 187–8577 東京都小平市津田町 2–1–1
†††† 情報・システム研究機構　〒 105–0001 東京都港区虎ノ門 4–13–13 ヒューリック神谷町ビル 2 階
E-mail: †{tamaki,maru.chihiro,oguchi}@ogl.is.ocha.ac.jp, ††g1120536@is.ocha.ac.jp, †††miyuki@tsuda.ac.jp

あらまし　近年，生成 AI，特に大規模言語モデル（LLM）を用いた情報検索，要約，機械翻訳などの技術が大きく進展している。また，IT 技術の進展により小型のセンサーデバイスが様々な場所で利用されており，スマートフォンの GPS 記録からヘルスケアに関する体温，脈拍，あるいは，医療デバイスによる心電図情報の収集など，個人情報を含むデータ利用が急速に進んでいる。現在の LLM の利用ではクラウド上でサービスが提供されることが一般的であるが，上記のような個人情報を含むデータの扱いについては，漏洩リスクのある外部サーバへ送信せずに端末内で安全な処理をすることが求められている。つまり，エッジデバイスの性能が向上しつつある状況において，デバイス上のデータを用いた分野特化型の小型生成 AI モデル（SLM:Small Language Model) が期待されている。つまり，エッジデバイス上の計算リソースには限界があり，大規模言語モデルを動作させることは難しいため，用途に適応した SLM の開発が求められている。本稿では，デバイスで動作可能な小規模言語モデルの構築を目指し，まずは，分野特化のための手法として，言語モデルの中に特定分野の情報も学習するファインチューニングと，特定分野の情報をプロンプティング時にその一部をモデルに渡すことで出力精度を向上させる RAG の二つについて検討を行う。まずは，小さな特殊データに対する精度向上について実験を行い，エッジデバイスで特定のデータを活用する手法について考察を行った。

キーワード　SLM，RAG，ファインチューニング

1 はじめに

近年，エッジデバイスが広く普及し，個人利用が進んでいる。例えばヘルス・スポーツ分野等で個人の体調や記録データが日常的に記録・蓄積されている。これらの個人情報が含まれるデータは漏洩のリスクがある外部サーバに送信するのではなく，端末内で処理することが求められている。情報漏洩対策やデータ主権の観点から，海外のストレージサービスを利用せずに国内でデータ保存をする動きが進んでいる。

また，IT 技術の発展によりエッジデバイスの性能が向上し，AI モデルの利用が可能となっており，大規模言語モデルを利用することが期待されている。つまり，エッジデバイス上の計算リソースには限界があり，大規模言語モデルを動作させることは難しいため，用途に適応した SLM の開発が求められている。本稿では，デバイスで動作可能な小規模言語モデルの構築を目指し，まずは，分野特化のための手法として，言語モデルの中に特定分野の情報も学習するファインチューニングと，特定分野の情報をプロンプティング時にその一部をモデルに渡すことで出力精度を向上させる RAG の二つについて検討を行う。まずは，小さな特定データに対する精度向上について実験を行い，エッジデバイスで特定のデータを活用する手法について考察を行った。

2 関連研究

2.1 機密情報を扱う大規模言語モデルの課題と関連研究

近年，大規模言語モデル（LLM）は文書要約，質問応答，文章生成など多様なタスクにおいて高い性能を示している。しかしながら，ChatGPT などのサービス運用においては，膨大なデータを大規模計算機資源を使って学習を行い，モデルの運用においても高性能な計算機環境が要求される。このような汎用的な LLM サービスに対し，特定の分野，ある限定された機能に特化した小規模言語モデルを構築することで，学習コストも運用コストも低減することが可能でありながら，フロントエンド側で動作することでリアルタイム性に優れ，特定の用途における回答精度は十分に担保できるという提案がなされている [3]，[2]。小規模言語モデルとして，Phi-3 （Microsoft），OpenELM（Apple），Llama3（Meta），Gemma2（Google）等が提供されているが，いずれもエッジデバイス上で動作するには大きなモデルと考えられる。

また，ChatGPT 等の商用クラウド型 LLM を利用する場合，入力データを外部サーバへ送信する必要があり，機密情報や個人情報を含む組織内文書の活用には情報漏洩のリスクが伴う。このため，機密文書を安全に扱うためのローカル環境で動作する LLM の構築手法が重要な課題となっている。柳原ら [1] は，組織内機密文書の利用を前提として，Local な利用を前提とし

た大規模言語モデルの構築を目指している。汎用の大規模言語モデルに機密文書の学習を行う手法として，ファインチューニング（LoRA）および RAG に用い，その精度の評価を行っている。

3 特定データの学習手法の検討

大規模言語モデルにおける特定データの学習手法として，ファインチューニングが用いられる。また，個人情報を含むデータがセンサーデバイス上での取得となることを考慮すると，データは大規模言語モデルで学習するような大規模パラメタにはならないと想定される。今回の実験では，パラメタ数を抑えられる LoRA によるファインチューニングとモデルの外で関連文書群を提供する RAG の二つの方法を検討する。以下，二つの手法の特徴を述べる。

3.1 ファインチューニング

ファインチューニングは，大規模言語モデルに特定分野の情報をさらに学習させることによって，モデルを構築する。従って，モデルを学習させるための環境が必要となり，デバイス上などで動作させる場合には外部からのモデル更新等が必要になると考えられる。一方で，特定分野の専門知識をモデル内に取り込むことでより専門的な問合せへの精度向上が期待できる。

3.2 RAG による文章知識の活用

RAG（Retrieval-Augmented Generation）は，外部の知識源から関連文書を検索し，その情報を基に LLM が応答を生成する手法である。LLM 単体では対応が難しい最新情報や非公開文書に基づく質問応答を可能とし，LLM が事実に基づかない情報を生成するハルシネーションの抑制にも効果がある。ローカル環境において RAG を構築することで，機密文書を外部に送信することなく，安全に情報検索・質問応答を実現できる。一方で，チャンク分割方法や検索精度に応答品質が大きく依存するため，ベクトルデータベース設計や検索戦略の最適化が重要な課題となる。

4 ローカルデータに対するファインチューニングと RAG の評価実験

4.1 使用したデータ

鈴鹿工業高等専門学校令和 6 年度の 5 学科分 pdf 版シラバスの授業内容が記述されている 7 ページ以降を使用し，合計 614 科目の情報を用いた。シラバスの情報と担当教員名の CSV ファイルに整形した。

データ項目を絞っていないシラバスデータの項目は,「学校名開校年度，授業科目，科目番号，科目区分，授業形態，単位の種別と単位数，開設学科，対象学年，開設期，週時間数，教科書/教材，担当教員，到達目標，ルーブリック，概要，授業の数目方・方法，注意点，授業の属性・履修上の区分，授業計画，モデルコアカリキュラムの学習内容と到達目標，評価割合」である。

データ項目を絞ったシラバスデータの項目は,「授業科目，開設学科，教科書，担当教員，到達目標」である。

Accuracy(正解率) を求める為に,「OO の担当教員名は誰ですか？」という問い合わせと担当教員名のファイルを作成し，正解率を求めた。

4.2 実 験 環 境

CPU が 20 コアの Intel Xeon，GPU が NVIDIA の GeForce GTX 1080Ti である，HPC5000-xsl サーバ上で実験を行った。実験では主に CPU を利用した。

4.3 ファインチューニングの評価実験環境

4.3.1 使用したツール

a） 事前学習言語モデル

文書分類モデルの基盤として，日本語事前学習モデルである cl-tohoku/bert-base-japanese-v3 を使用した。モデルの実装および学習には HuggingFace Transformers および PyTorch を用いた。

b） 生成モデル

分類結果を基に自然言語出力を行うため，Ollama を用いてローカル環境で日本語 LLM， lucas2024/gemma-2-2b-jpn-it:q8_0 を実行した。

c） データセット構築

分類チューニングでは入力をシラバス本文とし，出力を担当教員名とする教師あり分類タスクとして定式化した。教員名は LabelEncoder により整数ラベルへ変換し，分類問題としてファインチューニングを行った。

インストラクションチューニングでは，各シラバス本文の先頭に「次のシラバスから担当教員名を答えてください。」という自然言語の指示文を付与し，指示付き入力としてモデルに与えた。教員名は LabelEncoder により整数ラベルへ変換し，分類問題としてファインチューニングを行った。

訓練・検証・テストデータを統合した全てのデータを用いて学習を行った。

4.3.2 モデル構成

本研究では，BERT の `CLS` トークン表現を用いた多クラス分類モデルを構築した。出力層には教員数に対応する線形層を配置し，クロスエントロピー損失を用いて学習を行った。

a） 学 習 設 定

モデルの学習には AdamW オプティマイザを使用した。バッチサイズは 8，エポック数は 3 とした。再現性確保のため，乱数シードを固定して学習を行った。

b） 推論及び生成

推論時には，学習済み BERT 分類モデルを用いて，入力されたシラバス文書から担当教員を推定する。得られた出力を基に，Ollama 上の日本語 LLM へ指示文を入力し，担当教員名のみを生成させた。

生成時には，出力形式を人名のみに限定するプロンプトを設計し，不要な説明文の生成を抑制した。

4.3.3 ファインチューニング手法

a） 分類チューニング

入力データをあらかじめ定義されたクラスラベルに割り当てることを目的としたファインチューニング手法。訓練中に遭遇したモデルの予測に限定されるため，データを事前に定義されたクラスを正確に分類しなければならないプロジェクトに適している。

b） インストラクションチューニング

特定の指示を使った一連のタスクで言語モデルを訓練するファインチューニング手法。自然言語のプロンプトで表されたタスクを理解して実行する能力を向上させる。モデルの柔軟性や対話の品質を向上，ユーザからの複雑な指示に基づいて様々なタスクを処理する必要があるモデルに適している。

4.4 RAG の評価実験環境

4.4.1 使用したツール

a） RAG 基盤

検索器，および生成モデルの統合には LangChain フレームワークを使用した。

b） 埋め込み表現（Embedding）

文書および検索クエリの意味表現を獲得するため，日本語事前学習モデルである `cl-tohoku/bert-base-japanese-v3` を使用した。モデルの推論処理には PyTorch を用い，トークン埋め込みの平均プーリングによって文書ベクトルを生成した。また，類似度計算の安定性を向上させるため，生成されたベクトルに対して L2 正規化を施した。

c） 生成モデル

検索結果を基に，Ollama を用いてローカル環境で日本語 LLM，lucas2024/gemma-2-2b-jpn-it:q8_0 で回答を生成した。検索で取得した文書の内容のみに基づいて質問に対し，担当教員名のみを出力するようプロンプト設計を行った。

4.4.2 システム設計

本研究で構築したシステムは，検索と生成を分離して実装した RAG 構成を採用している。処理の流れを以下に示す。

1. CSV ファイルからシラバス本文を読み込む
2. 各文書を埋め込み表現に変換する
3. 検索器により関連文書を取得する
4. 取得文書をコンテキストとして LLM に入力する
5. 担当教員名を生成し，正解データと比較して評価する

検索手法として，以下の 3 種類を実装し，比較評価を行った。

4.4.3 検 索 手 法

a） セマンティック検索

キーワードの一致だけでなく，ユーザの検索クエリの背後にあるコンテキスト上の意味と意図を理解し，その解釈に基づいた関連性の高い情報を検索する手法。セマンティック検索を実現する手法の一つにベクトル検索があり，文章や単語をベクトルに変換しベクトル間の距離を計算して，意味が近い単語や文章を検索する。

調べたい情報が汎用的な場合は関連情報が妨げになり，求めている情報とは異なる情報となる可能性がある。

b） キーワード検索

データベースの中にある全ての文字を対象としてキーワードや文字列を検索する手法。文字列と一致する内容を探しているだけで，言葉を理解している訳ではない。

c） ハイブリッド検索

複数の検索方法を組み合わせることで，検索結果の精度や関連性を向上させる手法。本稿ではベクトル検索とキーワード検索を用いた。ハイブリッド検索スコアリング (RRF) はベクトル検索とキーワード検索の結果を重みなしで統合して最終ランキングを作成した場合と、重みを与えたうえで統合して最終ランキングを作成した場合を実験した。

4.5 結　　果

データ項目を絞ったシラバスデータの項目は,「授業科目，開設学科，教科書，担当教員，到達目標」である。シラバスデータの csv ファイルサイズが 5,400KB に対して，データ項目を絞った csv のファイルサイズは 383KB である。

検索器が返す上位 k 件の文書をコンテキストとして用いた。

表 1: シラバスデータに対するファインチューニングによる担当教員推定の Accuracy

手法	Accuracy(正解数/総数)
学習をしていない結果	0.0000 (0/124)
分類チューニング シラバスデータ (5,400KB)	0.0000 (0/124)
分類チューニング 絞ったシラバスデータ (383KB)	0.0000 (0/124)
インストラクションチューニング シラバスデータ (5,400KB)	0.0323 (4/124)
インストラクションチューニング 絞ったシラバスデータ (383KB)	0.0565 (7/124)

表 2: RAG によるシラバスデータに対する検索手法毎の担当教員推定の Accuracy と実行時間

k	検索手法	Accuracy(正解数/総数)	実行時間 [s]
3	ベクトル検索	0.0161 (2/124)	298.32
	キーワード検索	0.0242 (3/124)	297.71
	ハイブリッド検索	0.0081 (1/124)	303.91
5	ベクトル検索	0.0242 (3/124)	304.71
	キーワード検索	0.0081 (1/124)	296.57
	ハイブリッド検索	0.0081 (1/124)	305.42
10	ベクトル検索	0.0323 (4/124)	309.46
	キーワード検索	0.0000 (0/124)	300.94
	ハイブリッド検索	0.0000 (0/124)	315.47
15	ベクトル検索	0.0484 (6/124)	317.66
	キーワード検索	0.0000 (0/124)	309.95
	ハイブリッド検索	0.0403 (5/124)	330.08

表 3: データ項目を絞ったシラバスデータに対する検索手法毎の担当教員推定の Accuracy と実行時間

k	検索手法	Accuracy(正解数/総数)	実行時間 [s]
3	ベクトル検索	0.2258 (28/124)	74.80
	キーワード検索	0.0323 (4/124)	37.89
	ハイブリッド検索	0.2177 (27/124)	82.77
5	ベクトル検索	0.2581 (32/124)	94.35
	キーワード検索	0.0403 (5/124)	41.45
	ハイブリッド検索	0.2984 (37/124)	113.90
10	ベクトル検索	0.3145 (39/124)	140.72
	キーワード検索	0.0565 (7/124)	42.04
	ハイブリッド検索	0.3065 (38/124)	194.68
15	ベクトル検索	0.3145 (39/124)	192.42
	キーワード検索	0.1532 (19/124)	49.03
	ハイブリッド検索	0.2016 (25/124)	280.30

表 4: ベクトル検索とキーワード検索の重みを 0.7:0.3 としたハイブリッド検索による担当教員推定の Accuracy と実行時間

k	検索手法	シラバスデータの Accuracy(正解数/総数)	データ項目を絞ったシラバスデータの Accuracy(正解数/総数)
3	ベクトル検索	0.0161 (2/124)	0.2258 (28/124)
	キーワード検索	0.0242 (3/124)	0.0323 (4/124)
	ハイブリッド検索	0.0161 (2/124)	0.2258 (28/124)
5	ベクトル検索	0.0242 (3/124)	0.2581 (32/124)
	キーワード検索	0.0081 (1/124)	0.0403 (5/124)
	ハイブリッド検索	0.0242 (3/124)	0.2581 (32/124)
10	ベクトル検索	0.0323 (4/124)	0.3145 (39/124)
	キーワード検索	0.0000 (0/124)	0.0565 (7/124)
	ハイブリッド検索	0.0323 (4/124)	0.3145 (39/124)
15	ベクトル検索	0.0484 (6/124)	0.3145 (39/124)
	キーワード検索	0.0000 (0/124)	0.1532 (19/124)
	ハイブリッド検索	0.0484 (6/124)	0.3226 (40/124)

シラバスデータに対し，ファインチューニングを行った際の手法による担当教員推定の正解率の違いを表 1 に示す。シラバスデータに対し、RAG による検索手法と検索器が返す文章の上位 k 件を採用した場合の担当教員推定の正解率と実行時間の違いを表 2 に示す。データ項目を絞ったシラバスデータに対し、RAG による検索手法と検索器が返す文章の上位 k 件を採用した場合の担当教員推定の正解率と実行時間の違いを表 3 に示す。ベクトル検索：キーワード検索=0.7:0.3 で重みをつけて、ハイブリッド検索スコアリングを定めた結果を表 4 に示す。

表 1 より，ファインチューニングのみを行った場合，担当教員推定の正解率は最大でも 0.0565 に留まり，全体として低い結果となった。特に分類チューニングでは正解率が 0.0000 となっており，本タスクにおいては単純な分類学習だけでは有効な特徴を獲得できていないことが分かる。一方，インストラクションチューニングではわずかながら精度が向上しており，自然言語形式での学習が一定の効果を持つことが考えれる。

表 2 および表 3 より，RAG を用いた場合の結果を比較すると，シラバス全体のデータでは正解率は最大 0.0484 に留まり，ファインチューニング単体と大きな差は見られなかった。一方でデータ項目を絞ったシラバスデータでは，正解率が大きく向上し，ベクトル検索では最大 0.3145，ハイブリッド検索では最大 0.3065 を達成した。また，実行時間も全体データでは約 300 秒前後であったのに対し，データ項目を絞ったシラバスデータでは最大でも約 280 秒，多くの条件で 100 秒前後となっており，大幅な短縮が確認された。このことから，担当教員推定に関係の薄い情報を削減することで，検索精度と処理効率の双方が向上したと考えられる。

検索件数 k に着目すると，データ項目を絞ったシラバスデータでは，k の増加に伴い正解率が向上する傾向が見られた。特にベクトル検索では，$k = 3$ の 0.2258 から $k = 10$ の 0.3145 まで上昇した。しかし，$k = 10$ から $k = 15$ では正解率の向上が見られず，実行時間のみが増加している。このことから，本実験条件では $k = 10$ 程度が精度と実行時間のバランスが取れた設定であると考えられる。

表 2 および表 3 より，ハイブリッド検索は必ずしもベクトル検索より高い正解率を示すとは限らず，精度が低下する場合が見られた。これは，ハイブリッド検索スコアリング（RRF）において，ベクトル検索とキーワード検索の結果を重みなしで統合しているためであると考えられる。そこで，ベクトル検索の方が多くの条件で高い正解率を示していたことから，ベクトル検索：キーワード検索＝ 0.7：0.3 の重みを設定し，ハイブリッド検索スコアリングを再定義した結果を表 4 に示す。表 4 と表 2，表 3 を比較すると，重み付けを行ったハイブリッド検索では，多くの条件で正解率が向上していることが確認できた。しかし，もともとハイブリッド検索の方が正解率が高かった条件では，重み付けによりわずかに正解率が低下する場合も見られた。

5 まとめと今後の課題

本実験では，シラバスデータを用いた担当教員推定において，ファインチューニング手法と RAG 手法の性能を比較した。その結果，シラバスデータ全体及び，データ項目を絞ったシラバスデータのいずれの場合においても，ファインチューニングより RAG を用いた手法の方が正解率が高いことを確認した。

ファインチューニングでは，分類チューニングにおいて正解率が 0.0000 となり有効な予測を行うことが出来なかった。これは学習時にはシラバス本文を与えたのに対して，推論時には「OO の担当教員名は誰ですか？」という問い合わせを利用したため，入力形式が大きく異なり，モデルが適切に対応出来なかった為であると考える。

一方，インストラクションチューニングでは僅かではあるが正解率の向上を確認できた。これは自然言語の指示形式で学習を行ったことにより，推論時の問い合わせ文との形式差が小さくなり，モデルの汎用性が向上した為であると考えられる。

RAG 手法では，検索によって関連するシラバス情報を直接

参照出来るため，ファインチューニングに比べ高い正解率を得ることが出来たと考えられる。特にデータ項目を絞ったシラバスデータは，そのままのシラバスデータを用いたサイト比較して，性能が大きく向上した。これは不要な情報を除去し，担当教員に関係する情報のみを検索対象としたことで，検索精度が向上したためであると考えられる。

また，検索時に使用する上位文書数 k の増加に伴い，正解率が向上する傾向がみられたが，同時に実行時間も増加することを確認した。このことから，RAG では精度と計算コストのトレードオフが存在すると考えられる。

ハイブリッド検索スコアリングの重みを変化させることで、ハイブリッド検索の正解率が変化することが確認した。データ特性に応じた適切な重みの設定が重要であると考えられる。

以上より，本実験のようにデータ量が限られている情報を活用する際には，モデル内部に知識を学習させるファインチューニングよりも，外部知識を検索して利用する RAG の方が有効であると考えられる。

文　　献

[1] 柳原皓之介，伊藤栄典 “LoRA による機密情報を扱う LLM 実現の試み，“ 研究報告データベースとデータサイエンス. 2025-DBS-181 (10)， pp.1-6，情処， 2025-09-09.

[2] Peter Belcak, Greg Heinrich, et al., "Small Language Models are the Future of Agentic AI, " https://arxiv.org/pdf/2506.02153, 2025.9

[3] Zhenyan Lu, Xiang Li, et al., "SMALL LANGUAGE MODELS:SURVEY, MEASUREMENTS, AND INSIGHTS" https://arxiv.org/pdf/2409.15790, 2025.2

[4] Patrick Lewis, Ethan Perez, Aleksandra Piktus, et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks" https://arxiv.org/pdf/2005.11401, 2020

[5] Hyung Won Chung et al., "Scaling Instruction-Finetuned Language Models" Journal of Machine Learning Research (JMLR), https://arxiv.org/abs/2210.11416, 2024

[6] Hiroki Watanabe, Motonobu Uchikoshi "Generating Privacy-Preserving Personalized Advice with Zero-Knowledge Proofs and LLMs" https://arxiv.org/abs/2502.06425, 2025-4

[7] S.Xu, W.Xie, L.Zhao, and P.He,“Chain of Draft: Thinking Faster by Writing Less,”https://arxiv.org/pdf/2502.18600, 2025.

[8] Noveen Sachdeva, Benjamin Coleman, et al., "How to Train Data-Efficient LLMs" https://arxiv.org/abs/2402.09668, 2024

[9] "Curriculum Reinforcement Learning from Easy to Hard Tasks Improves LLM Reasoning" https://arxiv.org/abs/2506.06632, 2025

[10] Yun Zhu, Jia-Chen Gu, Caitlin Sikora, et al., "ACCELERATING INFERENCE OF RETRIEVAL- AUGMENTED GENERATION VIA SPARSE CONTEXT SELECTION" https://arxiv.org/abs/2405.16178, 2024