アドホック検索タスクにおけるモデルマージの効果検証

佐々木泰河† 山本 岳洋† 大島 裕明† 藤田 澄男††

† 兵庫県立大学 大学院情報科学研究科 〒 651–2197 兵庫県神戸市西区学園西町 8-2-1
†† LINE ヤフー株式会社 〒 102–8282 東京都千代田区紀尾井町 1-3
E-mail: †ad24l026@guh.u-hyogo.ac.jp, ††t.yamamoto@sis.u-hyogo.ac.jp, †††ohshima@ai.u-hyogo.ac.jp,
††††sufujita@lycorp.co.jp

あらまし 本研究では，アドホック検索タスクにおけるモデルマージの有効性について検証する．モデルマージは，複数のモデルの異なる特性を組み合わせて基盤モデルを構築し，計算コストを抑えつつ性能を向上させる手法として注目されている．我々は，モデルマージをアドホック検索タスクに適用することで，短時間かつ低コストで検索モデルの性能を向上させられると考えた．そこで本研究では，検索タスクに対応するようファインチューニングされたモデルと，検索能力は持たないが特定のドメインに特化するように継続事前学習されたモデルを重みレベルで線形にマージし，医療および日本語ドメインにおける文書検索性能への影響を調査した．実験の結果，モデルマージによって検索性能が向上するケースが確認され，特に検索モデルの重みに比重を置いた場合に高い性能が得られる傾向が見られた．この結果は，アドホック検索タスクへのモデルマージの適用可能性を示唆し，低コストかつ高性能なドメイン特化検索モデルの構築に貢献するものと考えられる．

キーワード アドホック検索，大規模言語モデル，モデルマージ

1 はじめに

アドホック検索とは，クエリに対して，文書集合の文書を nDCG [14] などの検索評価指標を最大化するようランキングするタスクである．例えば，「コロナウイルスとは」というクエリが与えられた場合，コロナウイルスの定義に関する文書など，クエリに適合する文書を文書集合から取得し，上位にランキングするタスクである．このタスクは，Web 検索やレビュー検索，論文検索などに利用されている．また，外部情報の検索と組み合わせて，大規模言語モデル (LLM) によるテキスト生成の性能を上げる技術として注目を集めている RAG [22] にも利用されており，非常に重要なタスクである．

近年，このタスクを解くために LLM を活用した多くの検索モデルが提案されている．しかし，一般的な文書でファインチューニングされた検索モデルは，特定のドメインにおける文書検索の性能向上に限界があることが知られている [23]．その理由として，特定ドメインの文書検索は，専門的な知識や特有の表現方法の理解を必要とすることが挙げられる．したがって，ドメインに特化したモデル構築のためには，対応するデータセットを収集し，適切にアノテーションを施した上でファインチューニングを行う必要がある．しかし，これには大規模な計算リソースや時間が必要であるため，新たにドメイン特化の検索モデルを構築することは非常にコストがかかる．

一方，近年複数のモデルをマージすることで，新たな基盤モデルを構築するモデルマージと呼ばれる手法が注目されている．モデルマージは，それぞれのモデルが持つ能力を統合したモデルを構築することが可能である．例えば，数学的推論能力が高いモデルとコード生成能力のあるモデルをマージすると，数学能力が高く，コード生成能力を持つモデルを構築できることが知られている [42]．また，Shah らは独立して訓練されたスタイルと実体の LoRA [11] を効果的にマージすることで，ユーザが提供する任意の実体やスタイルの画像を生成することに成功している [31]．

これまで，モデルマージにより構築されたモデルは，主に文生成や画像生成タスクで利用されてきた．しかし，アドホック検索タスクにおいてモデルマージを適用する研究は見られない．そこで，本研究はアドホック検索タスクにおいてもモデルマージが有効であるか検証する．モデルマージの効果を確認するため，ドメイン特化の文書検索モデルの構築を試みる．マージ元のモデルとして，検索タスクに対応するようファインチューニングされたモデルと，文書検索能力は持たないが特定のドメインに特化するように継続事前学習されたモデルを用意する．それらのモデルをマージすることで，特定のドメインの文書検索性能が高いモデルを構築することができるか検証する．

モデルマージのメリットは，コスト削減に寄与する点である．新たにモデルを一から構築するには，多大なデータ収集コストや計算リソース，時間が必要である．しかし，既存の複数のモデルをマージすることで，そのコストを大幅に抑えることができる．既存モデルの学習済みの重みを活用するため，追加のデータ収集やアノテーション作業，タスクやドメインに適合するためのファインチューニングの負担が軽減される．これにより，短期間かつ低コストでの新たなモデル構築が実現できる．

本研究の，リサーチクエスチョンは以下の 2 つである．

RQ1: モデルマージにより特定ドメインのアドホック検索タスク性能を向上させることは可能か．

RQ2: 性能向上が可能であるとき，どのようなマージ方法が

有効か．

本研究では，2 つのドメインを対象に実験を行った．1 つ目は医療ドメインである．医療分野における文書検索は，臨床現場での迅速かつ正確な情報提供や，医療従事者の意思決定を支援するために非常に重要である．しかし，医療文書は専門性が高いため，通常の検索モデルでは適合性の高い情報を適切に取得できない可能性がある．このため，医療特有の専門用語や知識を理解したモデルを構築することで，検索性能の向上が期待される．そこで，既存の検索タスクに対応するようファインチューニングされたモデルと，検索能力は持たないが医療文書を用いて継続事前学習された医療特化モデルをマージをする．その結果，医療文書検索において，マージモデルがマージ元の単体の検索モデルよりも高い性能を示すか確認する．

2 つ目は，日本語ドメインである．既存の検索モデルの多くは英語を中心としたデータで学習されているため，日本語文書を検索対象にしたとき適合性の高い文書を取得できない可能性がある．日本語特有の言語構造や表現を理解することができる検索モデルを構築することで，検索性能の向上が期待される．そこで，既存の検索タスクに対応するようファインチューニングされたモデルと，検索能力は持たないが日本語文書で継続事前学習された日本語特化モデルをマージする．その結果，日本語文書検索において，マージモデルがマージ元の検索モデルよりも高い性能を示すか確認する．

実験の結果，次のことが分かった．

RQ1: モデルマージにより，マージ元の検索モデルよりも高性能なモデルを構築できる．

RQ2: マージ元の検索モデルの重みに比重を置くマージ設定のとき，マージモデルの検索性能が高くなる傾向にある．

本論文の構成は以下の通りである．2 節では，LLM を活用したアドホック検索やモデルマージに関する既存研究について述べる．3 節では，実験に利用するテストコレクションやモデル，マージ方法など実験の設定について述べる．4 節では，実験の結果を示し，5 節で，その結果に対する考察を述べる．6 節では，まとめと今後の展望について述べる．

2 関 連 研 究

2.1 アドホック検索における LLM の利用

アドホック検索タスクを解くための手法として疎検索と密検索 [44] がある．疎検索は，クエリと文書の関連度を，単語の一致に基づいて計算する検索方法である．疎検索では，通常，クエリや文書は疎なベクトルで表現される．疎なベクトルとは，大きな次元空間に対して，ほとんどの要素がゼロであるようなベクトルを指す．代表的な手法には，TF-IDF [16] や BM25 [30] がある．

密検索は，クエリと文書の関連度を，密なベクトルに基づいて計算する検索手法である．この検索手法では，クエリや文書の意味を表現するベクトルを LLM により生成することが多い．疎検索では見つけにくかった意味的な関連性を考慮した検索が可能な点が特徴である．密検索に利用する LLM には，バイエンコーダモデルとクロスエンコーダモデルがある．バイエンコーダモデルは，クエリと文書を個別にベクトル化するため，効率的な類似度計算が可能である．一方，クロスエンコーダモデルは，クエリと文書を結合してベクトル化するため，バイエンコーダモデルより精度の高い関連度スコアを提供するが，計算コストが高いという課題がある．

アドホック検索タスクに LLM を用いることにより，性能が向上することが報告されている [5]．本研究でも，アドホック検索タスクに LLM を活用し，その中でもバイエンコーダモデルに焦点をあてる．クエリや文書のベクトル化には，BERT [6] が頻繁に使用されている [8]．アドホック検索タスクに BERT を利用するアプローチとして DPR [17] がある．DPR は，クエリおよび文書の `[CLS]` トークンの出力を利用してベクトル化し，これを入力全体の意味的な表現として扱う．この `[CLS]` ベクトルを使用し，クエリと文書の類似度を計算することで，関連性の高い文書を効率的に検索することが可能である．さらに，クエリと文書のトークンごとのペアワイズ類似度を計算する ColBERT [18] や，モデルに疎なベクトルを生成させる SPLADE [7] などのアプローチも提案されている．阿部らは，DPR，ColBERT，SPLADE のいずれのアプローチが日本語ドメインの文書検索において有効であるか検証している [46]．

Contriever [13], OpenAI embedding [27], e5 [37] はアドホック検索タスクのための強力な埋め込みモデルとして有名なモデルである．これらのモデルは，コントラスト学習 [19] を用いてトレーニングされている．コントラスト学習は，似た意味を持つテキストは似た表現，異なる意味を持つテキストは異なる表現に埋め込むことをモデルに学ばせる学習方法である．コントラスト学習によりトレーニングされたモデルは，MS MARCO [3] や BEIR [35] など主要な情報検索テストコレクションにおいて，高い性能を示すことが報告されている．

近年では，Llama2 [36] や Mistral [15] など 70 億パラメータ規模のモデルもテキスト埋め込みの基盤モデルとして利用されている．Ma らは Llama2 をアドホック検索タスクに適応するように LoRA ファインチューニングすることで，RepLlama や RankLlama を構築した [24]．また，Wang らは，Mistral をアドホック検索タスクに適応するようにファインチューニングすることで，`intfloat/e5-mistral-7b-instruct`[1] を構築した [38]．RepLlama，RankLlama，および，`intfloat/e5-mistral-7b-instruct` は，いずれも高い性能を示すことが報告されている．本研究では，アドホック検索タスクに適応するようにファインチューニングされたモデルである `intfloat/e5-mistral-7b-instruct` を，マージ元のモデルとして利用する．

2.2 モデルマージ

モデルマージとは，複数の基盤モデルをもとに，新たなひとつの基盤モデルを構築する処理である．モデルマージは，重みレベルのマージと層レベルのマージに大別される [2]．重みレベル

1：`https://huggingface.co/intfloat/e5-mistral-7b-instruct`

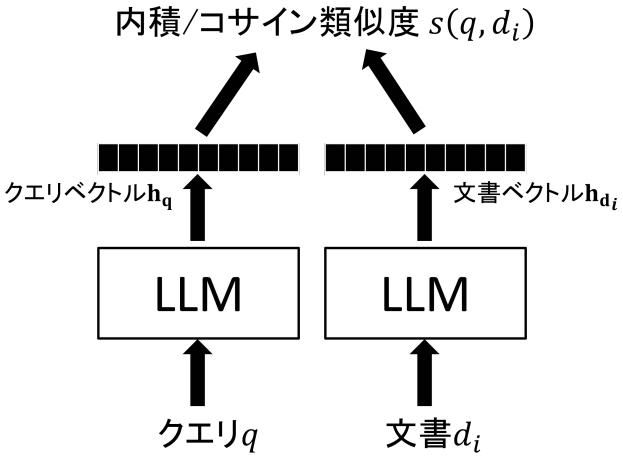

図 1　バイエンコーダモデルの仕組み.

のマージは，マージ元のモデル $M^{(1)}$ のパラメータ $\theta^{(1)}$ と，と $M^{(2)}$ のパラメータ $\theta^{(2)}$ を用いて，新たなマージモデル $M^{(\text{new})}$ のパラメータ $\theta^{(\text{new})}$ を計算する方法である．一方，層レベルのマージは，重みは固定し，マージ元のモデルから，利用する層を選択して，新たなマージモデルを構築する方法である．重みレベルのマージは，同じアーキテクチャ，同じパラメータ数のモデル同士のマージに限定されるが，層レベルのマージは異なるアーキテクチャのモデルをマージすることが可能である．

重みレベルのマージの代表的な手法として，マージ元モデルのパラメータの線形補間により，新たなモデルのパラメータを計算する方法がある．Wortsman らは，事前学習された複数のモデルをこの手法でマージすることにより，画像分類タスクの性能を向上させることに成功している [39]．この他にも，SLERP [32]，TIES [41]，DARE [42]，Task Arithmetic [12] など多くのマージ手法が提案されている．一方，Kim らは，層レベルのマージを行うことで，指示応答能力の高い LLM の構築に成功している [20]．本研究では，最もシンプルで一般的な重みレベルのマージ手法である重みの線形補間によるマージを行い，マージモデルの検索性能を評価する．

3　実 験 設 計

3.1　アドホック検索

アドホック検索は，クエリ q に対して，文書集合 D から k 件の順位付き文書リストを作成するタスクである．このタスクでは，まず文書集合 D の各文書 d_i について，スコア $s(q, d_i)$ を計算する．その後，得られたスコア $s(q, d_i)$ を降順にソートすることで作られる文書リストの上位 k 件を取得することで，k 件の順位付き文書リストを取得することができる．

本研究では，スコア $s(q, d_i)$ を計算する際にバイエンコーダモデルを用いる．バイエンコーダモデルを用いて，スコア $s(q, d_i)$ を計算する仕組みを図 1 に示す．スコア計算のために，まずクエリ q と文書 d_i の意味を表すベクトルをそれぞれ LLM に生成させる．その後，この 2 つのベクトルの内積やコサイン類似度をとることで，得られる値をスコアとする．スコアの計算は文書集合 D 内のすべての文書 d_i に対して行う．

表 1　医療ドメインテストコレクション統計情報.

テストコレクション	クエリ数	文書数
NFCorpus	323	3,633
TREC CT 2021	75	26,162

本研究では，クエリや文書をベクトル化する LLM をモデルマージにより構築する．具体的には，検索タスクに対応するようファインチューニングされたモデルと，検索能力は無いが特定ドメインの文書で継続事前学習されたモデルをモデルマージし，単一のモデルを構築する．このモデルが，特定ドメインのテストコレクションにおいて，検索性能を向上させる高品質なベクトルを生成できるか検証する．

なお，本論文では，マージ元となる検索タスクに対応するようファインチューニングされたモデルを「マージ元検索モデル」，マージ元となる特定ドメインの文書を用いて継続事前学習されたモデルを「マージ元ドメイン特化モデル」と呼ぶ．

3.2　テストコレクション

3.2.1　医療ドメイン

医療ドメインを対象とした実験のデータについて説明する．本研究では，2 つの医療文書検索テストコレクションを用いた．

1 つ目は，NFCorpus [4] である．NFCorpus は，医療文書から有用な情報を正確に検索する能力を評価するために設計されたテストコレクションである．クエリは，NutritionFacts.org から収集し，文書コーパスは，PubMed から収集した生物医学文献を使用している．

2 つ目は，TREC CT 2021 [1] である．TREC CT 2021 のタスクは，患者の症例報告から，その患者にふさわしい臨床試験を検索することである．したがって，クエリは，患者の症例報告であり，文書コーパスは，臨床試験の概要で構成されている．

いずれのテストコレクションも，情報検索分野で使用されるデータを簡単に取得・利用するための Python ライブラリである ir_datasets [25] を通じて取得した．テストコレクション名は，`beir/nfcorpus/test` および `clinicaltrials/2021/trec-ct-2021` である．

NFCorpus においては，クエリとしてクエリ用データフレームの `text` 列をそのまま利用し，文書は文書用データフレームの `title` 列と `text` 列を結合したテキストを用いた．TREC CT 2021 では，クエリとしてクエリ用データフレームの `text` 列を利用し，文書は文書用データフレームの `title` 列，`summary` 列，および `eligibility` 列から抽出した *Inclusion Criteria* 部分を結合したテキストを使用した．ただし，`eligibility` 列から *Inclusion Criteria* を抽出できない場合は，`title` 列と `summary` 列を結合したテキストを文書として利用した．

テストコレクションの統計情報を表 1 に示す．なお，本研究では，TREC CT 2021 は適合性判定がついている文書のみを検索対象文書とした．

表 2　日本語ドメインテストコレクション統計情報.

テストコレクション	クエリ数	文書数
MIRACL	860	8,066
JQaRA	1667	100

3.2.2　日本語ドメイン

日本語ドメインを対象とした実験のデータについて説明する．日本語ドメインにおいても 2 種類のテストコレクションを利用して，実験を行った．

1 つ目は，MIRACL [43] である．MIRACL は，多言語情報検索を対象としたテストコレクションであり，日本語を含む 18 種類の言語に対応している．本研究では，日本語の検証データを利用した．クエリは，多様な質問であり，歴史的な出来事や科学的な事実，文化的な話題に関するものが多い．文書コーパスには，Wikipedia が利用されている．MIRACL においても，医療ドメインの実験と同様に，ir_datasets を利用してデータを取得し，適合性判定のある文書のみを検索対象とした．テストコレクション名は，`miracl/ja/dev` である．クエリとして，クエリ用データフレームの `text` 列を利用し，文書には，文書用データフレームの `text` 列を用いた．

2 つ目は，JQaRA[2] [34] である．JQaRA は，クエリに対し，候補となる 100 件の文書から適合文書を検索するタスクである．クエリは，JAQKET [45] の質問データである．JAQKET とは，クイズを題材にした日本語 QA データセットである．文書コーパスには，Wikipedia が利用されている．クエリとして，データフレームの `question` 列を利用し，文書には，`text` 列を用いた．

日本語ドメインテストコレクションの統計情報を表 2 に示す．なお，JQaRA は，クエリごとに検索対象文書が異なる．1 つのクエリで検索対象とする文書の数は 100 件で，100 件の文書コーパスがクエリ数である 1667 通り存在する．

3.3　マージ元検索モデル

本研究では，マージ元検索モデルとして `intfloat/e5-mistral-7b-instruct` を用いた [38]．このモデルは，`mistralai/Mistral-7B-v0.1`[3]を検索タスクに対応するようファインチューニングしたものである．学習データとして，LLM により生成された，約 50 万件の多様なテキスト埋め込みタスク合成データを利用している．学習データに利用されている言語の数は 93 種類である．合成データに占める日本語の割合は，2.9%である．

モデルの学習に際し，関連するクエリ・文書ペア (q^+, d^+) に指示テンプレートを適用し，クエリ q^+ を以下の形式で指示付きクエリ q^+_{inst} に変換している．

$$q^+_{\text{inst}} = \text{Instruct: } \{\text{task\_definition}\} \,\backslash n\, \text{Query: } \{q^+\} \quad (1)$$

クエリを変換することで，クエリの目的やタスクを明確化し，特定タスクに応じた埋め込み生成を可能にしている．

埋め込みモデルの学習には，InfoNCE 損失 L を利用している．InfoNCE 損失 L は，式 (2) により定義される．

$$L = -\log\left(\frac{\varphi(q^+_{\text{inst}}, d^+)}{\varphi(q^+_{\text{inst}}, d^+) + \sum_{n_i \in N} \varphi(q^+_{\text{inst}}, n_i)}\right) \quad (2)$$

ここで，N は，非適合文書の集合を表し，$\varphi(q, d)$ は，クエリ q と文書 d の関連度を計算する関数である．関連度の計算は式 (3) により行っている．

$$\varphi(q, d) = \exp\left(\frac{1}{\tau}\cos(\boldsymbol{h_q}, \boldsymbol{h_d})\right) \quad (3)$$

$\boldsymbol{h_q}$ はクエリベクトル，$\boldsymbol{h_d}$ は文書ベクトルを表す．クエリと文書の末尾に `[EOS]` トークンを付加し，それぞれモデルに入力する．最終層の `[EOS]` ベクトルを取ることで，クエリと文書の埋め込み $\boldsymbol{h_q}$ および $\boldsymbol{h_d}$ を得ている．温度パラメータ τ は，0.02 に固定している．

3.4　マージ元ドメイン特化モデル

3.4.1　医療ドメイン

医療文書を対象とする実験において，マージ元となる医療特化モデルについて説明する．本研究では，マージ元の医療特化モデルとして，`BioMistral/BioMistral-7B`[4] [21] を用いた．これは，`mistralai/Mistral-7B-Instruct-v0.1`[5]を医療文書で継続事前学習することで構築されたモデルである．学習に利用されたデータは，PMC Open Access Subset[6]である．これは，医療研究論文のコレクションであり，PMC-LLaMA [40]，PubMedBERT [10]，および SciFive [28] の学習にも利用されている．学習に利用されたデータの量は，30 億トークン (約 147 万件のドキュメント) である．完成したモデルは，10 種類の医療質問応答タスクから成るベンチマークで，高い性能を示すことが報告されている．医療文書を対象とする実験では，マージ元ドメイン特化モデルとして医療特化モデルである `BioMistral/BioMistral-7B` を，マージ元検索モデルとして検索タスクに対応するようファインチューニングされたモデルである `intfloat/e5-mistral-7b-instruct` を用いてマージする．そうして得られるモデルが，`intfloat/e5-mistral-7b-instruct` 単体のモデルより，医療文書検索性能が高くなるかを検証する．

3.4.2　日本語ドメイン

日本語文書を対象とする実験において，マージ元となる日本語特化モデルについて説明する．本研究では，マージ元の日本語特化モデルとして，`stabilityai/japanese-stablelm-base-gamma-7b`[7]を用いた．これは，`mistralai/Mistral-7B-v0.1` を日本語タスク向けに継続事前学習することで構築されたモデルである．

2：https://huggingface.co/datasets/hotchpotch/JQaRA
3：https://huggingface.co/mistralai/Mistral-7B-v0.1
4：https://huggingface.co/BioMistral/BioMistral-7B
5：https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.1
6：https://pmc.ncbi.nlm.nih.gov/tools/openftlist/
7：https://huggingface.co/stabilityai/japanese-stablelm-base-gamma-7b

Japanese/English Wikipedia[8], Japanese mc4[9] [29], Japanese CC-100[10], Japanese OSCAR[11], SlimPajama without the Books3 subset[12] [33] のコーパスの組み合わせから約 1000 億トークンが，継続事前学習に利用されている．

日本語文書を対象とする実験では，マージ元ドメイン特化モデルとして日本語特化モデルである `stabilityai/japanese-stablelm-base-gamma-7b` を，マージ元検索モデルとして検索タスクに対応するようファインチューニングされたモデルである `intfloat/e5-mistral-7b-instruct` を用いてマージする．そうして得られるモデルが，`intfloat/e5-mistral-7b-instruct` 単体のモデルより，日本語文書検索性能が高くなるかを検証する．

3.5 マージ方法

本研究では，最もシンプルで代表的なマージ手法である重みの線形補間により，マージを行う．

本研究において，マージ元となるモデルは，すべて Mistral ベースのモデルであり，Mistral は，32 個の層により構築されている．Kevin らは，この 1 つ 1 つの層には，層ごとの役割が存在することを示唆している [26]．そこで，検索性能を向上させるマージ方法についてより深く考察するため，1 層目から 16 層目，17 層目から 32 層目という 2 つの区分に分け，それぞれ異なるハイパーパラメータ設定でマージを行う．具体的には，1 層目から 16 層目までは式 (4)，17 層目から 32 層目は式 (5) に基づいて新たなモデルの重みを計算する．

$$\theta_{\text{lower}}^{(\text{new})} = \alpha_{\text{lower}}\theta_{\text{lower}}^{(1)} + (1-\alpha_{\text{lower}})\theta_{\text{lower}}^{(2)} \tag{4}$$

$$\theta_{\text{upper}}^{(\text{new})} = \alpha_{\text{upper}}\theta_{\text{upper}}^{(1)} + (1-\alpha_{\text{upper}})\theta_{\text{upper}}^{(2)} \tag{5}$$

$\theta_{\cdot}^{(\text{new})}$ は，マージにより新たに構築されるモデルの重みを示す．$\theta_{\cdot}^{(1)}$ および $\theta_{\cdot}^{(2)}$ は，マージ元となるモデルの重みである．本研究では，$\theta_{\cdot}^{(1)}$ がマージ元検索モデルの重みを示す．つまり，$\theta_{\cdot}^{(1)}$ は，`intfloat/e5-mistral-7b-instruct` の重みを示す．$\theta_{\cdot}^{(2)}$ は，マージ元ドメイン特化モデルの重みを示す．つまり，医療文書を対象とする実験においては，$\theta_{\cdot}^{(2)}$ は，`BioMistral/BioMistral-7B` の重みを示し，日本語文書を対象とする実験においては，$\theta_{\cdot}^{(2)}$ は，`stabilityai/japanese-stablelm-base-gamma-7b` の重みを示す．α_{lower} および α_{upper} は，どちらのモデルに重きを置くかを制御するハイパーパラメータである．本研究では，α_{lower} および α_{upper} を，グリッドサーチして各モデルの性能を比較することで，ハイパーパラメータがマージ後のモデルの検索性能に与える影響を確認する．グリッドサーチ対象の重みは，それぞれ 0，0.25，0.50，0.75，1.00 の 5 種類で，計 25 種類の組み合わせで検索性能を比較する．なお，マージ後のモデルのトークナイザと token embedding 層の重みは，マージ元検索モデルのものを活用する．

8：https://dumps.wikimedia.org/other/cirrussearch/
9：https://huggingface.co/datasets/legacy-datasets/mc4
10：https://data.statmt.org/cc-100/ja.txt.xz
11：https://oscar-project.github.io/documentation/
12：https://huggingface.co/datasets/cerebras/SlimPajama-627B

表 3 NFCorpus における nDCG@10

$\alpha_{\text{upper}} \backslash \alpha_{\text{lower}}$	1.00	0.75	0.50	0.25	0
1.00	0.3902	**0.4059**	**0.3963**	0.2695	0.0149
0.75	0.3888	**0.4086**	0.3889	0.2076	0.0230
0.50	0.3868	**0.4057**	0.3626	0.0939	0.0186
0.25	0.3717	0.3757	0.3058	0.0284	0.0125
0	0.2516	0.2163	0.2445	0.0199	0.0149

表 4 TREC CT 2021 における nDCG@10

$\alpha_{\text{upper}} \backslash \alpha_{\text{lower}}$	1.00	0.75	0.50	0.25	0
1.00	0.5604	**0.5843**	0.5578	0.3137	0.0116
0.75	**0.5699**	**0.5874**	0.5374	0.1925	0.0190
0.50	**0.5764**	**0.5903**	0.4815	0.0889	0.0140
0.25	0.5266	0.5454	0.3807	0.0256	0.0110
0	0.3946	0.4335	0.2974	0.0157	0.0116

このとき，α_{lower} と α_{upper} が共に 1.00 であるモデルは，マージ元検索モデルの `intfloat/e5-mistral-7b-instruct` であることを意味する．このモデルが本研究における実験のベースラインとなる．

モデルマージには，Arcee AI が提供するツールである Mergekit [9] を活用する．Mergekit は，マージ方法やハイパーパラメータを設定することで，ユーザが容易に，複数の事前学習済みモデルをマージできる機能を備えたツールである．

4 実験結果

本節では，医療ドメインおよび日本語ドメインにおける実験の結果を示す．各実験では，nDCG@10 を評価指標として使用している．

4.1 医療ドメイン

表 3 は NFCorpus における各モデルの nDCG@10 の値を示している．マージ元検索モデル ($\alpha_{\text{lower}} = 1.00$, $\alpha_{\text{upper}} = 1.00$) の nDCG@10 の値は，0.3902 である．それに対して，表中の数値が太字となっている α の組み合わせでマージ元検索モデルを超える検索性能を示した．特に，$\alpha_{\text{lower}} = 0.75$, $\alpha_{\text{upper}} = 0.75$ であるモデルの nDCG@10 の値は 0.4086 で，マージ元検索モデルと比較して約 2 ポイントの検索性能の向上が見られる．

表 4 は，TREC CT 2021 における各モデルの nDCG@10 の値を示している．マージ元検索モデル ($\alpha_{\text{lower}} = 1.00$, $\alpha_{\text{upper}} = 1.00$) の nDCG@10 の値は，0.5604 である．それに対して，表中の数値が太字となっている α の組み合わせでマージ元検索モデルを超える検索性能を示した．最も性能が向上したモデルは，$\alpha_{\text{lower}} = 0.75$, $\alpha_{\text{upper}} = 0.5$ であるモデルで nDCG@10 の値は 0.5903 である．このモデルは，マージ元検索モデルと比較して約 3 ポイント検索性能が向上している．

4.2 日本語ドメイン

表 5 は，MIRACL における各モデルの nDCG@10 の値を示している．マージ元検索モデル ($\alpha_{\text{lower}} = 1.00$, $\alpha_{\text{upper}} = 1.00$)

表 5 MIRACL における nDCG@10

$\alpha_{\text{upper}} \backslash \alpha_{\text{lower}}$	1.00	0.75	0.50	0.25	0
1.00	0.7549	**0.7773**	**0.7657**	0.6450	0.0135
0.75	**0.7582**	**0.7790**	**0.7658**	0.3120	0.0051
0.50	0.7486	**0.7780**	0.7541	0.0138	0.0017
0.25	0.7134	0.7497	0.6870	0.0024	0.0008
0	0.4605	0.5593	0.3861	0.0009	0.0006

表 6 JQaRA における nDCG@10

$\alpha_{\text{upper}} \backslash \alpha_{\text{lower}}$	1.00	0.75	0.50	0.25	0
1.00	0.6080	**0.6437**	**0.6389**	0.4617	0.2558
0.75	**0.6123**	**0.6464**	**0.6338**	0.2958	0.1551
0.50	0.6047	**0.6401**	**0.6100**	0.1347	0.1291
0.25	0.5650	**0.6131**	0.4888	0.1142	0.1300
0	0.3344	0.4438	0.2403	0.1301	0.1343

の nDCG@10 の値は，0.7549 である．それに対して，表中の数値が太字となっている α の組み合わせでマージ元検索モデルを超える検索性能を示した．最も大きく性能向上を示したモデルは，$\alpha_{\text{lower}} = 0.75$, $\alpha_{\text{upper}} = 0.75$ の組み合わせで，nDCG@10 の値は 0.7790 となり，マージ元検索モデルと比較して 2 ポイント以上検索性能が向上した．

表 6 は，JQaRA における各モデルの nDCG@10 の値を示している．マージ元検索モデル ($\alpha_{\text{lower}} = 1.00$, $\alpha_{\text{upper}} = 1.00$) の nDCG@10 の値は，0.6080 である．それに対して，表中の数値が太字となっている α の組み合わせでマージ元検索モデルを超える検索性能を示した．最も顕著な性能向上を示したのは，$\alpha_{\text{lower}} = 0.75$, $\alpha_{\text{upper}} = 0.75$ であるモデルで，nDCG@10 の値は 0.6464 となり，マージ元検索モデルと比較して約 4 ポイントの検索性能向上が確認された．

5 考　　察

本研究では，医療ドメインおよび日本語ドメインにおけるモデルマージの効果を検証し，検索性能が向上することを確認した．実験結果から得られた知見を以下に示す．

5.1 モデルマージによる検索性能の向上

実験結果から，医療ドメインおよび日本語ドメインにおいて，検索タスクに対応するようファインチューニングされたモデルと，検索能力は持たないが特定ドメインの文書を用いて継続事前学習されたモデルをマージすることにより，検索性能が向上する可能性があることが明らかになった．

この結果は，検索タスクにおいてモデルマージが有効であることを強く示唆している．検索タスクに対応するようファインチューニングされたモデルは，クエリと適合文書の類似度が高くなるようにベクトルを生成する役割を担い，ドメイン特化モデルは専門的な知識や語彙の理解を補完する役割を果たす．この補完関係により，特定ドメインに特化した検索システムの性能向上が実現可能であると考えられる．したがって，モデルマージは，特定ドメインの文書検索性能の改善を目指すうえで有効なアプローチであると結論づけられる．

5.2 ハイパーパラメータ設定の重要性

ハイパーパラメータをグリッドサーチで探索した結果，マージモデルの性能が，マージにおける重み付けを制御するハイパーパラメータである α_{lower} および α_{upper} に大きく依存することが明らかになった．実験結果では，$\alpha_{\text{lower}} = 0.75$ かつ α_{upper} が高い値であるとき，マージ元検索モデルを超える高い検索性能を示す傾向にある．したがって，この設定は，マージ元検索モデルとマージ元ドメイン特化モデルの強みを最大限に引き出していることを示している．一方で，α_{lower} および α_{upper} が低い設定 (0 や 0.25) では，検索性能が低下する傾向が確認された．ハイパーパラメータの設定によりマージ後のモデルの検索性能に大きな差があることは，適切なマージ設定がマージモデルの検索能力において重要な鍵となることを示している．

本実験において，α_{lower} および α_{upper} が低い設定で，検索性能が低下した理由は，マージ元ドメイン特化モデルが検索タスクを直接的に解決する能力を持たないためであると考えられる．マージ元検索モデルは，クエリと適合する文書の意味的な類似度が高くなるようにベクトルを生成する能力を持つ．一方で，マージ元ドメイン特化モデルは，そのようなベクトル生成能力は持たず，専門的な知識や語彙を理解する能力を持つ．したがって，マージ後に検索タスクを解くうえで，主要な役割を担っているのは，検索モデルであり，ドメイン特化モデルは，検索モデルが専門的な知識や語彙を理解する補助的な役割をしていると考えられる．そのため，ドメイン特化モデルに重きを置いたマージ設定において検索性能が低下したと予想される．

5.3 層ごとの重要性分析

NFCorpus を対象とした実験結果によると，$\alpha_{\text{lower}} = 1.00$, $\alpha_{\text{upper}} = 0$ の組み合わせで，nDCG@10 の値は 0.2516 であるが，$\alpha_{\text{lower}} = 0$, $\alpha_{\text{upper}} = 1.00$ の組み合わせで，nDCG@10 の値は 0.0149 である．この結果は，マージモデルの 1 層目から 16 層目の重みを検索モデルの重みとし，17 層目から 32 層目の重みをドメイン特化モデルの重みとした場合には，一定の検索能力を維持できるのに対し，1 層目から 16 層目の重みをドメイン特化モデルとし，17 層目から 32 層目の重みを検索モデルの重みとした場合には，検索能力が著しく低下することを示している．この傾向は，NFCorpus 以外のテストコレクションを用いた実験においても同様であった．これらの結果から，検索タスクにおけるクエリと文書のベクトル生成能力が，主に検索モデルの 1 層目から 16 層目に依存していることが示唆される．

6 ま と め

本研究では，モデルマージをアドホック検索タスクに適用し，その有効性を検証した．その結果，検索モデルとドメイン特化モデルを線形にマージすることで，医療および日本語ドメインにおいて検索性能を向上させられることを示した．

本研究は，アドホック検索におけるモデルマージの可能性を示す初期的な試みであり，今後の研究として以下のような方向

性が考えられる．

まず，より高度なマージ手法の適用が挙げられる．本研究では単純な線形結合を用いたが，他のマージ手法を試行することでさらなる性能向上が期待できる．特に，進化的モデルマージ [2] は，タスクに応じた最適なマージ戦略を進化的アルゴリズムにより自動探索する手法であり，この手法を活用することで，検索性能のさらなる改善が見込まれる．本研究は，このような可能性を示唆する点で意義があるといえる．

また，本研究ではバイエンコーダモデルに焦点を当てたが，クロスエンコーダモデルやその他のアーキテクチャの検索モデルへの適用についても検討する必要がある．幅広いアーキテクチャにおけるモデルマージの効果を分析することで，より多様な検索タスクへの適用が可能となる．

今後の研究では，これらの課題に取り組むことで，モデルマージの有効性をさらに明らかにしていくことを目指す．

謝辞 本研究は JSPS 科学研究費助成事業 JP24K03228，JP22H03905，JP21H03554，JP21H03775 による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] Trec clinical trials track. `https://www.trec-cds.org/2021.html`, 2021. 2025 年 2 月 10 日閲覧.

[2] Takuya Akiba, Makoto Shing, Yujin Tang, Qi Sun, and David Ha. Evolutionary optimization of model merging recipes. *arXiv preprint arXiv:2403.13187*, 2024.

[3] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. MS MARCO: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*, 2016.

[4] Vera Boteva, Demian Gholipour, Artem Sokolov, and Stefan Riezler. A full-text learning to rank dataset for medical information retrieval. In *Proceedings of the 38th European Conference on IR Research*, pp. 716–722, 2016.

[5] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, Ellen M Voorhees, and Ian Soboroff. TREC deep learning track: Reusable test collections in the large data regime. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pp. 2369–2375, 2021.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 17th Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186. Association for Computational Linguistics, 2019.

[7] Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. SPLADE: Sparse lexical and expansion model for first stage ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2288–2292, 2021.

[8] Luyu Gao and Jamie Callan. Condenser: a pre-training architecture for dense retrieval. *arXiv preprint arXiv:2104.08253*, 2021.

[9] Charles Goddard, Shamane Siriwardhana, Malikeh Ehghaghi, Luke Meyers, Vlad Karpukhin, Brian Benedict, Mark McQuade, and Jacob Solawetz. Arcee's MergeKit: A toolkit for merging large language models. *arXiv preprint arXiv:2403.13257*, 2024.

[10] Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare*, Vol. 3, No. 1, pp. 1–23, 2021.

[11] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[12] Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*, 2022.

[13] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*, 2021.

[14] Kalervo Järvelin and Jaana Kekäläinen. IR evaluation methods for retrieving highly relevant documents. In *ACM SIGIR Forum*, Vol. 51, pp. 243–250, 2017.

[15] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

[16] Thorsten Joachims. A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning*, Vol. 97, pp. 143–151, 1997.

[17] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*, 2020.

[18] Omar Khattab and Matei Zaharia. ColBERT: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pp. 39–48, 2020.

[19] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, Vol. 33, pp. 18661–18673, 2020.

[20] Sanghoon Kim, Dahyun Kim, Chanjun Park, Wonsung Lee, Wonho Song, Yunsu Kim, Hyeonwoo Kim, Yungi Kim, Hyeonju Lee, Jihoo Kim, Changbae Ahn, Seonghoon Yang, Sukyung Lee, Hyunbyung Park, Gyoungjin Gim, Mikyoung Cha, Hwalsuk Lee, and Sunghun Kim. SOLAR 10.7B: Scaling large language models with simple yet effective depth up-scaling. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)*, pp. 23–35, 2024.

[21] Yanis Labrak, Adrien Bazoge, Emmanuel Morin, Pierre-Antoine Gourraud, Mickael Rouvier, and Richard Dufour. Biomistral: A collection of open-source pretrained large language models for medical domains. *arXiv preprint arXiv:2402.10373*, 2024.

[22] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, Vol. 33, pp. 9459–9474, 2020.

[23] Ji Ma, Ivan Korotkov, Yinfei Yang, Keith Hall, and Ryan McDonald. Zero-shot neural passage retrieval via domain-targeted synthetic question generation. *arXiv preprint arXiv:2004.14503*, 2020.

[24] Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. Fine-tuning llama for multi-stage text retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2421–2425, 2024.

[25] Sean MacAvaney, Andrew Yates, Sergey Feldman, Doug Downey, Arman Cohan, and Nazli Goharian. Simplified data wrangling with ir_datasets. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2429–2436, 2021.

[26] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems*, Vol. 35, pp. 17359–17372, 2022.

[27] Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas Tezak, Jong Wook Kim, Chris Hallacy, Johannes Heidecke, Pranav Shyam, Boris Power, Tyna Eloundou Nekoul, Girish Sastry, Gretchen Krueger, David Schnurr, Felipe Petroski Such, Kenny Hsu, Madeleine Thompson, Tabarak Khan, Toki Sherbakov, Joanne Jang, Peter Welinder, and Lilian Weng. Text and code embeddings by contrastive pre-training. *arXiv preprint arXiv:2201.10005*, 2022.

[28] Long N Phan, James T Anibal, Hieu Tran, Shaurya Chanana, Erol Bahadroglu, Alec Peltekian, and Grégoire Altan-Bonnet. SciFive: a text-to-text transformer model for biomedical literature. *arXiv preprint arXiv:2106.03598*, 2021.

[29] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, Vol. 21, No. 140, pp. 1–67, 2020.

[30] Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval*, Vol. 3, No. 4, pp. 333–389, 2009.

[31] Viraj Shah, Nataniel Ruiz, Forrester Cole, Erika Lu, Svetlana Lazebnik, Yuanzhen Li, and Varun Jampani. ZipLoRA: Any subject in any style by effectively merging loras. 2023.

[32] Ken Shoemake. Animating rotation with quaternion curves. In *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pp. 245–254, 1985.

[33] Daria Soboleva, Faisal Al-Khateeb, Robert Myers, Jacob R Steeves, Joel Hestness, and Nolan Dey. SlimPajama: A 627B token cleaned and deduplicated version of RedPajama. https://www.cerebras.net/blog/slimpajama-a-627b-token-cleaned-and-deduplicated-version-of-redpajama, 2023.

[34] Yuichi Tateno. JQaRA: Japanese question answering with retrieval augmentation - 検索拡張 (RAG) 評価のための日本語 Q&A データセット.

[35] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Proceedings of the 35th Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[36] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[37] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*, 2022.

[38] Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Improving text embeddings with large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11897–11916, 2024.

[39] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 23965–23998, 2022.

[40] Chaoyi Wu, Weixiong Lin, Xiaoman Zhang, Ya Zhang, Weidi Xie, and Yanfeng Wang. PMC-LLaMA: toward building open-source language models for medicine. *Journal of the American Medical Informatics Association*, p. ocae045, 2024.

[41] Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pp. 7093–7115, 2023.

[42] Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Forty-first International Conference on Machine Learning*, 2024.

[43] Xinyu Zhang, Nandan Thakur, Odunayo Ogundepo, Ehsan Kamalloo, David Alfonso-Hermelo, Xiaoguang Li, Qun Liu, Mehdi Rezagholizadeh, and Jimmy Lin. MIRACL: A multilingual retrieval dataset covering 18 diverse languages. *Transactions of the Association for Computational Linguistics*, Vol. 11, pp. 1114–1131, 2023.

[44] Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. Dense text retrieval based on pretrained language models: A survey. *ACM Transactions on Information Systems*, Vol. 42, No. 4, pp. 1–60, 2024.

[45] 鈴木正敏, 鈴木潤, 松田耕史, 西田京介, 井之上直也. JAQKET: クイズを題材にした日本語 QA データセットの構築. 言語処理学会第 26 回年次大会, 2020.

[46] 阿部健也, 薄羽皐太, 加藤誠. 大規模言語モデルに基づく検索手法の日本語文書検索への適用. 第 16 回データ工学と情報マネジメントに関するフォーラム, T3–A–8–04, 2024.