一般発表 | Track 3: 情報検索・情報推薦・ソーシャルメディア

2026年2月28日(土) 13:00 ~ 15:10 | F会場

[2F] 情報検索

座長:金子 邦彦(福山大学)　コメンテータ:渡辺 知恵美(筑波技術大学)　ジュニアコメンテータ:橋口 友哉(兵庫県立大学)

13:00 ~ 13:25

[2F-01] 文書拡張プロンプト最適化に基づく同時更新文書検索

*伊藤 拓誠[1]、黒川 悠馬[2]、加藤 誠[2,4]、藤田 澄男[3] (1. 筑波大学大学院、2. 筑波大学、3. LINEヤフー株式会社、4. 国立情報学研究所)

13:25 ~ 13:50

[2F-02] クライアントサイド検索システムにおける最適なクエリ拡張手法推定

*花岡 愛梨[1]、丸田 敦貴[1]、加藤 誠[1,2] (1. 筑波大学、2. 国立情報学研究所)

13:50 ~ 14:15

[2F-03] 情報検索システムのための自動ドメイン適応フレームワークの検討

*宮沢 純正[1]、加藤 誠[1,2] (1. 筑波大学、2. 国立情報学研究所)

14:15 ~ 14:40

[2F-04] 地球環境データに対する周辺情報を用いた拡張的メタデータの検討

*清水 敏之[1]、中原 陽子[2]、島井 博行[3] (1. 九州大学、2. 国立情報学研究所、3. 大阪成蹊大学)

14:40 ~ 15:05

[2F-05] [技術報告] 生成AI時代の情報マネジメントにおける検索の役割 ― 非構造化業務データを対象とした設計と運用の知見

*清田 陽司[1,2] (1. 株式会社FiveVai、2. 麗澤大学)

文書拡張プロンプト最適化に基づく同時更新文書検索

伊藤　拓誠† 　黒川　悠馬†† 　加藤　誠†††,†††† 　藤田　澄男†††††

† 筑波大学大学院 人間総合科学学術院　〒 305–8550 茨城県つくば市春日 1–2
†† 筑波大学 知識情報・図書館学類　〒 305–8550 茨城県つくば市春日 1–2
††† 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　〒 101–8430 東京都千代田区一ツ橋 2–1–2
††††† LINE ヤフー株式会社　〒 102–8282 東京都千代田区紀尾井町 1–3 紀尾井タワー
E-mail: †,††{ito.takumi, yuma.kurokawa}@klis.tsukuba.ac.jp, †††, ††††mpkato@acm.org
†††††sufujita@lycorp.co.jp

あらまし　本論文は，文書の更新内容から同時に更新すべき文書を検索する新たなタスクと手法を提案する．本タスク向けに，日本および EU の法文書改正データから同時更新される文書の情報を抽出しデータセットを構築した．構築したデータセットを分析した結果，(1) 既存の検索モデルやクエリ拡張手法の効果が限定的である一方で文書拡張が比較的有効であること，(2) 本タスク向けにモデルを学習するにはデータ量が不十分であることが判明した．これらを踏まえ，文書拡張に用いる大規模言語モデルのプロンプトを最適化することで，学習データが限られた状況でも効果的に同時更新文書を検索する手法を提案する．文書拡張プロンプトをナイーブに評価すると，候補ごとにコーパス全体の再拡張・再索引が必要となり計算コストが大きい．そこで，コーパス全体を再処理せずに計算できる目的関数を設計し，最適化に用いる．構築したデータセットで実験を行った結果，提案手法により同時更新文書検索の効果が向上することを確認した．

キーワード　評価・データセット，文書拡張，プロンプト最適化

1　はじめに

大規模な文書集合の管理と維持は，企業・行政における法務文書の運用やソフトウェア開発における仕様書管理など，さまざまな領域で重要な課題となっている [1, 2]．これらの文書は独立して存在するのではなく，互いに参照関係や論理的な依存関係を持つことが多い．そのため，ある文書に更新が生じると，関連する他の文書も整合性を維持するために同時に更新する必要がある．本論文では，ある文書の更新に伴って同時に更新する必要が生じる文書を**同時更新文書**と呼ぶ．

しかし，文書集合の中から漏れなく同時更新文書を特定することは容易ではない．その主な理由は，更新の対象が多岐にわたることがある点 [3] や，文書間の依存関係が明示的なリンクで表現されない場合があり，内容の文脈や専門的な領域知識に基づく判断が必要となる点にある．従来，この作業は対象領域に関する知識を有し，文書集合全体の内容を把握する専門家によって行われてきた．しかし，高度な知識を有する人材は不足している上，人手による探索はコストが大きいだけでなく，見落としのリスクも伴う．したがって，更新情報から同時更新文書を自動的に特定する支援技術が求められる．そこで，本研究では，ある文書の更新に関する情報をクエリとし，被影響文書を特定する新たな検索タスクと手法を提案する．本タスクの目的は，与えられた更新内容に基づき，整合性を維持するために同時に更新が必要となる文書を大規模な文書集合の中から網羅的に検索することである．タスクに加え，タスク向けの手法を提案することにより，人手による探索コストを削減し，人材不足の緩和と業務の効率化に寄与することを目指す．

本タスクの確立と解決に向け，まず，データセットを構築する．データセットの題材としては，厳密な整合性が求められるのに加え，ある文書の更新が他の文書の更新を必要とする事例を多く含む法律文書を選定した．具体的には，日本および EU の法文書改正データから同時に更新される条文の情報を抽出し，言語や法体系の異なる 2 種類のデータを収集した．収集したデータに対し，検索タスク向けに処理を行ったところ，約 1,600 件のクエリ，約 90,000 件の文書からなるタスク向けのデータセットが 2 種類構築された．構築したデータセットを分析した結果，既存の疎検索および密検索モデルでは，更新内容と被影響文書の間の潜在的な関係性を十分に捉えられず，検索効果が限定的であることが判明した．

これらを踏まえ，本論文では文書拡張を提案手法の基盤として採用し，プロンプト最適化を用いた文書拡張手法を提案する．プロンプト最適化では，モデルの学習と比較して少量の学習データでタスクの性能を向上できることが報告されている．一方で，文書拡張プロンプトの最適化をナイーブに行うと，候補プロンプトを評価するたびに全文書の再拡張と再索引が必要となり，反復的な最適化の実行において計算コストが膨大となる課題がある．そこで，本研究では，コーパス全体を再処理せずに計算できる報酬関数を設計し，プロンプト最適化に用いる．具体的には，代表的なデータのサンプリングにより計算量削減を行なった上，少量のデータで効果的な最適化ができるよう利用するデータの選別やランクによる重み付けを行う．

提案手法の有効性を検証するため，構築した 2 種類のデータセットを用いて実験を行った．実験の結果，最適化後のプロンプトによる文書拡張は，日本法データセットおよび EU 法データセットの双方において，最適化時に用いた検索モデルの検索性能を向上させた．このことは，提案手法により学習データが限られた状況でも効果的に被影響文書検索の性能を向上できることを示唆している．

本論文の貢献を以下に示す：

1. ある文書の更新に伴って同時に更新する必要がある文書を特定する，被影響文書検索タスクを定義した．
2. 日本および EU の法律文書に基づき，2 種類の被影響文書検索タスクのデータセットを構築し，基本的な分析を行った．
3. 被影響文書検索に適した文書拡張を実現するため，学習データが乏しく拡張の正解データが存在しない状況でも利用できるプロンプト最適化手法を提案した．

本論文の構成は以下の通りである．2 節では，関連研究について述べる．3 節では，問題設定を説明し，データセットの構築方法と分析結果を示す．4 節では，提案手法について述べる．5 節では，実験結果を示す．最後に，6 節で今後の課題と共に本論文のまとめを述べる．

2 関 連 研 究

本節では，本研究の関連研究について述べる．まず，既存の文書更新タスクのデータセットが本研究の同時更新文書検索に適用可能かを検討する．次に，文書拡張手法について，代表的な手法と本研究との関係を説明する．

2.1 文 書 更 新

文書更新に関する研究では，更新後テキストの生成を主題とする研究が多数行われ，データセットも複数構築されている．本研究が対象とする同時更新文書検索は，更新された文書を手掛かりとして同時に更新すべき文書を検索する点に特徴がある．既存データセットを本タスクへ適用するためには，以下の 3 つの要件を満たすものを選ぶ必要がある．

1. 複数の内容が同時に更新され得ること
2. 更新が一つの文書内に留まらず複数の文書に渡り得ること
3. 同時更新箇所の特定が単純な手法では困難であること

要件 1 は，同時更新文書を扱うために必要である．しかし，多くの文書更新データセットは単一の文書内の単一箇所の編集を対象とするため，この要件を満たさない．EditEval [4] はさまざまな文書編集タスクを集めたベンチマークであり，言い換えを扱う STSB [5]，文法の誤りを訂正する JFLEG [6]，引用時のコメントを追記する WAFER-INSERT [7] などが含まれる．しかし，これらは一度の更新で複数箇所が同時に更新される状況を扱わない．EditEval に含まれないデータセットでも，同様に，要件 1 を満たさないものが多い [8, 9]．

次に，参照関係に基づいて更新を伝播させる設定として，FRUIT [10] がある．FRUIT では，Wikipedia のスナップショット間の差分から対象記事の更新箇所を抽出し，他の記事に追加された対象記事に関する言及を入力として，対象記事へ更新情報を反映する．このため，複数の内容および複数文書が同時更新され，要件 1 および 2 を満たす．一方で，同時更新される文書や箇所は対象記事のタイトルや関連する固有表現が含まれる場合が多く，単語一致に基づく検索でも候補の絞り込みが比較的容易であると考えられる．そのため，要件 3 を満たさない．

さらに，査読コメントに基づく論文の更新を扱うデータセットとして Re3 [11] がある．これは査読コメントを受け論文を修正する際に論文内で複数箇所が同時に更新されるため，要件 1 を満たす．また，修正対象の箇所は，語彙および意味的に類似しないことも多いため，要件 3 も満たす．しかし，論文という単一文書内の更新を対象としており，探索範囲が狭く検索が必要な課題が存在しないため，要件 2 を満たさない．

以上より，本研究に必要な要件を満たす既存のデータセットは存在せず，新たにデータセットを構築する必要がある．

2.2 文 書 拡 張

文書拡張は，検索対象の文書にテキストを付与し，索引時に文書の情報を拡充する手法である．クエリと文書の表現の差異，特に語彙的なギャップを緩和し，検索時の関連文書の漏れを抑制することで再現率の向上に寄与することが期待される．

Nogueira らは，クエリと適合文書の対応に基づいて学習した生成モデルにより，文書から生成した疑似クエリを文書へ付与する枠組みを提案した [12]．これは代表的な文書拡張の手法であり，学習対象のモデルを T5 へ変更したり [13]，付与する疑似クエリを厳選したり [14] することで性能向上がなされている．また，Boudin らは，論文アブストラクトからキーフレーズを生成して文書へ付与することで学術文書検索の性能が向上することを示した [15]．しかし，本研究が扱う同時更新文書検索では，更新内容を検索するためのクエリや文書に付与すべきテキストの正解データが存在せず，上記手法のように拡張に用いるテキストの生成を教師ありで学習することは難しい．

拡張の最適化に関連して，下流タスクの性能に基づき拡張を end-to-end で学習する研究も存在する．Tang らは，短文に対する拡張を下流の分類タスクに対して最適化する枠組みを提案した [16]．分類タスクでは，各入力サンプルに対して局所的かつ微分可能な損失関数を定義できるため，目的関数の算出および最適化が比較的低コストである．一方，検索タスクでは，評価指標が順位全体に依存する非局所かつ微分不可能な関数であるため，同様の枠組みを直接適用することは困難である．

このような制約に対し，本研究では検索性能に基づく報酬を用いて，LLM による文書拡張で用いるプロンプトの最適化を行うことで有効な文書拡張手法を提案する．

3 データセット

本節では，問題設定と，本タスク向けに構築した日本法および EU 法データセットの構築方法と分析結果を示す．

3.1 問 題 設 定

本研究では，ある文書の更新に伴って内容を更新する必要が

ある他の文書を検索する**同時更新文書検索タスク**を新たに提案する．以下，タスクの定式化を行う．法律文書や社内文書など特定の目的で管理される文書の集合を D とする．ある文書 $d \in D$ が更新されたとき，更新前の文書を d，更新後の文書を $\tilde{d}$ と表す．文書 d の更新に伴い内容の更新が必要となる他の文書の集合を，同時更新文書集合 $D' \subseteq D \setminus \{d\}$ とし，$d' \in D'$ を同時更新文書と呼ぶ．本タスクでは，ある文書が更新されたとき，更新前後の文書 d と $\tilde{d}$ から生成したクエリ q を用いて，文書集合 $D \setminus \{d\}$ を対象に検索を行う．クエリ集合を Q とし，クエリ生成関数を $g : D \times D \to Q$ と定義する．その際，文書 d の更新からクエリ $q = g(d, \tilde{d}) \in Q$ が生成され，そのクエリに対する適合文書集合は D' となる．

3.2 データセット構築

本節では，上記で定義したタスク向けのデータセット構築方法について説明する．データセットの題材としては，改正法令と呼ばれる法令の更新情報を利用する．

3.2.1 改正法令

法域によっては，法令の更新は改正法令として公布され，更新対象となる法令と，その更新箇所が明示される．一度の更新で同時に複数の法令が更新され得るのに加え，更新される文書の情報が公開情報として提供されるため，更新対象の同定を人手によるアノテーションをせずに行える．

一方で，改正法令が明示するのは同一の更新で同時に更新された法令の集合であり，特定の法令の更新が他の法令の更新を引き起こしたかどうかは，改正法令の記述のみからは判別できないことが多い．本研究では，更新の因果関係の同定は行わず，同時に更新された法令の集合を，同一の更新により更新が必要となる法令の集合として扱う．

3.1 節の問題設定に当てはめると，法令に含まれる各条文が文書，全法令の各条文の集合が D に相当する．ここで，同時に更新された条文集合を A とし，A から一つの条文を選択して更新が与えられた文書 d とみなす．残りの $A \setminus \{d\}$ を，d の更新に伴い同時更新が必要な文書集合 D' とみなすことで，同時更新文書検索タスクの学習例と評価例を構成できる．

3.2.2 改正法令データの収集

改正法令は複数の法域から収集することができる．本研究では，法体系と言語の多様性を確保しつつ収集可能性も考慮し，日本法と EU 法を選定した．日本法は主として国内で適用される法令，EU 法は EU 加盟国全体で適用される法令であり，両者は法体系や言語が十分に異なる．この差異は，条文の表現や参照表現の形式に影響し，タスクの難易度や検索に有効な手掛かりが異なるデータセットを構築できる．

改正法令データの収集にあたっては，両データセットの改正法令の件数が同程度となるように収集範囲を調整し，日本法では 2019 年以降，EU 法では 2010 年以降に公布された改正法令を収集した．EU 法は複数言語で公開されるが，データの欠損が最も少ない英語版を対象とした．

また，改正法令ごとに更新対象となる法令の情報と更新前後の本文を収集した．日本法は e-Gov 法令検索 [1]で提供される法令 API を用い，EU 法は EUR-Lex[2]のページをスクレイピングすることでデータの収集を行なった．収集したデータはオープンデータである CC0，または CC BY 4.0 のライセンスで提供されている．データセット構築時には，CC BY 4.0 における再利用の条件である出典の明記と変更内容の明示処理を施した．

3.2.3 収集した改正法令データの処理

改正法令からは，法令単位で更新箇所の情報を収集できるが，更新前後の条番号の対応関係は明示されない．例えば，更新前の第 3 条が更新後の第 4 条に移動した場合でも，この対応関係は機械可読な形では提供されない．本研究では，条文単位の更新データを得るために，条文テキストの一致と類似度に基づき，更新前後の条文を対応付けた．

各条文を条番号，キャプション，本文の 3 フィールドに分割した．まず，更新前後で 3 フィールドが全て一致する条文ペアは更新なしとして除外した．次に，本文が完全一致し，条番号またはキャプションのみが更新された条文ペアは，内容の更新が軽微であるため軽微な更新として扱い除外した．本文が完全一致しない条文のうち，キャプションが完全一致する場合は，その一致に基づいて対応付けた．本文とキャプションの双方が一致しない条文についてのみ，本文の類似度に基づく対応付けを行った．類似度として Dice 係数を用い，日本法では文字単位，EU 法では単語単位で算出した．Dice 係数が高い順に条文ペアを列挙し，Dice 係数が 0.7 以上である条文ペアを貪欲法により一対一で採用した．Dice 係数が 0.7 未満である条文ペアについては，大幅な削除や新規追加により類似度が低下した場合でも共通部分が保持される状況を想定し，そのような更新を検出しやすい Simpson 係数を追加で算出した．Simpson 係数が 0.95 以上であれば対応付けの候補として扱い，Dice 係数が 0.7 未満かつ Simpson 係数が 0.95 未満である条文ペアは，条文の更新ではなく条文の削除と新規追加が生じたものとして扱った．これらの閾値は実例を確認しながら調整し，本来は対応付けられない条文ペアが対応付けられる誤りを抑制することを優先し設定した．ただし，Dice 係数と Simpson 係数がいずれも閾値を下回る場合でも，条番号が一致し，かつ対象の改正法令において同一法令内の条番号の移動が検出されない場合は，条番号の一致に基づき対応付けを行った．

対応付けが成立した条文ペアについては，内容の更新が軽微である場合を除外した．軽微な更新の例として，誤字修正，単純な表現の置換，参照先条番号の更新に伴う参照表現の更新などが挙げられる．このような更新は，他条文との論理的関係を十分に反映しない，あるいは語彙的一致に基づく検索で容易に発見可能であるため，本タスクの目的に適さない．そこで，正規表現を設定して検出したほか，本文テキストの Dice 係数が 0.99 以上である条文ペアも軽微な更新として除外した．

検索タスクでは固定した一時点の法令集合をコーパスに設定する必要がある．本研究では，各法令について，収集範囲内

1：https://laws.e-gov.go.jp/
2：https://eur-lex.europa.eu/

表 1　データセットの統計情報

	改正法令数	クエリ数	文書数	平均クエリ長	平均文書長	Avg. D/Q
日本法	363	1,653	90,170	679.9 文字	380.1 文字	26.3
EU 法	340	1,590	91,361	380.8 単語	238.7 単語	18.8

表 2　データセット分割ごとの改正法令数とクエリ数

	訓練		検証		テスト	
データセット	改正法令数	クエリ数	改正法令数	クエリ数	改正法令数	クエリ数
日本法	121	563	121	555	121	535
EU 法	113	504	114	502	113	503

で最初に行われた改正における更新前のバージョンをコーパスに用いた．一部の条文のみに更新が生じる場合でも，そのバージョンの法令全体をコーパスに用いた．

同一条文が複数回更新される場合，2 回目以降の更新は更新前の条文がコーパスにあるバージョンと一致しない問題が生じる．そこで，各条文について最初の更新のみを対象とし，同一条文に対する 2 回目以降の更新は除外した．

3.2.4　検索データセットの構築

本節では，前節の処理を施した改正法令データから検索データセットを構築する方法を述べる．各改正法令について，前節の対応付け結果に基づき，更新または削除が行われた条文の集合を A とする．条文の新規追加は更新前の条文テキストが存在せず，検索クエリと検索対象を対応付けられないため，A に含めない．また，適合文書が少なくとも 1 件存在するように，$|A| \geqq 2$ を満たす改正法令のみを対象とする．

A から一つの条文 d を更新が与えられた文書として選択し，残りの $A \setminus \{d\}$ を d に対する同時更新文書集合 D' として，検索データセットのクエリとそれに対する適合文書集合を構成する．このとき，基準となる文書 d の選び方を変えて同じ処理を行うことで，一つの改正法令から複数のクエリとそれに対する適合文書集合が生成できる．ただし，$|A|$ が大きい場合は d の選び方，すなわちクエリ数も比例して大きくなり，特定の改正法令が全体の評価に与える影響が大きくなるため，改正法令ごとのクエリ数を最大 5 件に制限した．

また，コーパスとしては全法令の条文を用いる．このため，改正法令の収集期間に更新がなかった法令についても，収集開始時点のバージョンをコーパスに追加した．

表 1 に構築したデータセットの統計情報を示す．Avg. D/Q はクエリあたりの平均適合文書数を表す．文書長およびクエリ長は，日本法データセットでは文字数，EU 法データセットでは単語数で算出した．また，表 2 に訓練・検証・テストデータごとの改正法令数とクエリ数を示す．改正法令数はいずれも 300 件台で，そこから作成したクエリ数はいずれも約 1600 件程度と，一般的な検索データセットと比べるとクエリ数は少ない．一方で，コーパスは約 9 万文書から構成され，検索対象の文書集合としては一定の規模を有する．

3.3　データセット分析

3.3.1　実 験 設 定

本節では，構築した 2 つのデータセットの特徴やタスクとしての課題を明らかにするために，既存検索モデルや LLM を用いた検索改善手法で検索実験を行い，結果を分析する．

表 3　既存検索モデルによる日本法データセットの検索結果

measure	BM25	mDPR	mContriever	mE5-small	mE5-base	mE5-large
R@10	0.195	0.174	0.195	0.194	0.177	**0.198**
R@100	0.418	0.394	0.412	0.422	0.393	**0.424**
R@1000	0.634	0.610	0.625	0.631	0.615	**0.638**
nDCG@10	**0.273**	0.244	0.261	0.265	0.247	0.263
nDCG@100	0.307	0.286	0.301	**0.307**	0.288	0.306
nDCG@1000	**0.376**	0.354	0.369	0.374	0.358	0.374

表 4　既存検索モデルによる EU 法データセットの検索結果

measure	BM25	mDPR	mContriever	mE5-small	mE5-base	mE5-large
R@10	**0.149**	0.085	0.094	0.126	0.122	0.120
R@100	**0.311**	0.179	0.207	0.275	0.284	0.287
R@1000	0.478	0.320	0.368	0.460	0.464	**0.488**
nDCG@10	**0.201**	0.120	0.137	0.169	0.164	0.159
nDCG@100	**0.233**	0.136	0.156	0.199	0.200	0.200
nDCG@1000	**0.279**	0.174	0.200	0.249	0.250	0.255

以降の検索実験では，データセットのテストデータを用いる．また，クエリとしては更新内容が与えられた文書の更新前のテキスト全体，すなわち，$q = g(d, \tilde{d}) = d$ を用いる．更新後のテキストや更新前後の差分箇所をクエリとする場合の実験も行ったが，いずれも更新前のテキストと比較して性能が低かったため，本論文では結果を割愛する．

検索モデルとしては，伝統的な単語一致に基づく検索モデルとして BM25 [17] を採用した．また，意味的類似に基づく検索を行う密検索モデルに，日本語と英語の両方のデータセットで利用できる多言語検索モデルの mDPR [18]，mContriever [19]，mE5 [20] を採用した．いずれの検索モデルも，Pyserini [21] の実装を用いて実験を行った．

3.3.3 節から 3.3.5 節では LLM を用いた検索改善手法として，キーワード形式のクエリ生成，文書拡張，そしてリランキングの実験を行った．モデルには GPT-4.1 Mini [22] (openai/gpt-4.1-mini-2025-04-14) を用い，temperature は 0.0 に設定した．利用したプロンプトは Github レポジトリに掲載する [3]．

評価指標には R@k と nDCG@k を用いる．本タスクでは，検索結果を人手で確認し，同時に更新が必要かどうかの最終判断を行うことが想定される．このため，検索結果に同時に更新が必要な文書を漏れなく含めることが重要である．上記を評価する指標として R@k を採用する．加えて，効率的な確認のために提示された候補の上位に同時に更新が必要な文書が位置付けられていることが望ましい．上記を評価する指標として，一般的なランキングの評価指標である nDCG@k を用いる．

以降では，まず既存検索モデルのベースラインにより本タスクの課題を確認し，その後，課題に対する仮説を検証するためにキーワードクエリ，文書拡張，リランキングを評価する．

3.3.2　既存検索モデルのベースライン

表 3 に日本法データセット，表 4 に EU 法データセットの検索結果を示す．

3：https://github.com/kasys-lab/simultaneous-update-document-retrieval-prompts

表 5　キーワードクエリおよび文書拡張による R@1000 の比較

	日本法			EU 法		
Model	拡張なし	キーワードクエリ	文書拡張	拡張なし	キーワードクエリ	文書拡張
BM25	0.634	**0.669**	0.649	0.478	0.503	**0.517**
mDPR	0.610	0.660	**0.676**	0.320	**0.464**	0.453
mContriever	0.625	0.590	**0.633**	0.368	0.343	**0.385**
mE5-small	0.631	0.634	**0.636**	0.460	0.466	**0.501**
mE5-base	0.615	0.632	**0.642**	0.464	0.484	**0.510**
mE5-large	0.638	0.641	**0.651**	0.488	0.472	**0.491**

表 3 を見ると，日本法データセットでは R@k はいずれも mE5-large が最も高く，続く BM25 も同程度の性能を示していることが分かる．一方で，nDCG@10 は BM25 が最も高く，次いで mE5-small の性能が高い．表 4 を見ると，EU 法データセットでは R@k および nDCG@k のほとんどの指標で BM25 の性能が最も高いことが分かる．一般的な ad-hoc 検索タスクの MIRACLE [23] では，mE5 の性能が BM25 を大きく上回ることが報告されている [20]．これらの結果からは，本タスクにおいて単語一致に基づく検索が有効である可能性，および，通常の検索タスクと本タスクの性質が大きく異なり，通常の検索タスク向けに学習された密検索モデルが十分にタスクに対応できていない可能性が示唆される．

また，日本法，EU 法のデータセットのどちらも評価指標の値が全体的に低いことが読み取れる．上位 1000 件はコーパス全体の約 90 分の 1 に相当するが，R@1000 は日本法で 0.634，EU 法で 0.478 に留まり，網羅性が不十分である．nDCG@10 の値も日本法で 0.273，EU 法で 0.201 と低く，上位候補の順位付けにも改善の余地がある．

上記の結果を踏まえて検索効果が不十分である要因を考察し，それを改善するための仮説を以下に示す．構築したデータセットにおけるクエリの訓練データ数約 1,000 件と小規模であり，タスク向けのモデルを学習するのに十分なデータが存在しないため，教師なしでの改善手法に着目する．

1. 条文全体をクエリとしているため検索に有効でない語がスコアに与える悪影響が大きく，検索に有効な語のみをクエリに用いることで不適合文書のスコアが上昇しづらくなり，検索における適合文書の網羅性が向上する．
2. 条文同士を比較するだけでは同時更新が必要かを判断するのに必要な文脈が不足しており，文書に文脈情報を付加することで検索における適合文書の網羅性が向上する．
3. 通常の検索モデルでは同時に更新されるという文書間の関係性を考慮できず，汎用的なテキスト処理能力の高い LLM による追加処理で検索効果が向上する．

以降の節では，仮説 1 から仮説 3 を検証するための実験を行う．

3.3.3　キーワードクエリ

仮説 1 を検証するため，更新前の文書全体をクエリとして用いるのではなく，更新前後の文書から検索に寄与すると考えられる語を生成し，短いクエリとして検索する実験を行った．キーワード生成には大規模言語モデルを用いてキーワードを生成した．評価指標には，仮説 1 が適合文書の網羅性の改善に寄与するかを確認するために R@1000 を用いた．

実験結果を表 5 に示す．表からは，両方のデータセットで BM25 および mDPR の R@1000 が改善するものの，その他のモデルでは一貫した改善は見られないことが分かる．単語一致検索の性能が向上したことから，検索に有効でない語がスコアに与える悪影響を軽減できる可能性が示唆されたが，密検索モデルでは一貫した改善が見られず，仮説 1 の有効性は限定的であると考えられる．

表 6　リランキングの有無による nDCG@10 の比較

データセット	リランクなし	リランクあり	リランクあり w/ 更新後の文書
日本法	0.273	0.283	**0.289**
EU 法	0.201	**0.202**	0.200

3.3.4　文書拡張

仮説 2 を検証するため，条文のみでは不足する文脈を文書側に付加する文書拡張を評価した．不足する文脈を補うために，各条文の核心的な内容や社会的な目的を説明するテキストを生成した．条文間の語彙・意味的な類似度が低くても，この説明文を通じて関連性を捉えやすくなることが期待される．評価指標には，仮説 2 が適合文書の網羅性の改善に寄与するかを確認するために R@1000 を用いた．

文書拡張には大規模言語モデルを用いて説明文を生成した．文書集合に含まれる全ての文書に対して説明文を生成し，説明文と文書をこの順に結合したテキストに対して検索を行った．

表 5 に示す通り，文書拡張は両データセットにおいて全てのモデルで一貫して R@1000 を改善した．前節のキーワードクエリ手法と比較しても，ほとんどのモデルでより大きな改善が得られている．同時更新文書は更新された文書と直接の語彙・意味的な類似度が低い場合があり，文書拡張により，条文同士の比較だけでは捉えられない関連性を捉えた可能性が示唆される．

3.3.5　リランキング

仮説 3 を検証するため，既存の検索モデルで得た検索結果に対して大規模言語モデルによるリランキングを適用し，同時に更新が必要かどうかに基づく順位付けの改善を評価した．検索モデルには両データセットで R@10 が高い BM25 を用い，検索結果上位 10 件をリランキングした．プロンプトはリランキング対象の文書のテキストを直接プロンプトに含めて並べ替えさせる RankGPT [24] を参考に設計し，更新後文書をプロンプトに含める場合と含めない場合の 2 通りの手法を評価した．評価指標には，仮説 3 が順位づけの改善に寄与するかを確認するために nDCG@10 を用いた．

表 6 に，リランキング適用前後の nDCG@10 を示す．EU 法データセットでは改善は限定的であったが，少なくとも日本法では LLM によるリランキングが有効であることが分かる．また，既存の検索モデルのクエリとしては効果的でなかった更新後の文書が，LLM によるリランキングでは有効に働いていることが分かる．これは，LLM が更新前後の文章から更新内容を把握し，その内容に基づいて同時に更新が必要な文書を識別できる可能性があることを示唆している．リランキングにより nDCG が改善することが示されたため，リランキングの前段階の検索において R@k を高めることで，リランキングの効果がより高まることも期待される．

構築したデータセットはクエリ数が小規模であり，タスク向けのモデルを学習するのに十分なデータが存在しないため，教師なしで検索を改善する手法を検討した．単順なクエリ設計の工夫による改善は限定的であったが，文書拡張は検索における適合文書の網羅性を，大規模言語モデルによるリランキングは上位の順位づけを改善した．以上より，同時更新文書検索では，文書拡張により網羅性を高めた上で，リランキングにより順位づけの精度を高める構成が有効であると考えられる．特に，リランキングの効果を最大化するためには，前段階の検索の R@k を左右する文書拡張の品質が重要となる．

4 提案手法

本節では，同時更新文書検索における文書拡張手法の性能を，学習データが限られた状況でも改善させるためのプロンプト最適化手法を提案する．提案手法は，LLM による文書拡張を前提とし，拡張に用いるプロンプトを検索性能に基づく報酬で最適化する．最適化時に候補プロンプトを R@k などで単純に評価すると，候補ごとにコーパス全体の再拡張・再索引が必要となり計算コストが膨大になるという課題がある．そこで本研究では，コーパス全体を再処理せずに計算できる効率的な報酬関数を設計し，最適化に用いる．

4.1 文書拡張のためのプロンプト最適化

第 3 節の分析で示した通り，本タスクではクエリと文書の語彙・意味的な類似度が低く，文書の背景情報を LLM による文書拡張で補う手法が有効であった．

文書拡張手法を改善するために，本研究では文書拡張に用いるプロンプトを最適化する手法を用いる．LLM に入力するプロンプトをタスクに適した報酬関数に基づいて最適化することで，モデルの学習と比較して少量の学習データでタスクの性能を向上できることが報告されている [25–27]．プロンプト更新の具体的手法は複数提案されており，候補プロンプトの生成，評価と選択を反復する枠組みがしばしば用いられる．

文書拡張におけるプロンプトは，第 3 節で利用したような，条文の核心的な内容や社会的な目的といった文脈情報をどのように生成させるかを規定するため，検索性能に直接影響する．しかし，本タスクでは拡張テキストそのものの正解データが存在しないため，拡張されたテキストに基づく報酬算出は困難である．そこで本研究では，検索効果に基づく報酬を直接最適化の基準として，プロンプトを反復的に更新する手法をとる．

4.1.1 最適化問題の定式化

文書拡張に用いるプロンプトとして，初期プロンプトを P_0，最適化により更新されるプロンプトを P とする．また，学習に用いるクエリ集合を $\mathcal{Q} \subset Q$ とする．本研究では，クエリ q に対するプロンプト P の報酬を $r(q;P) : Q \to \mathbb{R}$ として定義し，その平均を最大化するプロンプトを求める．すなわち，

$$\max_{P} \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} r(q;P) \tag{1}$$

を解く．報酬 $r(q;P)$ の定義は，最適化手法により異なる．単純な手法としては，プロンプト P を用いた手法で全文書を拡張した上で検索を実行し，R@k を用いることが考えられる．向上させたい評価指標を直接報酬関数として用いることは理想的であるが，文書拡張プロンプトの最適化においては，候補プロンプトを評価するたびに全文書の再拡張と再索引が必要となり，反復的な最適化において計算コストが膨大となる課題がある．

4.1.2 反復的なプロンプト更新

式 (1) で定義される目的関数を $J(P)$ とおき，$J(P)$ を最大化するための反復的なプロンプト更新を考える．反復回数を T とし，初期プロンプト P_0 から開始する．プロンプトが取りうる文字列全体の集合を $\mathbb{P}$ とし，反復 t までに得られたプロンプト列を $\mathbf{P}_{0:t} = (P_0, \ldots, P_t) \in \mathbb{P}^{t+1}$ と定義する．反復 t において，プロンプト候補生成関数 $\mathcal{G}_t : \mathbb{P}^{t+1} \to 2^{\mathbb{P}}$ は，$\mathbf{P}_{0:t}$ の情報を用いて候補プロンプト集合 $\mathcal{P}_t \subset \mathbb{P}$ を生成する．すなわち，$\mathcal{P}_t = \mathcal{G}_t(\mathbf{P}_{0:t})$ と表せる．プロンプト候補生成関数 $\mathcal{G}_t$ は，強化学習に基づく手法や遺伝的アルゴリズムに基づく手法など，さまざまな方法で設計できる．時点 $t+1$ では，候補プロンプト集合 $\mathcal{P}_t$ から $J(P)$ を最大化するプロンプトを選択し，次のプロンプトを $P_{t+1} = \arg\max_{P \in \mathcal{P}_t} J(P)$ として更新する．最終的に，反復全体で最大の目的関数値を与えるプロンプトを $P^* = \arg\max_{P_t;\, t \in \{0,\ldots,T\}} J(P_t)$ として採用する．

次の節では，$J(P)$ の評価に用いる報酬関数 $r(q;P)$ を，コーパス全体の再処理を伴わずに計算できる形で設計する．

4.2 効率的な報酬関数

本節では，前節で述べた反復的なプロンプト更新において，候補プロンプト P の評価を効率的に行うための報酬関数 $r(q;P)$ を設計する．第 4.1.1 節で述べた通り，R@k などの評価指標を直接報酬とすると，候補プロンプトの評価ごとにコーパス全体の再拡張・再索引が必要となり計算コストが膨大となる．そこで本研究では，代表的なデータのサンプリングにより計算量削減を行なった上，少量のデータで効果的な最適化ができるよう利用するデータのさらなる選別やランクによる重み付けを行う．以降では，まずサンプリング手法を説明し，次にペアに関する制約と正例の重みを導入し，最後に提案する報酬関数を示す．

4.2.1 サンプリングによる計算量削減

候補プロンプト P の評価を，クエリ q ごとに抽出した正例・負例の部分集合上で行うことにより計算量を大幅に削減する．クエリ q に対する正例文書集合を $D^+(q)$ とする．これは，第 3 節で定義した同時更新文書集合 D' に対応し，$D^+(q) = D'$ である．検索対象の文書集合を $D(q) = D \setminus \{d\}$，負例文書集合を $D^-(q) = D(q) \setminus D^+(q)$，検索モデルが付与するスコア $s(q,d;P_0)$ が高い順に上位 k 件からなる文書集合を $D_k(q) \subset D(q)$ とする．$s(q,d;P)$ は，後述するプロンプト P に基づく検索スコアである．また，上位 k 件に含まれる負例集合を $D_k^-(q) = D_k(q) \setminus D^+(q)$ と定義する．$D_k^-(q)$ は検索モデルのスコアが高い負例であり，正例と識別しづらいハードネガティブに相当する．本研究では，一様分布 $\mathcal{U}(\cdot)$ に基づき，以下のようにクエリごとの正例・負例のサンプリングを行う．

$$\hat{D}^+(q) \sim \mathcal{U}(D^+(q)), \qquad \hat{D}^-(q) \sim \mathcal{U}(D_k^-(q)) \tag{2}$$

4.2.2 初期プロンプトによる拡張結果に基づくデータ選別

本研究では，正例文書 d^+ と負例文書 d^- のスコアの比較に基づいて報酬を計算する．しかし，正例と負例を一様に組み合わせると初期プロンプト P_0 における順位付けにおいて既に正例が負例より上位に位置している組み合わせも含まれる．そのような組み合わせは改善余地が小さく，最適化への寄与は限定的である．そこで本研究では，以下のように各正例 d^+ に対し，初期プロンプト P_0 における順位付けで正例より上位に位置している負例を選別し，報酬計算に用いる．

$$\mathcal{C}(q) = \left\{ (d^+, \mathcal{N}(q, d^+)) \;\middle|\; d^+ \in \hat{D}^+(q),\ \mathcal{N}(q, d^+) \neq \emptyset \right\},$$
$$\mathcal{N}(q, d^+) = \left\{ d^- \in \hat{D}^-(q) \;\middle|\; s(q, d^-; P_0) > s(q, d^+; P_0) \right\}. \tag{3}$$

ただし，$s(q, d; P_0)$ は，初期プロンプト P_0 を用いて文書拡張した際の検索スコアである．

4.2.3 正例の順位に基づく重み

検索結果 $D_k(q)$ における正例文書 d^+ の順位を $r(d^+)$ とする．ただし，$r(d^+) > k$ の際は $r(d^+) = k+1$ とする．初期プロンプト P_0 における順位が下位の正例は改善余地が大きいと考えられるため，順位に基づく重みを導入する．$\hat{D}^+(q) \cup \hat{D}^-(q)$ に含まれる文書の順位の最小値と最大値 $r_{\min}, r_{\max}$ を用いて，正規化した重みを

$$w(d^+) = 10 \cdot \frac{r(d^+) - r_{\min}}{r_{\max} - r_{\min}} \tag{4}$$

と定義する．この重み付けにより，下位に位置する正例の寄与を相対的に大きくすることができる．

4.2.4 ペアワイズ比較に基づく報酬

文書集合 $\mathcal{S}(q, d^+) = \{d^+\} \cup \mathcal{N}(q, d^+)$ 内での文書の相対的な位置を評価するため，温度パラメータ τ を用いた InfoNCE [28] 形式の正規化スコアを用いる．具体的には，$d \in \mathcal{S}(q, d^+)$ に対して以下のように定義する．

$$\tilde{s}_{d^+}(q, d; P) = \frac{\exp\left(s(q, d; P)/\tau\right)}{\sum_{d' \in \mathcal{S}(q, d^+)} \exp\left(s(q, d'; P)/\tau\right)} \tag{5}$$

これを用いて，プロンプト P に対するクエリ単位の報酬を，$\mathcal{C}(q)$ に含まれるデータから構築した正例と負例のペア (d^+, d^-) に対する正規化スコアの差の重み付き平均として

$$\Delta(q, d^+; P) = \sum_{d^- \in \mathcal{N}} \left[\tilde{s}_{d^+}(q, d^+; P) - \tilde{s}_{d^+}(q, d^-; P) \right],$$
$$r(q; P) = \frac{\sum_{(d^+, \mathcal{N}) \in \mathcal{C}(q)} w(d^+)\, \Delta(q, d^+; P)}{Z(q)}. \tag{6}$$

のように定義する．ただし，$Z(q) = \sum_{(d^+, \mathcal{N}) \in \mathcal{C}(q)} |\mathcal{N}| w(d^+)$ である．式 6 は，候補プロンプトを用いて拡張した正例のスコアが大きくなるほど，負例のスコアが小さくなるほど大きな値を取る．また，クエリごとに選別したデータ集合 $\mathcal{C}(q)$ に対する計算のみで求められ，コーパス全体の再拡張・再索引を要しない．これにより，評価に要する計算量を大幅に削減でき，文書拡張プロンプトの反復的な更新を実現することが可能となる．

表 7　R@1000 における文書拡張プロンプト最適化の実験結果

	日本法			EU 法		
Model	拡張なし	最適化前	最適化後	拡張なし	最適化前	最適化後
mContriever	0.625	0.661	**0.679**	0.368	0.418	**0.455**
BM25	0.634	**0.650**	0.649	0.478	0.507	**0.508**
mDPR	0.610	0.631	**0.636**	0.320	**0.356**	0.354
mE5-small	0.631	0.654	**0.656**	0.460	**0.463**	0.458
mE5-base	0.615	0.636	**0.649**	0.464	**0.488**	0.486
mE5-large	0.638	**0.646**	0.645	0.488	**0.497**	0.469

プロンプト集合 $\mathcal{P}$ を式 1 で評価することを考える．報酬算出時のコストは文書拡張と索引の再構築が大部分を占めるため，これらに要する計算量を比較する．文書 1 件の拡張，索引付けに要するコストをそれぞれ c_e，c_i とする．ナイーブな手法では，学習に用いるクエリ集合の大きさ $|\mathcal{Q}|$ によらず，プロンプトごとにコーパス C 全体の再拡張・再索引が必要であるため，計算量は $O\big(|\mathcal{P}||C|(c_e + c_i)\big)$ となる．一方提案手法では，クエリ q ごとに評価に用いる部分集合 $\hat{D}^+(q)$ および $\hat{D}^-(q)$ に対してのみ拡張と索引付けを行う．最悪計算量は，クエリごとに選ばれる文書が重複しないような場合に $O\big(|\mathcal{P}||\mathcal{Q}|(|\hat{D}^+(q)| + |\hat{D}^-(q)|)(c_e + c_i)\big)$ となる．ここで，クエリ q ごとの正例・負例の数はコーパス全体の文書数と比べて十分に小さい．すなわち，$|\hat{D}^+(q)| + |\hat{D}^-(q)| \ll |D|$ である．本実験設定では $|C|$ がおよそ 90,000 であり，仮に学習に用いるクエリ集合の大きさを 150，クエリごとの正例・負例をそれぞれ 3 件とすると，$|\mathcal{Q}|(|\hat{D}^+(q)| + |\hat{D}^-(q)|) = 150 \times 6 = 900$ である．この場合，提案手法の最悪計算量はナイーブな手法と比較しておよそ 1/100 となる．

5 実　　験

5.1 実験設定

本節では，提案手法の有効性を検証するため，構築した日本法データセットおよび EU 法データセットを用いた実験を行う．

a） 反復的なプロンプト更新手法

第 4.1.2 節で述べたプロンプト最適化における反復更新を実現する手法として，GEPA [25] を用いる．GEPA は，スカラー値の報酬に加えて自然言語によるフィードバックをプロンプト改善用の LLM が解釈し，効率的に候補プロンプトを生成する手法である．パレートフロントに基づく遺伝的アルゴリズムを採用しており，複数の評価サブセットにおいて他候補に劣らないプロンプト集合を維持することで，局所解への早期収束を抑えつつ探索を行うことが可能である．本研究では，式 (1) の最大化を目的として GEPA によりプロンプトを更新する．自然言語によるフィードバックは，3.3.1 節に示したレポジトリに掲載した．プロンプト改善用の LLM として temperature=1.0 の Qwen3-8B [29] を用い，訓練時のミニバッチ数は 3 とした．また，DSPy [30] の実装を用いて実験を行った．

b） データセット

両データセットの訓練・検証・テストへの分割は，第 3 節の

表 8　アブレーションスタディ

Method	日本法	EU 法
ペアワイズ形式	**0.679**	0.455
w/o 拡張結果に基づくデータ選別	0.671	0.451
w/o 順位に基づく重み	0.674	0.425
InfoNCE 形式	0.668	0.447
w/o 拡張結果に基づくデータ選別	0.668	**0.456**
w/o 順位に基づく重み	0.674	0.438

表 2 に示した通りである．GEPA によるプロンプト更新では，訓練データに含まれる全クエリをロールアウトに用いる．一方で，パレートフロントに基づく探索における計算効率および探索の有効性を保つため，候補プロンプトの評価と選択には，検証データから無作為に抽出した 150 件のクエリを用いる．

c） 比 較 条 件

本タスクは同時更新文書検索という新規タスクであり，同一の評価設定で直接比較可能な既存手法は確認できなかった．また，文書拡張プロンプトを最適化する手法も確認できなかった．そこで，本実験では，拡張なしの検索手法および最適化前の初期プロンプトによる文書拡張の 2 手法をベースラインとし，最適化したプロンプトによる文書拡張と比較する．

d） 文 書 拡 張

文書拡張には，GEPA のプロンプト改善用の LLM と同じく Qwen3-8B を用いる．初期プロンプト P_0 としては，3.3.4 節で用いたプロンプトを利用する．上記の設定で文書集合中の全ての文書に対して説明文を生成し，説明文と文書をこの順に結合したテキストを検索対象とする．

e） 報酬関数の実装

式 (6) の計算に用いる検索モデルとしては，日本法データセットにおいて最適化前の文書拡張結果に対する R@1000 が最も高かった mContriever を採用する．負例集合 $D_k^-(q)$ を構成するための k は $k = 1000$ に設定した．InfoNCE 形式の正規化スコアの温度には $\tau = 0.5$ を設定した．また，クエリ q ごとにランダムサンプリングにより正例・負例をそれぞれ最大 3 件抽出し，抽出した部分集合に基づいて報酬を算出した．

f） 評　　価

最終的な検索性能の評価には，テストデータに含まれる全クエリを用いる．検索モデルとして，最適化に用いた mContriever に加え，BM25，mDPR，mE5-small，mE5-base，mE5-large も用い，性能を比較する．評価指標には適合文書の網羅性を測るために R@k を用いる．

5.2 実 験 結 果

5.2.1 文書拡張プロンプト最適化の効果

表 7 に，提案手法による文書拡張プロンプト最適化の実験結果を示す．日本法データセットでは，mContriever の R@1000 が 0.661 から 0.679 に改善している．EU 法データセットでも，mContriever の R@1000 が 0.418 から 0.455 に改善している．このように，最適化後のプロンプトによる文書拡張は，報酬関数の算出に用いた mContriever の検索性能を一貫して改善していることが分かる．この結果は，提案手法により，学習データが限られた状況においても同時更新文書検索の性能を向上できる可能性を示している．

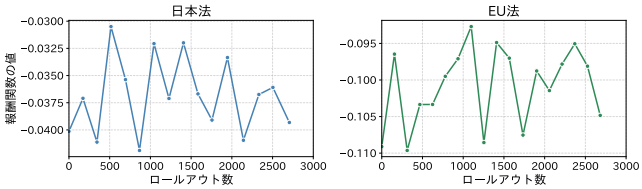

図 1　プロンプト最適化における検証データの報酬平均値の推移

一方で，その他の検索モデルでは，最適化後の性能は最適化前と比較して僅かに改善するにとどまるか，低下している．特に，EU 法データセットでは BM25 の R@1000 が 0.507 から 0.508 に僅かに向上した一方で，それ以外のモデルでは最適化前のプロンプトによる拡張の性能を下回っている．以上より，提案した報酬関数に基づく文書拡張プロンプト更新は，最適化時に用いた検索モデルに適合する形で文書拡張を更新し，その効果がモデルに依存することが示唆される．

5.2.2 アブレーションスタディ

表 4.2 節で紹介した報酬関数の形式と，拡張結果に基づくデータ選別・順位に基づく重みの要素の有効性を検証するための実験結果を示す．最適化時の設定は表 7 と同じで，mContriever で検索した際の R@1000 を示している．また，式 6 のペアワイズ形式の報酬関数には正例と負例の正規化スコアの差を用いた．この正規化スコアは最適化に用いる検索モデルが，正例と複数の負例を与えられたとき，拡張後の各文書を正例であるとみなす確率と捉えることができる．そのため，正例を入力した際の正規化スコアの値を最大化することでも最適化を進めることができると考えられる．これを InfoNCE 形式の報酬関数とし，$Z(q) = \sum_{(d^+,\mathcal{N})\in\mathcal{C}(q)} w(d^+)$ を用いて以下のように定義する．

$$r(q;P) = \frac{\sum_{(d^+,\mathcal{N})\in\mathcal{C}(q)} w(d^+)\tilde{s}_{d^+}(q,d^+;P)}{Z(q)} \tag{7}$$

表を見ると，日本法データセットにおいてはペアワイズ形式の報酬関数が最も高い性能を示し，InfoNCE 形式を一貫して上回った．一方，EU 法データセットでは，最良性能は InfoNCE 形式において拡張結果に基づくデータ選別を行わない設定で得られたものの，ペアワイズ形式との差は小さかった．このことから，ペアワイズ形式は両データセットにおいて比較的良好な性能を示したといえる．

ペアワイズ形式においては，両データセットにおいて要素を取り除くと性能が低下しており，性能に寄与していることが確認された．一方，InfoNCE 形式では要素を取り除くと性能が向上する場合も確認された．これらの結果は，データ選別や重み付けの要素の有効性が，報酬関数の形式と強く相互作用することを示している．

5.2.3 プロンプト最適化の学習過程

図 1 に，プロンプト最適化における検証データの報酬平均値の推移を示す．左が日本法，右が EU 法データセットの結果を

示している．1 ロールアウトは 1 クエリで報酬関数の値を算出することを示す．一定量の訓練データでロールアウトし候補のプロンプトを生成した後，報酬関数の期待値が高いプロンプトが選択され，全検証データを用いて報酬の平均値が算出される．図にはその報酬の平均値がプロットされている．

図 1 では，両データセットにおいて，報酬平均値はロールアウトの進行に伴い単調に推移するのではなく，探索方針の更新に応じて上下に変動している．GEPA は，検証データ上の評価に基づき探索方針を更新しながら探索を反復する．そのため，探索過程では評価値が一様に改善するとは限らず，探索方針の更新に伴い一時的に評価値が低下することもある．

また，日本法データセットでは 3 回目の評価時点で，EU 法データセットでは 7 回目の評価時点で報酬平均値が最大となり，それ以降の評価ではこれを上回るプロンプトが得られていない．この結果は，現状の報酬関数では，探索更新を継続しても追加的な改善を安定して得られない可能性を示唆する．報酬関数の設計として，サンプリング数の調整，サンプリング方法の見直し，重み付けの方法の改善などを行うことによって，より長期の探索更新により性能改善が得られる可能性がある．

一方，表 7 で mContriever の性能が最適化後に向上していることから，この報酬関数の値の増加が，報酬算出に用いた検索モデルの性能改善に結び付いていることが確認できる．

5.2.4 最適化後のプロンプト

最適化後の文書拡張プロンプトを 3.3.1 節に示したレポジトリに掲載した．

日本法データセットで最適化した文書拡張プロンプトには，条文の核心的な内容と社会的な目的を説明すべきという抽象的であった指示に対し，具体的にどのように説明すべきかを補足する項目が追加されている．また，関連条文を明記するように指示が追加されている．関連条文を明記することで，もとの条文に含まれなかった他条文の情報が拡張文に含まれる可能性が高まり，この結果として，同時に更新される条文をより広い範囲で検索しやすくなり，網羅性の改善に寄与し得る．

また，最適化後のプロンプトには「文書拡張の目的は，検索モデルに入力する際に正例が負例よりも高いスコアを獲得するようにするためです」や,「プロンプトはシンプルで明確にし，情報過多にならないように注意してください」といった指示が含まれる．これらはいずれも，プロンプト改善用の LLM に与えたテキストフィードバックに含まれる観点が，文書拡張用の LLM への指示として取り込まれた結果であると解釈できる．前者は文書拡張の目的を明示し，拡張文が検索に有用な情報を含むよう誘導する点で，文書拡張用のプロンプトに含めることが妥当である．一方で後者は，プロンプト改善過程における制約であり，文書拡張に直接寄与しない指示が混入し得ることを示している．このことから，プロンプト改善用の LLM がフィードバックを適切に解釈し，文書拡張に必要な指示へと整理できる能力が重要となることが示唆される．

EU 法データセットで最適化した文書拡張プロンプトでも，日本法データセットと同様に，条文の核心的な内容や社会的な目的を説明するという抽象的な指示に対する補足が追加されている．ただし，日本法データセットを単に英語に訳した内容ではなく，EU 法文書に特有の制度的主体と法令体系を反映した指示が含まれている．具体的には，条文が置かれた法的枠組みや対象領域を特定し，欧州委員会や理事会，欧州議会，加盟国などの主体の役割と，手続上または実体上の義務を明確に記述することが求められている．

一方で，最適化後のプロンプトは具体例として多数の法令名を提示している．このような例示は，条文の主題に即した用語を想起させる点で有用であるが，拡張対象の条文と関係のない法令名が拡張文に含まれる場合には，検索に不要な用語が増加し，正例と負例の区別に必要な手がかりが弱まるおそれがある．

6 まとめ

本研究では，ある文書の更新に関する情報をクエリとし，同時更新文書を特定する新たな検索タスクと手法を提案した．本タスクでは，与えられた更新内容に基づき，整合性を維持するために同時に更新が必要となる文書を大規模な文書集合の中から網羅的に検索することを目的とした．

タスク向けに構築したデータセットでは，日本および EU の法文書改正データから同時に更新される条文の情報を抽出し，言語や法体系の異なる 2 種類のデータを収集した．収集したデータに対し，検索タスク向けに処理を行ったところ，約 1,600 件のクエリ，約 90,000 件の文書からなるタスク向けのデータセットが 2 種類構築された．構築したデータセットを分析した結果，既存の疎検索および密検索モデルでは，更新内容と同時更新文書の間の潜在的な関係性を十分に捉えられず，検索効果が限定的であることが判明した．また，キーワードクエリを用いた手法，文書拡張手法，リランキング手法を比較した結果，文書に情報を付加する文書拡張手法が本タスクにおいて比較的有効であることが示唆された．

これらを踏まえ，本論文では文書拡張を提案手法の基盤として採用し，プロンプト最適化を用いた文書拡張手法を提案した．文書拡張プロンプトの最適化をナイーブに行うと，候補プロンプトを評価するたびに全文書の再拡張と再索引が必要となり，反復的な最適化の実行において計算コストが膨大となる課題に対し，本研究では，コーパス全体を再処理せずに計算できる報酬関数を設計し，プロンプト最適化に用いた．具体的には，代表的なデータのサンプリングにより計算量削減を行なった上，少量のデータで効果的な最適化ができるように利用するデータのさらなる選別やランクによる重み付けを行った．

提案手法の有効性を検証するため，構築した 2 種類のデータセットを用いて実験を行った．実験の結果，最適化後のプロンプトによる文書拡張は，日本法データセットおよび EU 法データセットの双方において，最適化時に用いた検索モデルの検索性能を向上させた．この結果は，提案手法により，学習データが限られた状況においても同時更新文書検索の性能を向上できる可能性を示している．

今後の課題として，第一に，報酬関数の計算に用いる代表データのサンプリング方法改善が挙げられる．本研究ではラン

ダムサンプリングを採用したが，データの難易度やクエリの多様性を考慮したサンプリング戦略を設計することで，より少ない計算量で検索性能の向上が得られる可能性がある．第二に，提案手法の堅牢さの検証である．本研究では，検索モデルにmContriever，大規模言語モデルに Qwen3-8B を用いた実験のみを行った．そのため，検索モデルおよび大規模言語モデルを変更した場合にも同様の性能向上が得られるかを検証し，提案手法の一般性を確認する必要がある．

謝　　辞

本研究は JSPS 科研費 JP23K25154 の助成を受けたものです．ここに記して謝意を表します．

参 考 文 献

[1] 株式会社 LegalOn Technologies. 企業法務の実態調査 2022 年 6 月実施. https://legalontech.jp/wp-content/uploads/2022/07/legalissues.pdf, 2022. 参照日: 2025-12-18.

[2] 独立行政法人情報処理推進機構. 2023 年度ソフトウェア開発に関するアンケート調査結果. https://www.ipa.go.jp/digital/software-survey/software-engineering/nq6ept000000476m-att/software-engineering2023-comments.pdf, January 2023. 参照日: 2025-12-18.

[3] 国立国会図書館. 会社法の施行に伴う関係法律の整備等に関する法律. https://hourei.ndl.go.jp/#/detail?lawId=0000102829, July 2005. 参照日: 2025-12-18.

[4] Jane Dwivedi-Yu, Timo Schick, Zhengbao Jiang, Maria Lomeli, Patrick Lewis, Gautier Izacard, Edouard Grave, Sebastian Riedel, and Fabio Petroni. EditEval: An instruction-based benchmark for text improvements. In *CoNLL*.

[5] Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *SemEval*, pp. 1–14, 2017.

[6] Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. JFLEG: A fluency corpus and benchmark for grammatical error correction. In *EACL*, pp. 229–234, 2017.

[7] Fabio Petroni, Samuel Broscheit, Aleksandra Piktus, Patrick Lewis, Gautier Izacard, Lucas Hosseini, Jane Dwivedi-Yu, Maria Lomeli, Timo Schick, Michele Bevilacqua, et al. Improving wikipedia verifiability with ai. *Nature Machine Intelligence*, Vol. 5, No. 10, pp. 1142–1148, 2023.

[8] Alexander Spangher, Xiang Ren, Jonathan May, and Nanyun Peng. NewsEdits: A news article revision dataset and a novel document-level reasoning challenge. In *NAACL*, pp. 127–157, 2022.

[9] Timo Schick, Jane Dwivedi-Yu, Zhengbao Jiang, Fabio Petroni, Patrick Lewis, Gautier Izacard, Qingfei You, Christoforos Nalmpantis, Edouard Grave, and Sebastian Riedel. Peer: A collaborative language model. *ICLR*, 2023.

[10] Robert Iv, Alexandre Passos, Sameer Singh, and Ming-Wei Chang. FRUIT: Faithfully reflecting updated information in text. In *NAACL*, pp. 3670–3686, 2022.

[11] Qian Ruan, Ilia Kuznetsov, and Iryna Gurevych. Re3: A holistic framework and dataset for modeling collaborative document revision. In *ACL*, pp. 4635–4655, 2024.

[12] Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. Document expansion by query prediction. *arXiv preprint arXiv:1904.08375*, 2019.

[13] Rodrigo Nogueira, Jimmy Lin, and AI Epistemic. From doc2query to doctttttquery. *Online preprint*, Vol. 6, No. 2, 2019.

[14] Mitko Gospodinov, Sean MacAvaney, and Craig Macdonald. Doc2query–: when less is more. In *European Conference on Information Retrieval*, pp. 414–422. Springer, 2023.

[15] Florian Boudin, Ygor Gallina, and Akiko Aizawa. Keyphrase generation for scientific document retrieval. In *ACL*, pp. 1118–1126, 2020.

[16] Jian Tang, Yue Wang, Kai Zheng, and Qiaozhu Mei. End-to-end learning for short text expansion. In *SIGKDD*, pp. 1105–1113, 2017.

[17] Stephen Robertson, Hugo Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, Vol. 3, No. 4, pp. 333–389, 2009.

[18] Xinyu Zhang, Xueguang Ma, Peng Shi, and Jimmy Lin. Mr. TyDi: A multi-lingual benchmark for dense retrieval. In *MRL*, pp. 127–137, 2021.

[19] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning, 2021.

[20] Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*, 2024.

[21] Peilin Yang, Hui Fang, and Jimmy Lin. Anserini: Enabling the use of lucene for information retrieval research. In *SIGIR*, p. 1253–1256, 2017.

[22] OpenAI. GPT-4.1 series, April 2025. Large language model series.

[23] 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. *TACL*, Vol. 11, pp. 1114–1131, 09 2023.

[24] Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. Is ChatGPT good at search? investigating large language models as re-ranking agents. In *EMNLP*, pp. 14918–14937, 2023.

[25] Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, Christopher Potts, Koushik Sen, Alexandros G. Dimakis, Ion Stoica, Dan Klein, Matei Zaharia, and Omar Khattab. kk: Reflective prompt evolution can outperform reinforcement learning, 2025.

[26] Krista Opsahl-Ong, Michael J Ryan, Josh Purtell, David Broman, Christopher Potts, Matei Zaharia, and Omar Khattab. Optimizing instructions and demonstrations for multi-stage language model programs. *arXiv preprint arXiv:2406.11695*, 2024.

[27] Minchan Kwon, Gaeun Kim, Jongsuk Kim, Haeil Lee, and Junmo Kim. StablePrompt : Automatic prompt tuning using reinforcement learning for large language model. In *EMNLP*, pp. 9868–9884, 2024.

[28] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[29] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

[30] Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. Dspy: Compiling declarative language model calls into self-improving pipelines. 2024.

クライアントサイド検索システムにおける最適なクエリ拡張手法推定

花岡　愛梨†　　丸田　敦貴††　　加藤　　誠†††,††††

† 筑波大学 知識情報・図書館学類　〒 305-8550 茨城県つくば市春日 1-2
†† 筑波大学大学院 人間総合科学学術院　〒 305-8550 茨城県つくば市春日 1-2
††† 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　〒 101-8430 東京都千代田区一ツ橋 2-1-2
E-mail: †{s2210080,s1711567}@klis.tsukuba.ac.jp, ††mpkato@acm.org

あらまし　近年，LLM を活用した検索アルゴリズムは高い検索性能を示している一方で，インデックスの再設計や GPU 計算基盤の整備など多大なコストを要し，特に中小規模の検索システムでは導入の障壁が高い．先行研究では，クエリ拡張は基盤となる検索モデルの性能が高い場合には却って精度を低下させる例も報告されており，モデルに依存しない一様な適用は適切でないことが示唆されている．本研究では，サーバサイドの検索アルゴリズムが不明なブラックボックス環境を想定し，クライアントサイドのみで検索性能を向上させるために，最適なクエリ拡張手法を選択する枠組みを提案する．まず，ユーザクエリの入力に先立ち，サーバサイドのシステム特性を推定するための事前クエリ群を検索システムに投入し，得られた検索結果からシステムの検索結果の傾向を推定する．その上で，これらの傾向に基づき，複数のクエリ拡張手法の中から最大の検索性能が期待される手法を多クラス分類により選択する．BEIR データセットを用いた実験により，提案手法が異なる検索モデルに対してクエリ拡張の効果を動的に判断し，検索性能を向上させうることを示した．

キーワード　情報検索，オンデバイス，クエリ拡張

1　はじめに

近年，LLM を活用した検索アルゴリズムの実サービスへの導入が進んでいる．BERT [5] のような大規模言語モデルを，クエリや文書のエンコーダとして用いることで，単語の一致関係だけでなく文脈・意味類似性を反映した密なベクトル表現が可能になった．実際，複数のオープンドメイン QA データセットにおいて，Dense Passage Retrieval（DPR）は Lucene-BM25 を上回り，上位 20 件の候補文書に少なくとも 1 件の正解文書を含む割合で 9〜19%高い性能が報告されている [10]．この結果は，従来の単語一致による検索アルゴリズムよりも密なベクトル表現による検索が高い検索有効性を示しており，適合文書をより上位に提示できるという点において，ユーザの検索体験を大きく向上させる可能性がある．

LLM を活用した検索アルゴリズムの実サービスへの導入には，既存の転置インデックスを中心とした検索基盤と比べて，文書・クエリの表現形式，候補文書の取得方式，およびインデックスが異なるため，既存基盤の再利用が難しいという課題がある．従来の検索システムは，文書をトークン列として分かち書きしたうえで転置インデックスを構築し，オンラインではクエリを同様に分かち書きをして BM25 等の語彙一致に基づくスコアリングを行う．一方，LLM を用いた Dense Retrieval では，文書集合やクエリを高次元ベクトルに変換し，近似最近傍探索（ANN）により類似文書を取得する．具体例として Dense Retrieval の代表例である DPR [10] では，全文書を事前にエンコーダでベクトル化し，Faiss などの ANN インデックスに格納したうえで，検索時にはクエリも同一エンコーダでベクトル化して近傍探索を実行する．この構成では，転置インデックスに代わって「文書ベクトルと ANN インデックス」を保持するストレージが必要となり，さらに文書・クエリ双方のベクトル化や ANN 探索を含む処理系が要求される．したがって，Dense Retrieval の導入は，インデックスとベクトル表現に伴う処理系を含む検索基盤全体の改修を伴う．さらに，Dense Retrieval で行われる文書・クエリ双方のベクトル化は行列演算を主体とするため計算負荷が高い．その結果，CPU のみで実運用水準のスループットや応答時間を安定して確保することは難しく，GPU の計算資源を前提とした運用になりやすい．このとき導入障壁となるのは，GPU の計算資源に要するインフラ費用に限られない．実運用では，モデル更新，インデックスの再構築，さらに推論基盤の監視・障害対応といった継続的な保守運用が必要となり，これらを担う機械学習・検索基盤の専門人材の確保・人件費も追加的に発生する．したがって，Dense Retrieval の導入は計算資源費と運用人件費の両面でコストを押し上げやすく，結果として検索基盤に十分な投資が可能な大規模サービスと，投資が難しい中小規模サービスとの間で検索品質の格差が拡大し得る．このような背景から，中小規模サービスにおいてもサーバサイドの検索基盤を大幅に改修せずに検索性能を改善する方策が求められる．その候補の一つがクエリ拡張であり，ユーザクエリの表現を補うことで，サーバサイドの検索アルゴリズムを変更しないまま検索結果の改善を狙うことができる．ただし，Weller らは，基盤となる検索モデルの性能が高い場合には，クエリ拡張の改善効果が小さくなる，あるいは性能を低下させる場合があることを報告している [17].

したがって，本研究では，サーバサイドの検索アルゴリズムが改変できない状況において，特定の拡張手法を一律に適用するのではなく，サーバサイドに適応的にクエリ拡張手法を選択するクライアントサイドクエリ拡張の実現を目的とする．

本手法は，(1) 事前調査フェーズと (2) ユーザ検索時の補助フェーズの二段階で構成される．事前調査フェーズでは，あらかじめ用意した事前クエリ集合をサーバサイドの検索システムに送信し，得られた検索結果を解析することで，当該検索システムの挙動を捉え，プロファイリングする．このプロファイルに基づき，拡張なしを含む複数のクエリ拡張手法の中から，当該環境で検索性能を最大化し得る手法をクエリ拡張手法決定モデルが多クラス分類として推定する．続く補助フェーズでは，ユーザが検索フォームに入力したクエリを送信直前に取得し，事前調査フェーズで選択された手法に従ってクエリ表現を変換した上で，既存の検索システムへ送信する．これにより，語彙的なずれの影響を受けやすい BM25 などの疎検索モデルを採用するシステムではクエリ拡張を積極的に適用し，一方で拡張が逆効果となり得る高性能な検索モデルを採用するシステムでは拡張を行わない，といったように，サーバサイドの検索システムに応じて拡張手法を切り替える．その結果，一律なクエリ拡張適用に伴う性能低下のリスクを抑制しつつ，検索性能の向上を図る．

本研究では，データセットと検索モデルの組合せごとに最適なクエリ拡張手法を推定可能か実験を行った．実験には BEIR ベンチマークに含まれる SciFact [14] および NFCorpus [3] を用い，検索モデルとして疎検索モデル BM25 [12] と密検索モデル BGE-base-en-v1.5 [18]，Contriever [8] を採用した．ここで本研究は,「データセット C を検索モデル m で検索する状況」を 1 つの検索環境 (m, C) と呼ぶ．本実験では，2 つのデータセットと 3 つの検索モデルの組合せにより，計 6 環境を対象とした．まず，各環境における最適手法を定めるため，データセットの訓練データのクエリ全体に対して，拡張なし・Q2D・Q2E の各手法を適用した場合の Recall@100 を比較し，性能が最大となる手法を当該環境の最適手法とした．次に，訓練データのクエリに対して得られた検索結果上位 10 件から構成した観測情報を入力とし，環境ごとの最適手法を予測する分類器を学習した．統合データを 8:2 に分割した暫定評価では分類精度が 0.726 となり，検索結果という外部観測情報のみから拡張手法を推定し得る可能性が示唆された．

本論文の貢献は，以下の通りである：

1. クライアントサイドのみで検索性能を向上させるための手法を提案し，サーバサイドの検索アルゴリズムを改変できない状況においても有効に機能する手法を示した．
2. 複数のクエリ拡張手法の中から，検索システムの特性に応じて最適な手法を選択できるかを検証し，クエリ拡張の効果を動的に判断するアプローチの有効性を示した．

本論文の構成は以下の通りである．2 章では，オンデバイス研究と，クエリ拡張に関する既存研究について述べる．3 章では，クライアントサイド検索システム，クライアントサイドクエリ拡張，最適なクエリ拡張手法の予測といった提案手法の詳細について述べる．4 章では，実験設定や実験結果について示し，5 章では今後の課題と共に本論文の結論を述べる．

2 関 連 研 究

本章では，オンデバイス研究の概要について述べ，既存研究と本研究の違いについて説明する．その後，本研究においてクライアントサイドで行う処理であるクエリ拡張技術について説明する．

2.1 オンデバイス研究の概要

Zhou らは，オンデバイス学習を「モデルの学習および推論の手続きをエッジデバイス上へ移し，他の計算機とのデータ交換を要しない形で完結させる枠組み」と説明している [20]．本研究ではこの定義を踏まえ，学習に限らず，検索や推薦などの情報アクセス処理をサーバサイドに依存せずクライアントサイドで実行する形態を総称してオンデバイス処理と呼ぶ．従来，推薦や検索など多くの情報アクセス処理はサーバサイドのモデルで一括して実行されてきた．しかし，近年はエッジデバイスの技術開発が急速に進み，ストレージ，通信，計算能力が向上したため，ユーザーのエンドデバイスに，従来サーバサイドで行ってきた処理を移行する技術が進展している [6]．その具体例として，オンデバイス推薦システム（DeviceRSs），オンデバイス検索システム，オンデバイス RAG システムがあげられる．Yin らのサーベイでは，オンデバイス推薦システム（DeviceRSs）を, (1) 端末上でのモデル配置と推論, (2) 端末上での学習・更新, (3) セキュリティ・プライバシー，の 3 つの観点に分類している [6]．本研究で前提とするオンデバイス検索環境は，学習や更新を端末上で行うわけでも，プライバシー保護を主目的とするわけでもなく，クエリ拡張戦略の推測をクライアントサイドで実行する点で, (1) の「端末上でのモデル配置と推論」に最も近い位置づけにある．ただし，既存の DeviceRSs が推薦モデル本体をクライアントサイドに配置するのに対し，本研究では検索モデル自体はサーバサイドに残したまま，クエリ拡張戦略の選択と適用のみをクライアントサイドで行う点が異なる．

オンデバイス検索システムやオンデバイス RAG システムについても，多数提案されている．Rawassizadeh らは，モバイル端末およびウェアラブルデバイス上で完全にローカルに動作するオンデバイス検索フレームワーク ODSearch を提案している [11]．ODSearch では，ストレージ容量や計算資源がサーバに比べて制約されるため，検索対象と処理量をクライアントサイドで極力切り詰める必要がある．このため，圧縮によって索引や検索対象データのサイズを縮小するとともに，Bloom filter を用いて語の出現有無をを高速に判定できるようにする．また，該当しない候補を早期に除外して検索範囲を縮小することで，不要な文書参照や読み出しを削減している．これにより，ネットワークに依存せずに自然言語検索を実現している．さらに，Wang らは，Web ブラウザ内で完結して動作し，クライアントサイド上のデータベースのベクトル検索によってサーバ不要の RAG 型テキスト生成を可能にする MeMemo を提案して

いる [16]．これらはいずれも，インデックス構築からランキング処理までの検索処理をクライアントサイドで完結させることを主眼とした研究・実装である．一方，本研究は検索インデックスおよびランキング処理自体はサーバサイドの検索システムに委ね，クエリ拡張戦略の選択と適用のみをクライアントサイドで行う点が既存研究と異なる．

2.2 クエリ拡張の概要

クエリ拡張 [2], [4], [19] とは，ユーザが最初に入力した検索クエリに対して，関連する語を追加したり，元クエリを書き換えたり，各語の重み付けを調整する手法である．自然言語では，同じ内容が異なる表現や類義語によって記述されることが多く，ユーザの検索クエリと文書中で実際に用いられている語の間に語彙的なずれが生じやすい．よって，この語彙的なずれを緩和することで，サーバサイドの検索アルゴリズムを変更せずに再現率やランキング精度の向上を図ることが可能になる．

近年は，大規模言語モデル（LLM）にユーザの元クエリを与えて拡張情報そのものを生成させるクエリ拡張手法が提案されている [1], [9], [15]．その代表例が Query2Doc（Q2D）[15] および Query2Expansion（Q2E）[9] である．Q2D [15] は，Wang らによって提案された手法であり，LLM に対して「与えられたクエリに答える文書（passage）を書け」というプロンプトを与え，生成された擬似文書をユーザの元クエリに連結して新しいクエリとして用いる手法である．生成される擬似文書は，クエリに対する背景知識や言い換え表現を多く含むため，BM25 などの疎検索モデルに対してはコーパス側の語彙とマッチしやすくなり，密検索モデルに対しても効果が確認されている．拡張に使用するテキストが LLM から直接生成されるテキストであるため，First-stage Retrieval の上位検索結果の品質が十分でない状況においても，クエリ拡張の基盤となる情報がそれによって劣化しにくいという点が利点である．一方，Q2E [9] は，Jagerman らによって提案されたクエリ拡張手法の一種であり，Q2D が「文書（passege）」を生成するのに対し，クエリに関連したキーワードリスト（a list of keywords）を直接生成させる手法である． 具体的には，ユーザの元クエリに対して同様の拡張語を出力させ，それらを元クエリに連結し再検索する．これらの手法は，疑似関連性フィードバックが抱える「拡張語は，First-stage Retrieval の上位の検索結果の品質に強く依存してしまう」「ノイズ語によって本来意図した情報要求から変更後のクエリが乖離してしまう」といった問題に対して，LLM が持つ事前知識を利用し元クエリに不足している情報や関連語を補うことで，この問題に対処しようとするものである．

もっとも，クエリ拡張は必ずしも普遍的に有効であるとは限らない．Weller らは，各種検索モデルにおいて，モデルの基礎性能が高くなるほどクエリ拡張による性能向上が小さくなり，場合によっては逆効果となることを報告している [17]．例えば，TREC Deep Learning Track 2019 において，First-stage Retrieval のモデルである DPR は Q2E を用いることで，ベースラインの nDCG@10（38.4%）が 6.6%改善した一方で，Reranker である LLaMA は，同様の拡張によりベースラインの nDCG@10（72.6%）が 2.9%低下している．この先行研究の結果から，クエリ拡張の有効性は検索モデルの性能水準に強く依存することが示唆される．そこで本研究では，元のユーザクエリに対してクエリ拡張を適用するか否か，またどのクエリ拡張手法を用いるかといったクエリ拡張戦略を，検索システムごとに適応的に切り替える手法に主眼を置く．

3 提 案 手 法

本章では，まず本研究が目指すクライアントサイド検索システムの全体像について述べる．次に，その中核となるアプローチであるクライアントサイドクエリ拡張の概要を説明する．最後に，提案手法である「最適なクエリ拡張手法の予測」のための問題設定と，それを実現する予測モデルおよび学習方法について詳述する．

3.1 クライアントサイド検索システム

本研究がが対象とする状況は，検索システムの検索アルゴリズムやインデックス等の内部実装が非公開で改変不可な，いわゆるブラックボックスなサーバサイドの検索システムを前提とする状況である．この時，クライアントサイドから制御可能なのは，検索システムに入力するユーザクエリと，それに対する検索結果のみである．以降，本章ではこのようなシステムを「**クライアントサイド検索システム**」と呼ぶ．本研究が目指すクライアントサイド検索システムは，ユーザクエリの送信から検索結果の提示に至る処理過程に介在し，(1) 事前調査フェーズと，(2) ユーザ検索時の補助フェーズの 2 段階で動作する構成をとる．

まず，**事前調査フェーズ**では，ユーザによる検索に先立ち，あらかじめ用意した事前クエリ集合をブラックボックスなサーバサイドの検索システムに送信し，得られた検索結果を解析する．この解析結果に基づいて，当該検索システムがクエリに対してどのように応答するかといった傾向を推定し，プロファイリングする．本プロファイルを入力として，後続のユーザ検索時の補助フェーズでどの検索補助処理を適用するか支援処理の適用方針を決定する．決定した適用方針は，以降のユーザ検索時の補助フェーズ中，クライアントサイドで保持される．

次に，**ユーザ検索時の補助フェーズ**では，ユーザの検索操作に対してクライアントサイドのみで完結する検索補助処理を挿入する．補助処理は，クエリ送信前のクエリ拡張のような入力時の処理や，検索結果に対するリランキングのような出力時の処理を含む．本フェーズでは，事前調査フェーズで決定した適用方針に従い，対象となる補助処理を必要なタイミングで実行する．検索処理自体は従来どおりサーバサイドで実行される．なお，本研究ではこのシステムのうち入力時の処理としてのクライアントサイドクエリ拡張を対象とし，次節で具体的に述べる．

3.2 クライアントサイドクエリ拡張

クライアントサイド検索システムの中核として，本研究ではクライアントサイドクエリ拡張を扱う．ここでの要点は，「拡張

を常に適用する」のではなく，サーバサイドの検索システムの性質に応じて拡張なしを含む拡張手法を選択する点にある．この選択を行うため，本研究では複数の拡張手法の中から最適手法を推定するクエリ拡張手法決定モデルを導入する．

事前調査フェーズでは，推定したプロファイルに基づき，サーバサイドの検索システムにおいて検索性能を最大化し得るクエリ拡張手法をクエリ拡張手法決定モデルが多クラス分類の問題として推測する．

ユーザ検索時の補助フェーズでは，ユーザが検索フォームにクエリを入力し送信する直前で，クライアントサイドのクエリ拡張モデルが，選択された手法に従ってクエリ表現に変換し，検索システムに送信する．

実装形態としては，Web ブラウザの拡張機能としてクエリ拡張を実装することが考えられる．具体的には，検索画面上で動作する拡張機能のスクリプトが，ユーザクエリをフォーム送信の直前に取得し，選択された手法でクエリを変換した上で既存の検索システムに送信する形を想定している．

以上の設計により，語彙的なずれで検索性能が大きく変わるようなクエリ拡張が有効に働きやすい検索システムに対しては積極的にクエリ拡張を適用し，ベースライン性能が高くクエリ拡張が逆効果となりやすい検索システムに対しては拡張なしを選択する，という形で，検索システムごとの特性に適合したクエリ表現に変換し，全体としてより良い検索性能を狙う．

3.3 最適なクエリ拡張手法の予測

3.3.1 問 題 設 定

本研究では，文書コレクションと検索モデルの組を 1 つの「検索環境」とみなし，その検索環境に対してどのクエリ拡張手法を採用すべきかを推定する問題を扱う．ここではまず，入力と出力を数学的に定義し，予測問題を定式化する．

はじめに，本研究で扱う検索環境を定義する．文書コレクション を C，そのコレクション上で評価に用いる検索クエリの集合を Q_C とする．各クエリには，どの文書が適合文書であるかという適合判定ラベルが与えられている．また，検索モデルの候補集合を M とし，その要素 $m \in M$ は BM25 や BGE, Contriever などの個別の検索モデルを表す．さらに，クエリ拡張手法の候補集合を E とし，その要素 $e \in E$ は，クエリ $q \in Q_C$ を拡張クエリ $e(q)$ に変換する「拡張なし」を含んだ手法を表す．クエリ $q \in Q_C$ と検索モデル $m \in M$ が与えられたとき，コレクション C 上で得られるランキング結果を $D_{q,m,C}$ と表記する．同様に，拡張クエリ $e(q)$ に対する検索結果を $D_{e(q),m,C}$ と表す．この時の，検索性能の指標は $\mathrm{Eval}(D_{e(q),m,C})$ と表す．検索環境 (m, c) において，クエリ集合 Q_C 上の平均検索性能を最大にするクエリ拡張手法を，当該検索環境の最適クエリ拡張手法 $e^*_{m,C}$ と定義する．すなわち，

$$e^*_{m,C} = \arg\max_{e \in E} \frac{1}{|Q_C|} \sum_{q \in Q_C} \mathrm{Eval}\big(q, D_{e(q),m,C}\big)$$

とおく．このとき $e^*_{m,C}$ は，個々のクエリ毎に異なる最適手法を選ぶのではなく，検索環境ごとに一意に定まる「環境単位」の最適手法を表す．

一方，本研究の前提では，サーバサイドの検索アルゴリズムは内部のモデル m やインデックスには直接アクセスできないブラックボックス設定である．クライアントサイドが利用可能な情報は，事前クエリを入力した際に取得可能な検索結果のみである．そこで，事前調査フェーズに用いるクエリ集合を Q_C^{probe} とし，事前クエリ $q \in Q_C^{\mathrm{probe}}$ を入力して観測される情報を $x_{q,m,C}$ と表す．なお，$x_{q,m,C}$ の具体的な構成は次節で定義する．検索環境 (m, C) に対して得られる観測全体を

$$X_{m,C} = \{x_{q,m,C} \mid q \in Q_C^{\mathrm{probe}}\}$$

とおく．本研究の目的は，観測情報 $X_{m,C}$ のみに基づいて，検索環境の最適手法 $e^*_{m,C}$ を予測することである．次節では，この予測を行うモデルを定義する．

3.3.2 予測モデル

本節では，前節で定義した観測情報 $X_{m,C}$ を入力として，検索環境 (m, C) に対するクエリ拡張手法を出力する予測モデルを定義する．出力は，検索環境 (m, C) に対して一つに定まる予測手法 $\hat{e}_{m,C} \in E$ である．以下では，$X_{m,C}$ を構成する各要素 $x_{q,m,C}$ の具体的な表現と，それに基づく予測モデルの計算手順を述べる．

まず，事前クエリ $q \in Q_C^{\mathrm{probe}}$ に対して観測される情報 $x_{q,m,C}$ を次式のように構成する．

$$x_{q,m,C} = q \oplus \bigoplus_{d \in D_{q,m,C}} \big(d_{\mathrm{rank}} \oplus d_{\mathrm{title}} \oplus d_{\mathrm{text}}\big)$$

ここで，d_{rank} は文書 d の順位，d_{title} はタイトル，d_{text} は文書のテキスト内容を表す．また，$\oplus$ はこれらの情報をテキストとして連結する操作を表す．さらに，検索結果リストを順位順に $_{q,m,C} = (d_1, \ldots, d_K)$ と書くと，上式右辺の $\bigoplus$ は次の展開で表される：

$$\bigoplus_{d \in D_{q,m,C}} (d_{\mathrm{rank}} \oplus d_{\mathrm{title}} \oplus d_{\mathrm{text}}) = (d_{1,\mathrm{rank}} \oplus d_{1,\mathrm{title}} \oplus d_{1,\mathrm{text}}) \oplus \cdots \oplus (d_{K,\mathrm{rank}} \oplus d_{K,\mathrm{title}} \oplus d_{K,\mathrm{text}})$$

したがって，$x_{q,m,C}$ は「クエリ q」と「検索結果上位 K 件の各文書の $(\mathrm{rank}, \mathrm{title}, \mathrm{text})$」を順位順に連結した観測表現である．以上のように，各 $q \in Q_C^{\mathrm{probe}}$ に対して $x_{q,m,C}$ を構成する．これらを前節で定義した観測情報 $X_{m,C}$ としてまとめ，これをモデル入力として用いる．

しかし，本研究で推定したいのはクエリ単位の手法ではなく，検索環境 (m, C) ごとに一意に定まる手法である．そこで，事前クエリ集合 Q_C^{probe} に含まれる各 q について得られる分類確率を平均し，環境単位の予測手法 $\hat{e}_{m,C}$ を

$$\hat{e}_{m,C} = \arg\max_{e \in E} p(e \mid x_{q,m,C})$$

により定める．

3.3.3 学　　習

本節では，3.3.2 で定義したクエリ拡張手法決定モデルの学習手続きを述べる．学習では，各検索環境 (m, C) に対して環境単位の最適手法 $e^*_{m,C}$ を事前に決定し，その環境から得られる観測情報 $X_{m,C}$ に対する教師ラベルとして用いる．

学習データは，この観測情報 $X_{m,C}$ と教師ラベル $e^*_{m,C}$ の組 $(X_{m,C}, e^*_{m,C})$ を検索環境ごとに作成し，複数の環境について収集したものから構成される．以上の学習データに対して，多クラス分類損失を最小化することでモデルパラメータを更新する．このように，本研究の学習手続きは,「環境ごとに定まる最適手法 $e^*_{m,C}$ を教師ラベルとして付与し，観測情報 $X_{m,C}$ から，当該環境で有利な拡張手法を推定できるように分類器を訓練する手続き」として整理できる．

4 実　　験

本章では，前章で述べた提案手法（検索環境 (m, C) ごとに最適なクエリ拡張手法を推定する問題）について，実験によりその挙動を検証する．4.1 節では，各検索環境に対する「最適手法」$e^*_{m,C}$ の決定手順を定義し，その際に用いる評価指標を述べる．4.2 節では，実験で用いるデータセット，検索モデル，および学習設定を示す．4.3 節では，暫定的な実験結果を示し，考察と限界，今後の検証方針を整理する．なお，本章の結果はプロトタイプ段階の実験に基づく暫定値であり，評価分割の設計などについては今後の再実験により精査する必要がある．

4.1 データセット

実験には BEIR ベンチマークに含まれる SciFact [14] 及び NFCorpus [3] を用いた．scifact [14] は，全 1,409 件の主張と 5,183 件の抄録から構成され，科学文献に基づく主張の真偽判定を目的として構築されたデータセットである．具体的には，科学的な主張に対して，研究論文の抄録がその主張を支持するのか，反論するのか，あるいは関連情報が存在しないのかを判定するタスクを想定して設計されている．各主張には，関連する複数の抄録が対応づけられており，抄録ごとにラベルが付与されている．NFCorpus [3] は，全 3,244 件のクエリと 3,633 件の文書からなる，医療情報検索の改善を目的として構築されたデータセットである．健康情報サイト NutritionFacts.org における，一般利用者が書いた質問文と，それに関連付けられた科学研究論文のリンク構造を収集することで，平易な言語と医学専門用語のあいだに存在する語彙的ギャップを橋渡しするデータを提供する．医療領域に特化した検索モデルの学習や評価に利用することが可能である．本研究では，異なる 2 つのコレクションを用いることで，コレクションが変化した際に最適な拡張手法も変化し得るかを検証対象に含める．

4.2 検索モデル（検索環境）

検索環境 (m, C) は，文書コレクション C と検索モデル m の組として定義される．本研究では，疎検索モデルとして BM25 [12]，密検索モデルとして BGE-base-en-v1.5 [18] および Contriever [8] を用いた．

BM25 [12] は単語出現に基づいてクエリと文書の適合度を計算する疎検索モデルであり，単語頻度と逆文書頻度に基づくスコアに加えて，文書長の補正や頻出語の寄与を抑制する重み付けを含む．Web 検索システムを含む多くの実システムで事実上の標準的ベースラインとして用いられている．本モデルは，語彙ギャップに敏感な検索モデルの代表として用いる．

BGE-base-en-v1.5[1] [18] は英語テキストの意味表現を得るための BERT 系埋め込みモデルである．BGE シリーズの中で本モデルは，109M パラメータの BERT-base 規模の Transformer エンコーダを用い，入力文を 768 次元の密ベクトルに写像する．開発元によれば，これらのモデルは RetroMAE による事前学習の後，大規模なテキストペアに対するコントラスト学習によって学習されておいる．BGE シリーズは，文書検索やセマンティック検索といった情報検索タスク向けの埋め込みモデルとして設計されており，MTEB および BEIR を含むベンチマークにおいて，同程度のモデル規模の既存埋め込みモデルと比較して高い性能を示すことが報告されている．さらに 1.5 系列では，類似度スコアの分布の偏りを緩和し，ユーザの質問文をそのまま入力しても高い検索性能が出るように調整されている．

Contriever [8] は，教師なし密検索モデルであり，コントラスト学習に基づいて，文書集合から意味的に類似したテキスト同士を近く，異なるテキスト同士を離すような埋め込み空間を学習するモデルである．事前学習段階では明示的な適合性ラベルを用いずに，テキストの連続性などから自動的に生成した正例・負例ペアを用いたコントラスト学習によって訓練される．このようにして得られた埋め込みを用いてクエリと文書のコサイン類似度を計算することで，BM25 のような単語頻度ベースの手法と比較しても競合しうる検索性能を達成している．Gautier らは，BEIR ベンチマークにおいて，教師なしの Contriever が BM25 と同等あるいはそれ以上の Recall@100 を達成すること，さらに MS MARCO などでファインチューニングすることで性能が向上することを報告している．本研究では，密検索モデル間でも最適な拡張手法が変化し得るかを検証するため，BGE と併せて Contriever を用いる．

4.3 クエリ拡張手法（Q2D / Q2E）の生成設定

Q2D [15] および Q2E [9] は，いずれもクエリ q を入力として拡張クエリ $e(q)$ を生成するクエリ拡張手法である．本実験では両手法を zero-shot で生成し，生成には Azure OpenAI[2] の GPT-4o [7] を用いた．

4.4 最適なクエリ拡張手法の決定

本研究では，検索環境 (m, C) ごとに候補手法集合 $E = \{$ 拡張なし, Q2D, Q2E$\}$ の中から，平均検索性能を最大化する手法を最適手法 $e^*_{m,C}$ として定め，これを後述する学習で用いる教師ラベルとする．クライアントサイド検索システムは，後段処理として検索結果のリランキングを想定しているため，リランキングの前提条件である「First-stage Retrieval の結果に適合文書が十分に含まれていること」を重視する．この観点から，本節の平均検索性能には，First-stage Retrieval における適合文書の取りこぼしの少なさを直接評価できる Recall@100 を用いる．Recall は，適合文書集合を Rel，検索結果の文書集合を Res とすると，

1：https://huggingface.co/BAAI/bge-base-en-v1.5
2：https://azure.microsoft.com

表 1　各検索環境 (m, C) に対する 3 手法 $E = \{$ 拡張なし, Q2D, Q2E$\}$ の Recall@100 のスコア.

Dataset	Model	No expansion	Q2D	Q2E
SciFact	BM25	0.9289	**0.9548**	0.9470
	BGE	0.9724	**0.9784**	0.9736
	Contriever	0.9370	**0.9551**	0.9466
NFCorpus	BM25	0.2462	0.3157	**0.3264**
	BGE	0.3484	**0.3690**	0.3621
	Contriever	0.3130	0.3414	**0.3458**

$$\text{Recall} = \frac{|\text{Rel} \cap \text{Res}|}{|\text{Rel}|}$$

で定義され，適合文書のうち検索結果に含まれた割合を表す.（なお，本研究ではリランキング処理自体の実装・評価は含まない）

表 1 は，BM25・BGE-base-en-v1.5・Contriever と SciFact・NFCorpus の組で定義される各検索環境 (m, C) に対し，候補手法 $E =$ 拡張なし, Q2D, Q2E の Recall@100 を示す．表 1 より，最適な拡張手法 $e^*_{m,C}$ は検索環境 (m, C) に依存して異なることが分かる．これは，検索モデル m やコレクション C の違いにより，クエリと文書の表現差が検索結果に及ぼす影響が変化し，より有利な拡張手法が検索環境ごとに変わると解釈できる．

4.5 学習設定

本研究では，検索環境 (m, C) に対する最適クエリ拡張手法 $e^*(m, C)$ を推定するため，多クラス分類器として事前学習済み言語モデル DistilBERT [13] を用いた．

学習例は，事前クエリと検索結果から構成される観測情報 $x_{q,m,C}$ と，対応する環境の教師ラベル $e^*_{m,C}$ の組として生成する．このとき，同一環境 (m, C) から得られるすべての学習例は同一の教師ラベル $e^*_{m,C}$ を共有する．

以上により得られた学習例 $(x_{q,m,C}, e^*_{m,C})$ を，複数コレクション・複数検索モデルにまたがって単一のデータセットとして統合し，学習例単位でランダムにシャッフルした．次に，データセット全体を，訓練データ 80%：検証データ 20%に分割し，訓練データのみを用いてモデルパラメータを学習した．

ただし，本分割は学習例単位のランダム分割であるため，同一検索環境 (m, C) に由来する学習例が訓練データと検証データの双方に混在し得る．そのため，得られる分類精度は「未知環境への汎化性能」ではなく,「既知環境に対する識別性能」を反映している可能性がある．したがって，本研究の目的に整合する評価としては，検索環境 (m, C) を単位として学習・評価を分離する分割が必要であり，今後は環境単位の交差検証により，未知環境に対する $\hat{e}_{m,C}$ の正解率として再評価する．

4.6 実験結果

学習した分類器を検証データ上で評価した結果，分類精度（accuracy）は 0.726 であった．比較のため，単純なベースラインを考える．3 手法 $E = \{$ 拡張なし, Q2D, Q2E$\}$ のいずれかを一様ランダムに選択する場合，正解する確率は各例で 1/3 であるため，期待精度は約 0.333 となる．また，本実験では環境数が 6 と少なく，教師ラベルは Q2D が 4 環境，Q2E が 2 環境，拡張なしが 0 環境であったため，最頻値ラベル（Q2D）を常に選択するベースラインの精度は約 0.667 となる．本手法の accuracy である 0.726 は，これらのベースラインを上回っており，検索結果という外部観測情報のみに基づいて拡張手法を推定する試みが一定程度の識別性能を持ち得ることが示唆される．

一方で，この値のみから提案手法の有効性を結論づけることはできない．第一に，検索環境数が 6 と限られており，教師ラベルの分布に偏りがあるため，accuracy は少数派クラスの誤分類を十分に反映しない可能性がある．この点を確認するため，今後は混同行列を併記し，どの手法がどの手法に誤分類されやすいかを分析する．併せて，クラス不均衡の影響を受けにくい指標として macro-F1 を算出し，手法ごとの識別性能を評価する．

第二に，4.2.4 節で述べた通り，本実験では学習例単位の分割を採用しており，同一環境由来の学習例が訓練データと検証データに混在し得る．したがって，本結果は「未知環境に対する環境単位の予測性能」を直接示すものではない．今後は検索環境 (m, C) を単位とした環境数に基づく 6-fold 交差検証により，環境ごとの $\hat{e}_{m,C}$ を評価し直す必要がある．

さらに，事前クエリ集合の設計が予測性能に与える影響を調べるため，事前クエリごとに予測の正誤を集計し,「複数環境で一貫して正しく予測できるクエリ」などを抽出して定性的に分析する．これにより，事前クエリ集合の設計指針を得ることが期待される．

5 結　　論

近年，LLM に基づく Dense Retrieval は高い検索性能が報告されている一方，ベクトル化や ANN 探索など従来の転置インデックスとは異なる処理基盤を要し，導入・運用コストが大きい．その結果，検索基盤に投資可能な大規模サービスと投資が難しい中小規模サービスの間で検索品質の格差が拡大し得る．

本研究では，サーバサイドの検索アルゴリズムを改変できない状況を想定し，クライアントサイドのみで検索性能を改善するシステムとして，検索環境に応じて最適なクエリ拡張手法を選択するクライアントサイドクエリ拡張を提案した．提案法は,（1）事前クエリ群の検索結果からサーバサイドの挙動をプロファイルし,（2）その情報に基づき，拡張なしを含むクエリ拡張手法から最適手法を推定してユーザクエリを送信直前に変換する二段階で構成される．

評価として，BEIR の SciFact および NFCorpus と BM25・BGE-base-en-v1.5・Contriever の組からなる計 6 環境を対象に，訓練データ上の Recall@100 により環境ごとの最適手法を定義し，検索結果上位 10 件から最適手法を予測する分類器を学習した．暫定評価では分類精度 0.726 を得て，外部観測情報のみから環境に適応的な拡張手法を推定し得る可能性が示唆された．

一方で，本結果のみから有効性を結論づけることはできない．今後は，(i) 環境数が少なくラベル分布が偏っているため，混同行列や macro-F1 を併記して識別傾向を検証すること，(ii) 同一環境由来の観測が学習・検証に混在し得るため，環境単位の交差検証により未知環境への汎化性能を評価すること，(iii) 事前クエリ集合の設計が予測性能に与える影響を調べるため，事前クエリごとに予測の正誤を集計し，「複数環境で一貫して正しく予測できるクエリ」などを抽出して定性的に分析することで，事前クエリ集合の設計指針を得ることが期待される．

謝　　辞

本研究は JSPS 科研費 JP25K03229，JP23K28090，JP24K03048，日本財団 HUMAI プログラム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] Michael Antonios Kruse Ayoub, Zhan Su, and Qiuchi Li. A case study of enhancing sparse retrieval using llms. In *Companion Proceedings of the ACM Web Conference 2024*, pp. 1609–1615, 2024.

[2] Hiteshwar Kumar Azad and Akshay Deepak. Query expansion techniques for information retrieval: a survey. *Information Processing & Management*, Vol. 56, No. 5, pp. 1698–1735, 2019.

[3] Vera Boteva, Demian Gholipour, Artem Sokolov, and Stefan Riezler. A full-text learning to rank dataset for medical information retrieval. In *European Conference on Information Retrieval*, pp. 716–722. Springer, 2016.

[4] Claudio Carpineto and Giovanni Romano. A survey of automatic query expansion in information retrieval. *Acm Computing Surveys (CSUR)*, Vol. 44, No. 1, pp. 1–50, 2012.

[5] 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: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

[6] Jialiang Han, Yun Ma, Qiaozhu Mei, and Xuanzhe Liu. Deeprec: On-device deep learning for privacy-preserving sequential recommendation in mobile commerce. In *Proceedings of the Web Conference 2021*, pp. 900–911, 2021.

[7] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

[8] 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.

[9] Rolf Jagerman, Honglei Zhuang, Zhen Qin, Xuanhui Wang, and Michael Bendersky. Query expansion by prompting large language models. *arXiv preprint arXiv:2305.03653*, 2023.

[10] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wentau Yih. Dense passage retrieval for open-domain question answering. In *EMNLP (1)*, pp. 6769–6781, 2020.

[11] Reza Rawassizadeh and Yi Rong. Odsearch: Fast and resource efficient on-device natural language search for fitness trackers' data. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, Vol. 6, No. 4, pp. 1–25, 2023.

[12] Stephen Robertson, Hugo Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, Vol. 3, No. 4, pp. 333–389, 2009.

[13] 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.

[14] David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. Fact or fiction: Verifying scientific claims. *arXiv preprint arXiv:2004.14974*, 2020.

[15] Liang Wang, Nan Yang, and Furu Wei. Query2doc: Query expansion with large language models. *arXiv preprint arXiv:2303.07678*, 2023.

[16] Zijie J Wang and Duen Horng Chau. Mememo: on-device retrieval augmentation for private and personalized text generation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2765–2770, 2024.

[17] Orion Weller, Kyle Lo, David Wadden, Dawn Lawrie, Benjamin Van Durme, Arman Cohan, and Luca Soldaini. When do generative query and document expansions fail? a comprehensive study across methods, retrievers, and datasets. In *Findings of the Association for Computational Linguistics: EACL 2024*, pp. 1987–2003, 2024.

[18] Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, pp. 641–649, 2024.

[19] Jinxi Xu and W Bruce Croft. Quary expansion using local and global document analysis. In *Acm sigir forum*, Vol. 51, pp. 168–175. ACM New York, NY, USA, 2017.

[20] Qihua Zhou, et al. Towards efficient tiny machine learning systems for ubiquitous edge intelligence. 2023.

情報検索システムのための自動ドメイン適応フレームワークの検討

宮沢　純正† 　加藤　誠††,†††

† 筑波大学 知識情報・図書館学類　〒 305–8550 茨城県つくば市春日 1–2
†† 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　〒 101–8430 東京都千代田区一ツ橋 2–1–2
E-mail: †junseim@klis.tsukuba.ac.jp, ††mpkato@acm.org

あらまし　本研究では，検索システムに精通していないエンジニアでも高性能なドメイン特化型検索システムを構築可能とするフレームワーク「AutoIR」を提案する．AutoIR は，ドキュメント集合とクエリ例を入力として，LLM を活用した評価・学習用データセットの自動生成，複数候補モデルの評価に基づく最適モデルの探索，および選定モデルへのドメイン適応を一貫して実行するフレームワークである．本稿では，特にドメイン適応に最適なモデルの探索に着目し，ドメイン適応前のゼロショット性能を用いた手法や少量のデータを用いた学習を用いた手法等を検討した．結果として，少量のデータを用いた学習結果を用いた手法が有効であることが示唆された．

キーワード　情報検索, 検索モデル, ドメイン適応, モデル選択

1　はじめに

検索システムを主要な窓口とする Web サービスにおいて，検索性能の改善は収益増加およびユーザ体験向上の観点から重要である．近年，BERT などの言語モデルを用いてクエリや文書を密ベクトルに変換し，その類似度に基づいて検索する密検索モデル [7], [16] が広く用いられている．密検索モデルは語彙一致に依存する BM25 [14] 等の手法に比べて言い換えを含む検索要求に対応しやすい一方で，モデルの性能がドメインに強く依存することが指摘されている [15]．したがって，対象ドメインのデータによる追加学習によりドメイン適応を行うことが重要となる．また，複数存在する密検索モデルの中から，ドメイン適応後に高い性能を発揮するモデルを適切に選択することも重要な課題である．

現在，対象ドメインに適した検索モデルを選定するためには，候補となるすべてのモデルに対して対象ドメインのデータを用いた追加学習と評価を行う総当たり的なアプローチが一般的である．しかし，このプロセスには膨大な計算資源と時間が必要となる．この課題に対し，画像認識や自然言語処理などの分野では，事前学習済みモデルの特徴表現や出力分布の解析に基づき，転移学習後の性能を安価に推定する手法が提案されている [12], [19]．一方で，これらの研究は主に分類タスク等を対象としており，密検索モデルのドメイン適応後の性能を対象とした検証は限定的である．さらに，密検索モデルの選択に関する研究として，追加学習を行わない未適応状態で高い性能を発揮するモデルを推定する試みがなされている [9], [10]．しかしながら，追加学習によるドメイン適応を前提とした場合に，密検索モデルの性能を事前に予測できるかについては十分に検証されていない．

本研究が構想する自動ドメイン適応フレームワークは，主として「対象のドメインにおけるテストコレクションの自動構築」および「当該ドメインに最適なモデルの選択および追加学習」の 2 つのプロセスから構成される．本稿では，このうち特に計算資源の制約が課題となる後者のプロセス，すなわちモデルの最適化に焦点を当てる．多数の候補モデルすべてに対して追加学習を行うことは現実的ではないため，前段階として有望なモデルを効率的に選別する手法の確立が不可欠である．

そこで本稿では，すべての候補モデルに対して全データを用いた学習を行うことなく，追加学習後に高い性能を発揮するモデルを効率的に推定・選別するための枠組みについて検討する．具体的には，候補モデル選択の代理指標として (i) 追加学習を行わない未適応状態での評価，(ii) 学習用データの一部（$r\%$）を用いた追加学習後の評価を扱い，それぞれの推定精度を比較する．第一に，追加学習を行わない未適応状態での評価に基づき，各モデルの未適応状態での評価スコアと追加学習後の性能との間に存在する相関関係を分析することで，事前評価のみで最終性能をどの程度予測可能かを明らかにする．第二に，学習用データの一部（$r\%$）のみを用いた短期間の追加学習を実施した段階でモデルの評価を行い，その時点での性能順位が全データを用いて収束するまで学習を行った際の順位との相関を検証する．これにより，計算コストを最小限に抑えつつ，最適なモデルを高精度に選別できるかを明らかにする．

実験では，BEIR に含まれる 6 つのテストコレクションと 3 種の候補モデルを用いて評価を行い，少量データによる追加学習後の評価が未適応状態での評価より高い推定精度を示すことを確認した．

本研究の貢献は以下の通りである．

- 密検索モデルのドメイン適応後の性能を最大化する候補モデル選択問題について，問題設定を明確化した．未適応状態での評価と学習用データの一部（$r\%$）のみを用いた追加学習後の評価を代理指標とする推定手順を整理した．さらに，一致率，性能損失 $\Delta(c)$，順位相関による評価枠組みを整理した．

- BEIR の複数のテストコレクション上で，未適応状態での評価と学習用データの全てを用いて追加学習した（$r = 100$）後の性能の関係を分析した．その結果，未適応状態での評価のみでは最適候補モデルを安定して推定できない場合があることを示した．
- 学習用データの一部（r%）のみを用いた追加学習後の評価を代理指標として用い，学習用データの全てを用いて追加学習した後のモデル順位をどの程度近似できるかを検証した．本実験範囲では，学習用データの一部のみを用いた追加学習後の評価が未適応状態での評価より高い推定精度を示し，低コストなモデル選別に寄与し得ることを示唆した．

本稿の構成は以下の通りである．第 2 節では，機械学習全般および情報検索におけるモデル選択に関する関連研究について概説する．第 3 節では，提案するモデル選択フレームワークの問題設定および手法の詳細について述べる．第 4 節では，複数のテストコレクションを用いた実験設定と結果の分析を示す．最後に，第 5 節で本研究の結論と今後の展望について述べる．

2 関 連 研 究

2.1 機械学習におけるモデル選択

本節では，機械学習におけるモデル選択に関する研究を概観し，本研究との関連を整理する．

近年，汎用データで事前に学習したモデルを出発点とし，対象タスクのデータで追加学習して利用する枠組みが広く用いられている [13], [21]．事前学習データや学習目的，モデル構造の違いにより，利用可能な事前学習済みモデルは複数存在する一方で，追加学習後の性能は対象タスクとの親和性や学習方法に依存して変動する [11], [18]．したがって，新規タスクに対しては複数の事前学習済みモデルを候補として比較し，適切なものを選択する必要がある．しかし，候補モデル数が増えるほど，各候補について追加学習と評価を行う総当たりは計算資源・時間の面で高コストとなる．このため，追加学習前の情報，あるいは学習初期の情報から追加学習後の性能を推定する指標が提案されてきた．

LEEP [12] は，事前学習モデルがターゲットデータに対して出力するクラス確率と，少量のターゲットラベルから推定したラベル分布を入力とし，両者の整合性を対数尤度の期待値として定式化することで，追加学習後の精度を予測する転移適性指標である．同様に LogME [19] は，事前学習モデルで抽出したターゲットデータの特徴表現とターゲットラベルを入力とし，その表現が線形モデルでどの程度説明できるかを周辺尤度として評価することで，追加学習後の性能を予測する指標である．You らは，LogME が多様な視覚・言語モデルに対して追加学習前の評価値のみから転移後の精度を高い相関で予測できることを報告している [19]．Achille らの Task2Vec [1] は，データセットを固定次元のベクトルで表現する枠組みであり，損失に関するフィッシャー情報量にもとづくタスク埋め込みを用いて，タスクに適した特徴抽出器や事前学習モデルを選択するメタラーニング手法を提案した．Achille らは，この埋め込みとメトリック学習により，タスクに応じた最適な事前学習モデルの選択が，全モデルを訓練・評価する場合と同程度の精度で近似可能であることを示した．さらに Bolya ら [2] は，多数の事前学習モデルから最適なモデルを選択する大規模モデル選択問題を整理し，既存の転移性能指標が十分に汎用性を持たないことを指摘した上で，既存法を改良した PARC を提案し，従来法を上回る性能を報告している．You ら [20] も複数モデルをプールして転移適性でランク付けする枠組みを提案し，LogME を用いた最適モデル選択や B-Tuning による複数モデル同時利用を可能とすることを示している．また，学習の進行に伴って得られる検証性能（学習曲線）から最終性能を外挿し，性能が見込めない候補の学習を早期に打ち切る枠組みも提案されている [3], [4]．

以上の研究は，追加学習前の評価に基づいてモデルを順位付けする点で本研究の未適応状態での評価に近い．一方，対象は主に画像・言語の分類タスクであり，検索タスクへの適用例は十分に報告されていない．

そこで本研究では，未適応状態での評価と学習用データの一部（r%）のみを用いた追加学習後の評価という 2 種の代理指標を用い，計算コストと推定精度のトレードオフを踏まえつつ，学習用データの全てを用いた追加学習（$r = 100$）後の性能をどの程度予測できるかを検証する．

2.2 情報検索モデルにおけるモデル選択

本節では，情報検索分野におけるモデル選択に関する研究を概観し，本研究の位置づけを明確化する．

情報検索分野では従来より，BM25 などの語彙一致モデルが基本的な手法として用いられてきた [14]．近年は BERT などの事前学習言語モデルを用いたニューラル検索が注目されており，DPR のような双方向エンコーダ型密検索モデルは BM25 を上回る性能を示すことが報告されている [7]．また，クエリと文書のトークン間の相互作用を後段で計算する方式を用いる ColBERT [8] や，疎表現を学習する SPLADE [5] など，深層学習に基づく多様な検索モデルが提案されてきた．しかし，これらのモデルの性能は対象ドメインに強く依存し，追加学習を行わない設定ではデータセットにより優劣が大きく変動することが知られている [15]．したがって，未知なドメインで高い性能を得るには，対象ドメインのデータを用いた追加学習によるドメイン適応が重要となる．このような状況では，未知なドメインに対してどの密検索モデルを採用すべきかというモデル選択自体が重要な課題となる．情報検索分野でのモデル選択に関しては，Khramtsova ら [9], [10] が，未知な文書群に対して最適な密検索モデルを選択する問題を扱っている．これらの研究では，複数の未学習データセットに対して最適なモデルを推定するアプローチが検討された．しかし，実験により画像認識や自然言語処理分野で提案されたドメインシフト指標をそのまま応用しても，高性能モデルの選択は困難であることが示されている [9]．また，同研究は追加学習用のデータを一切用いない設定を想定しており，少量データを用いたモデルの再学習や適応後の性能の評価は考慮していない点でも本研究と異なる．すなわち，未適応状態での性能が必ずしも適応後の性能順位と一致し

ない可能性があるため，Khramtsova らの手法は追加学習によるドメイン適応を前提とするモデル選択に直接適用できない可能性がある．

以上より，情報検索分野における既存のモデル選択手法は，追加学習による性能向上を考慮しない未適応状態のモデルを前提とした設定か，あるいはモデル自体ではなくクエリの性質に着目したものが中心となっている．一方で，追加学習によるドメイン適応を前提とした密検索モデル選択において，追加学習後の性能を低コストに推定するための代理指標は十分に整理されていない．そこで本研究では，未適応状態での評価と学習用データの一部（$r\%$）のみを用いた追加学習後の評価という 2 種の代理指標を比較し，どの程度の計算コストでどの程度の推定精度が得られるかを明らかにする．次節では，本研究における問題設定と評価指標について述べる．

3 提 案 手 法

3.1 概　　要

本節では，密検索モデルの追加学習によるドメイン適応を前提としたモデル選択の問題設定を整理し，限られた計算資源の下で有望なモデルを効率的に選別するための枠組みを述べる．第 1 節で述べた通り，候補となるすべてのモデルを対象ドメインで十分に追加学習してから評価する総当たりは高コストである．そこで本研究では，(i) 追加学習を行わない未適応状態での評価，(ii) 学習用データの一部のみを用いた追加学習後の評価，のいずれかを代理指標として用い，学習用データの全てを用いた追加学習後に高い性能を示す候補モデル （真に最適なモデル）を推定する．

3.2 問 題 設 定

3.2.1 テストコレクションと評価

本研究では，検索対象文書集合，クエリ集合，および適合性判定からなるデータセットをテストコレクションと呼び，記号 c で表す．テストコレクション c は追加学習に用いる学習用データ c_{train} と，モデル選択に用いる検証用データ c_{val}，および最終評価に用いる評価用データ c_{test} に分割されているとする．候補モデルを m とし，その性能はテストコレクション上の検索評価指標により $\text{Eval}(m, c)$ と表す．また，分割データ $d \in \{c_{\text{val}}, c_{\text{test}}\}$ 上の評価値を $\text{Eval}(m, d)$ と表す．なお，本研究では評価スコアとして nDCG@10 [6] を用いた．

評価手順および実験設定の詳細は第 4 節で述べる．

3.2.2 候補モデル集合と追加学習

候補モデルの集合を M とする．各候補モデル $m \in M$ は事前学習済みの密検索モデルであり，対象テストコレクション c の学習用データ c_{train} を用いて追加学習することで，テストコレクション c の追加学習後モデル m_c を得る．また，計算コストを抑えるため，学習用データの利用率を r（%）とし，c_{train} の一部（$r\%$）のみを用いて追加学習したモデルを $m_c^{\text{train}(r\%)}$ と表す．特に，$r = 0$ のとき $m_c^{\text{train}(0\%)} = m$ であり，$r = 100$ のとき $m_c^{\text{train}(100\%)} = m_c$ である．

3.2.3 目的と真に最適な候補モデル

モデル選択の目的は，テストコレクション c に対し，学習用データの全てを用いた追加学習（$r = 100$）後の評価用データ c_{test} 上の性能 $\text{Eval}(m_c, c_{\text{test}})$ を最大化する候補モデル m を推定することである．このとき，テストコレクション c に対する真に最適な候補モデル $m^*(c)$ を次式で定義する：

$$m^*(c) = \underset{m \in M}{\text{argmax}}\, \text{Eval}(m_c, c_{\text{test}}) \tag{1}$$

式 (1) で定義される $m^*(c)$ は，全候補モデルについて学習用データの全てを用いた追加学習を行い，その後に評価用データ c_{test} 上の性能を比較することで事後的に定まる．しかし，c_{test} は最終評価に用いるため，モデル選択の段階では参照できない．さらに，すべての候補モデルに対する追加学習と評価が必要であり，計算コストの観点からも実運用では高コストである．したがって，本研究では，検証用データ c_{val} 上で計算できる代理指標に基づき最適モデルを推定し，その推定精度を実験的に評価する．

3.3 未適応状態での評価に基づくモデル推定

第 1 のアプローチは，未適応状態（追加学習なし）における検証用データ上の評価値を代理指標とみなし，真に最適な候補モデルを推定する方法である．具体的には，テストコレクション c に対し，各候補モデル $m \in M$ を検証用データ c_{val} 上で評価し，そのスコアが最大となるモデルを推定最適候補モデル $\hat{m}(c)$ と定義する：

$$\hat{m}(c) = \underset{m \in M}{\text{argmax}}\, \text{Eval}(m, c_{\text{val}}) \tag{2}$$

本手法は，候補モデルそれぞれに対する推論と評価のみで実施でき，追加学習を伴わない点で計算コストが低い一方で，未適応状態での性能順位が，追加学習後の性能順位と一致するとは限らない．したがって，第 4 節では，式 (2) により得られる $\hat{m}(c)$ がどの程度 $m^*(c)$ に一致するかを検証する．

3.4 学習用データの一部のみを用いた追加学習に基づくモデル推定

第 2 のアプローチは，学習用データの一部（$r\%$）のみを用いて各モデルを追加学習し，その時点の評価に基づき最適モデルを推定する方法である．本研究では，$r \in \{0, 10, 25, 50, 75, 90, 100\}$ を設定する．$r = 0$ は追加学習を行わない未適応状態での評価に対応する．$r > 0$ では，c_{train} の一部（$r\%$）のみを用いて追加学習したモデル $m_c^{\text{train}(r\%)}$ を作成し，検証用データ c_{val} で評価を行う．なお，$r = 0$ のとき $m_c^{\text{train}(0\%)} = m$ であるため，式 (3) により得られる $\hat{m}^{\text{lt}}(c; 0)$ は式 (2) により得られる $\hat{m}(c)$ と一致する．推定最適候補モデル $\hat{m}^{\text{lt}}(c; r)$ は次式で与えられる：

$$\hat{m}^{\text{lt}}(c; r) = \underset{m \in M}{\text{argmax}}\, \text{Eval}(m_c^{\text{train}(r\%)}, c_{\text{val}}) \tag{3}$$

以降では，式 (1) で定義した真に最適な候補モデル $m^*(c)$ を基準として，$\hat{m}^{\mathrm{lt}}(c;r)$ の推定精度を評価する．学習用データの一部のみを用いた追加学習に基づく推定（式 (3)）は，未適応状態での評価（式 (2)）より計算コストは増加するものの，各モデルの追加学習の収束特性を部分的に反映できる可能性がある．第 4 節では，各 r における順位が最終順位をどの程度近似できるかを検証する．

本研究では，未適応状態での評価が事前学習済み表現と対象テストコレクションの整合性を反映し，追加学習によるドメイン適応後の性能順位と一定の関係を持つ可能性があると仮定する．また，学習用データの一部のみを用いた追加学習後の評価は，追加学習の初期段階における最適化の進行度合いや学習の安定性を通じて，各モデルの適応の容易さを部分的に反映し得ると考える．

3.5 モデル選択品質の評価指標

本研究では，各テストコレクションにおけるモデル選択の品質を，一致率，性能損失，および順位相関により評価する．以降では，推定最適候補モデルを $\hat{m}(c)$，真に最適な候補モデルを $m^*(c)$ と表す．$\hat{m}(c)$ は，未適応状態での評価に基づく推定（式 (2)）または学習用データの一部のみを用いた追加学習に基づく推定 $\hat{m}^{\mathrm{lt}}(c;r)$（式 (3)）を表し，いずれの場合も同一の評価指標を用いる．学習用データの一部のみを用いた追加学習に基づく推定では r に応じて異なる $\hat{m}^{\mathrm{lt}}(c;r)$ が得られるため，評価指標も r ごとに算出する．

3.5.1 一 致 率

一致率は，各テストコレクションに対して推定最適候補モデル $\hat{m}(c)$ が真に最適な候補モデル $m^*(c)$ と一致した割合である．評価対象テストコレクションの集合を C とすると，一致率は次式で定義される：

$$\mathrm{Acc} = \frac{1}{|C|} \sum_{c \in C} \mathbf{1}[\hat{m}(c) = m^*(c)] \tag{4}$$

3.5.2 性 能 損 失

性能損失は，真に最適な候補モデルを選べなかった場合に生じる性能低下の大きさである．テストコレクション c に対する性能損失 $\Delta(c)$ を次式で定義する：

$$\Delta(c) = \mathrm{Eval}(m_c^*, c_{\mathrm{test}}) - \mathrm{Eval}(m_c^{\mathrm{sel}}, c_{\mathrm{test}}) \tag{5}$$

ここで m_c^* および m_c^{sel} は，それぞれ候補モデル $m^*(c)$ および $\hat{m}(c)$ をテストコレクション c で学習用データの全てを用いて追加学習した後のモデルである．

3.5.3 順 位 相 関

順位相関は，推定に用いた代理指標に基づくモデル順位と，学習用データの全てを用いて追加学習した後のモデル順位の一致度合いを表す．本研究では，検証用データ c_{val} 上のスコアに基づく順位を代理順位とし，学習用データ c_{train} 全体で追加学習した後の評価用データ c_{test} 上のスコアに基づく順位を最終順位とする．本研究では Kendall's τ を用いる．候補モデル数を $|M| = n$ とし，2 つの順位付けにおける一致ペア数を N_c，不一致ペア数を N_d とすると，

$$\tau = \frac{N_c - N_d}{\binom{n}{2}} \tag{6}$$

で定義される．τ が 1 に近いほど順位の一致度が高く，-1 に近いほど逆順であることを示す．

3.6 小 括

本節では，密検索モデルの追加学習後の性能を最大化する候補モデル選択の問題設定を整理し，未適応状態での評価と学習用データの一部のみを用いた追加学習後の評価の 2 つの代理指標に基づく推定方法を述べた．次節では，複数のテストコレクションと複数モデルを用いて，これらの推定手法がどの程度正確に真に最適な候補モデルを近似できるかを定量的に評価する．

4 実 験

4.1 実 験 概 要

本節では，第 3 節で述べたモデル選択フレームワークの有効性を，複数のテストコレクションと複数の候補モデルを用いて評価する．本研究が扱う問題は，対象テストコレクション c に対し，学習用データの全てを用いた追加学習（$r = 100$）後の評価用データ c_{test} 上で真に最適な候補モデル $m^*(c)$（式 (1)）を，総当たりの追加学習と評価を行わずに推定することである．

本節では，代理指標として次の 2 つを扱い，それぞれを実験 1 および実験 2 として検証する．

- 実験 1：追加学習を行わない未適応状態（$r = 0$）での評価（式 (2)）に基づく推定．
- 実験 2：学習用データの一部（r%）のみを用いた追加学習後の評価（式 (3)）に基づく推定．

両実験において，代理指標に基づく推定最適候補モデル $\hat{m}(c)$ を算出し，真に最適な候補モデル $m^*(c)$ との差を，一致率（式 (4)），性能損失（式 (5)），順位相関（式 (6)）により評価する．

4.2 共 通 設 定

本節では，評価対象テストコレクションとデータ分割，候補モデル，実験設計と評価手順，評価指標を述べる．

4.2.1 評価対象テストコレクションとデータ分割

本研究では，BEIR [15] に含まれるテストコレクションを用いた．本節の分析では，候補モデル 3 種すべてについて $r \in \{0, 10, 25, 50, 75, 90, 100\}$ の結果が得られた 6 つのテストコレクションを対象とする．各テストコレクションのドメイン／タスクと規模（文書数およびクエリ数）を表 1 に示す．また，各テストコレクションの公式の分割と，本研究での学習用データ c_{train}，検証用データ c_{val}，評価用データ c_{test} の構成（$r = 100$ 時）を表 2 に示す．

表 1 に示す通り，6 つのテストコレクションは金融，生医学，コミュニティ，科学分野など複数のドメインにまたがり，質問応答，情報検索，重複質問検索，ファクトチェック，引用予測といった異なるタスクを含む．文書数は `nfcorpus` の 3,633 から `quora` の 522,931 まで幅広く分布しており，クエリ数もテス

表 1　評価対象テストコレクションの概要 *

テストコレクション	ドメイン / タスク	文書数	クエリ数（公式 train/dev/test）	クエリ数（本研究 $c_{\text{train}}/c_{\text{val}}/c_{\text{test}}$）
fiqa	金融分野における質問応答	57,638	14,166 / 648 / 170	14,166 / 648 / 170
nfcorpus	生医学分野における情報検索	3,633	2,600 / 324 / 323	2,600 / 324 / 323
quora	コミュニティ分野における重複質問検索	522,931	– / 5,000 / 10,000	4,500 / 500 / 10,000
scifact	科学分野におけるファクトチェック	5,183	809 / – / 300	728 / 81 / 300
scidocs	科学分野における引用予測	25,657	– / – / 1,000	800 / 100 / 100
trec-covid	生医学分野における情報検索	171,332	– / – / 50	40 / 5 / 5

* 表中の「—」は公式分割が存在しないことを示す.「クエリ数（公式 train/dev/test）」は公式の train/dev/test 各分割に含まれるクエリ数であり,「クエリ数（本研究 $c_{\text{train}}/c_{\text{val}}/c_{\text{test}}$）」は本研究で構成した $c_{\text{train}}/c_{\text{val}}/c_{\text{test}}$ に含まれるクエリ数である.

表 2　評価対象テストコレクションとデータ分割（$r = 100$ 時）

テストコレクション	公式の分割	c_{train}	c_{val}	c_{test}
fiqa	train/dev/test	train	dev	test
nfcorpus	train/dev/test	train	dev	test
quora	dev/test	dev（90%）	dev（10%）	test
scifact	train/test	train（90%）	train（10%）	test
scidocs	test	test（80%）	test（10%）	test（10%）
trec-covid	test	test（80%）	test（10%）	test（10%）

トコレクション間で差がある．例えば，quora では質問文をクエリとして入力し，意味的に同一の質問文を関連文書として検索する一方で，scifact では主張文（claim）に対する根拠文献を検索する．このような多様な条件下で追加学習後の最適モデルが変動し得ることを踏まえ，本研究ではこれらを評価対象とした．

表 2 の通り，利用可能な公式の分割に基づき c_{train}，c_{val}，c_{test} を構成した．c_{test} には，可能な限り公式 test を用いた．公式の分割の構成に応じて，以下の方法で c_{train}，c_{val}，c_{test} を構成した．

- 公式 train と dev が存在する場合：公式 train を c_{train}，公式 dev を c_{val}，公式 test を c_{test} とした．
- 公式 train と test が存在し，dev が存在しない場合：公式 train を学習用 90%と検証用 10%に分割し，公式 test を c_{test} とした．
- 公式 dev と test が存在し，train が存在しない場合：公式 dev を学習用 90%と検証用 10%に分割し，公式 test を c_{test} とした．
- 公式 test のみが存在する場合：公式 test を学習用 80%，検証用 10%，評価用 10%に分割した．

r を指定する場合は，上記で構成した c_{train} のうち r% をランダムにサブサンプルして学習に用い，c_{val} および c_{test} は r によらず固定した．また，$r = 100$ は r を指定せずに学習を行う設定であり，表 2 に示した c_{train} 全量を用いた学習に対応する．

4.2.2　候補モデル

候補モデルの集合を M とし，事前学習手法や学習用データの異なる複数の密検索モデルから構成した．本実験では，bge，dpr，e5 を候補とした．bge は汎用的な中国語テキスト埋め込み資源を提供する枠組みとして提案された C-Pack の一部として整備された埋め込みモデル群であり，検索や分類など幅広い応用を想定している [17]．dpr はオープンドメイン質問応答における関連パッセージ検索を目的として提案された密検索モデルであり，質問とパッセージを双方向エンコーダでそれぞれベクトル化し，ベクトル類似度に基づく検索を行う [7]．e5 は弱教師あり信号を用いた対比学習により汎用的なテキスト埋め込みを学習することを目的として提案された埋め込みモデル群であり，検索・クラスタリング・分類など単一ベクトル表現を要する幅広いタスクへの適用を想定している [16].

4.2.3　実験設計と評価手順

本研究では，代理指標の算出に c_{val} のみを用い，最終的な性能比較には c_{test} のみを用いる（第 3 節）．実験 1 および実験 2 の手順は共通して次の通りである．

1. テストコレクション c ごとに，c_{train}，c_{val}，c_{test} を構成する．
2. 各候補モデル $m \in M$ について，実験 1 では未適応状態（$r = 0$）のまま c_{val} で評価し，実験 2 では学習用データ c_{train} の一部（r%）のみを用いて追加学習した後に c_{val} で評価する．
3. c_{val} 上の評価値に基づき，式 (2) または式 (3) により推定最適候補モデル $\hat{m}(c)$ を得る．
4. $\hat{m}(c)$ と $m^*(c)$ の差を，c_{test} 上のデータの全てを用いた追加学習（$r = 100$）の結果に基づいて評価し，一致率（式 (4)），性能損失（式 (5)），順位相関（式 (6)）を算出する．

ここで，$r = 100$ は c_{train} 全量を用いた追加学習に対応し，本研究で扱う設定のうち計算コストの上限である．また，$m^*(c)$ は c_{test} 上のスコアに基づいて事後的に定義される真に最適な候補モデルであり，推定過程では参照しない．

表 3　未適応状態での評価（$r = 0$）に基づくモデル選択の結果

テストコレクション	$\hat{m}(c)$	$m^*(c)$	$\Delta(c)$	τ
fiqa	bge	e5	0.0510	0.333
nfcorpus	e5	bge	0.0045	0.333
quora	bge	e5	0.0555	0.333
scidocs	bge	bge	0.0000	1.000
scifact	bge	bge	0.0000	1.000
trec-covid	e5	bge	0.1638	0.333

4.2.4　評価指標と集計方法

評価指標 Eval には nDCG@10 を用いた．モデル選択の品質は，一致率（式 (4)），性能損失（式 (5)），順位相関（式 (6)）で評価した．各指標はテストコレクションごとに算出し，必要に応じて評価対象テストコレクションの集合 C に対して平均を取って集計した．

4.3　実験 1：未適応状態での評価に基づくモデル推定

実験 1 では，追加学習を行わない未適応状態（$r = 0$）での評価を代理指標とし，式 (2) により $\hat{m}(c)$ を推定する．表 3 に，推定結果 $\hat{m}(c)$ と真に最適な候補モデル $m^*(c)$，およびモデル選択品質を示す．表中の $\Delta(c)$ は，$\hat{m}(c)$ を選択した場合に生じる最終性能（c_{test} 上の nDCG@10）の低下である．

表 3 より，6 つのテストコレクションのうち 2 つ（scidocs, scifact）では推定最適候補モデルが真に最適な候補モデルと一致した一方で，残る 4 つでは不一致であった．特に trec-covid では $\Delta(c) = 0.1638$ と大きく，未適応状態での評価に基づくモデル選択はテストコレクションによって大きな性能低下を引き起こす可能性がある．以上より，少なくとも本実験範囲では，未適応状態での評価のみでは追加学習後の性能を十分に推定できない場合があることが示される．

4.4　実験 2：学習用データの一部（r%）のみを用いた追加学習に基づくモデル推定

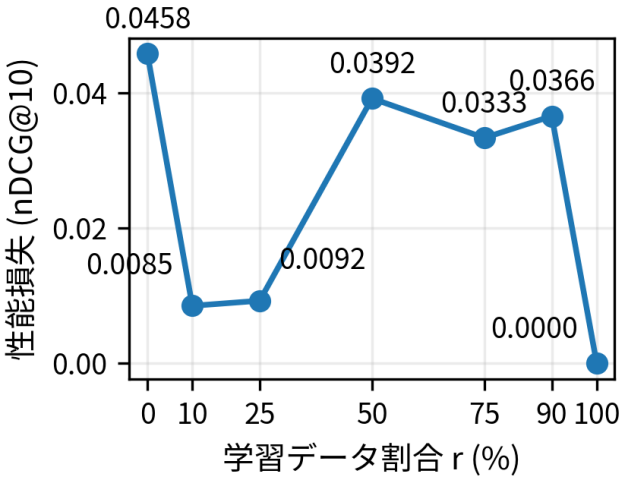

図 1　学習用データの割合 r に対する平均性能損失の推移．

実験 2 では，学習用データ c_{train} の一部（r%）のみを用いた追加学習後の評価を代理指標とし，式 (3) により $\hat{m}^{\text{lt}}(c; r)$ を推定する．表 4 に，r を変化させたときのモデル選択品質を示す．

表 4　代理指標に基づくモデル選択の結果（$r = 0$ は未適応状態での評価，$r > 0$ は学習用データの一部のみを用いた追加学習後の評価，$r = 100$ は学習用データの全てを用いた追加学習を指す）

r	一致率（%）	平均 $\Delta(c)$（nDCG@10）	平均 τ
0	33.3	0.0458	0.556
10	83.3	0.0085	0.889
25	66.7	0.0092	0.778
50	33.3	0.0392	0.556
75	66.7	0.0333	0.778
90	66.7	0.0366	0.778
100	100.0	0.0000	1.000

図 1 に，各 r における平均性能損失（平均 $\Delta(c)$）を示す．平均性能損失は $r = 10$ および $r = 25$ で小さい一方で，$r = 50$ では増大するなど，r の増加に対して単調に改善するとは限らない．

図 2 に，テストコレクションごとの性能損失 $\Delta(c)$ を示す．平均性能損失の増大は，特定のテストコレクションで大きな性能損失が生じる r が存在することに起因する．特に trec-covid では $r = 50, 75, 90$ で大きな性能損失が生じており，これが平均値を押し上げている．

4.5　計算量の比較

本節では，実験 2 の手順を $r = 10$ で実行し，その後提案手法により選択されたモデルのみを全学習データで追加学習する場合の計算量を，全候補モデルに対して追加学習を行う場合と比較する．

計算量の指標として FLOPs を用いる．FLOPs は浮動小数点演算の総回数を表す指標であり，実行時間のような実行環境に依存しやすい指標と比べて計算規模を比較しやすい尺度である．以降の単位記号は T とし，$1\,\text{T} = 10^{12}$ FLOPs とする．

候補モデル集合を M とし，候補数は $|M| = 3$ である．対象テストコレクション集合を C とし，対象数は $|C| = 6$ である．テストコレクション $c \in C$ に対し，学習用データ c_{train} の r% を用いて候補モデル $m \in M$ を追加学習する際の計算量を $\mathcal{F}(m, c, r)$ とおく．ここで $r = 100$ は学習用データの全てを用いた追加学習を指す．

総当たり法では，全候補モデルを $r = 100$ で追加学習した結果に基づいてモデルを選択するため，最終モデルを得るまでの計算量は

$$\text{Cost}_{\text{brute}} = \sum_{c \in C} \sum_{m \in M} \mathcal{F}(m, c, 100) \tag{7}$$

で与えられる．

一方，実験 2 の手順では，各候補モデルを r% のみ追加学習して c_{val} 上で評価し，式 (3) により $\hat{m}^{\text{lt}}(c; r)$ を推定する．本節で最も良い推定精度が得られた $r = 10$ を用いる場合，最終モデルを得るまでの計算量は

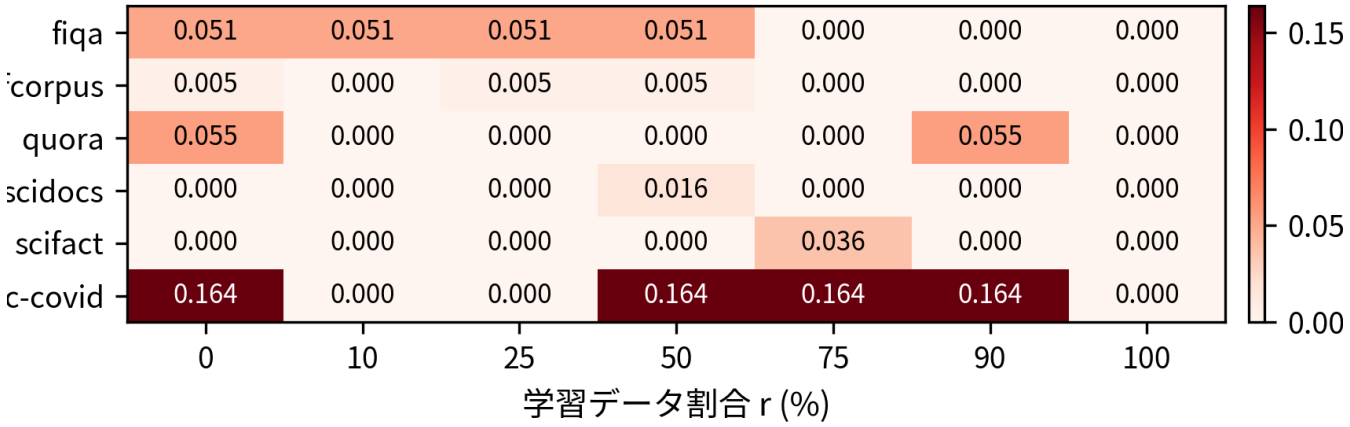

図 2　テストコレクション c と学習用データの割合 r ごとの性能損失 $\Delta(c)$（nDCG@10 の差）.

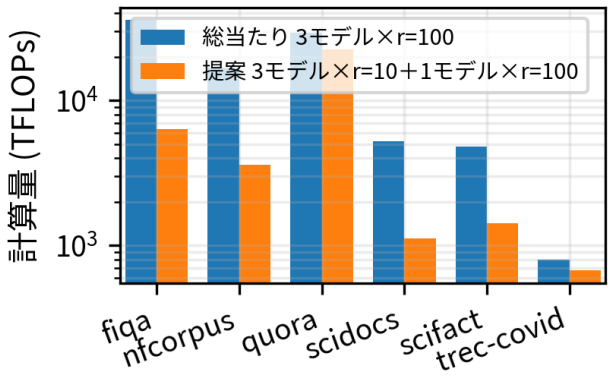

図 3　総当たり 3 モデル $\times r = 100$ と，実験 2 の推定に基づく手順 3 モデル $\times r = 10$ に加えて選択後 1 モデル $\times r = 100$ まで含めた際の追加学習に要する計算量の比較.

$$\mathrm{Cost}_{\mathrm{prop}}(10) = \sum_{c \in C} \left(\sum_{m \in M} \mathcal{F}(m, c, 10) + \mathcal{F}(\hat{m}^{\mathrm{lt}}(c; 10), c, 100) \right) \tag{8}$$

となる.

本実験範囲では，$\mathrm{Cost}_{\mathrm{brute}} = 93006.14\,\mathrm{T}$，$\mathrm{Cost}_{\mathrm{prop}}(10) = 35777.89\,\mathrm{T}$ であり，$\mathrm{Cost}_{\mathrm{prop}}(10)/\mathrm{Cost}_{\mathrm{brute}} = 0.385$ であった．すなわち，最終学習まで含めても，総当たりと比較して計算量が 61.5%減少する結果となった．図 3 にデータセット別の計算量を示す.

4.6　代理スコアと最終性能の順位相関

代理指標が最終性能とどの程度整合するかをより直接に確認するため，実験 1（$r = 0$）および実験 2（$r > 0$）で得られた代理スコアを用い，テストコレクションと候補モデルの組（6 つのテストコレクション $\times 3$ モデル，計 18 点）に対して代理スコアと最終性能の全点順位相関を算出した．各 r について，代理スコア（c_{val} 上の nDCG@10）に基づく 18 点の順位と，学習用データの全てを用いた追加学習（$r = 100$）後の最終性能（c_{test} 上の nDCG@10）に基づく 18 点の順位の Kendall の順位相関係数を $\tau_{\mathrm{all}}(r)$ と定義する．図 4 に，未適応状態（$r = 0$）の代理スコアと，学習用データの全てを用いた追加学習（$r = 100$）後の最終性能の散布図を示す.

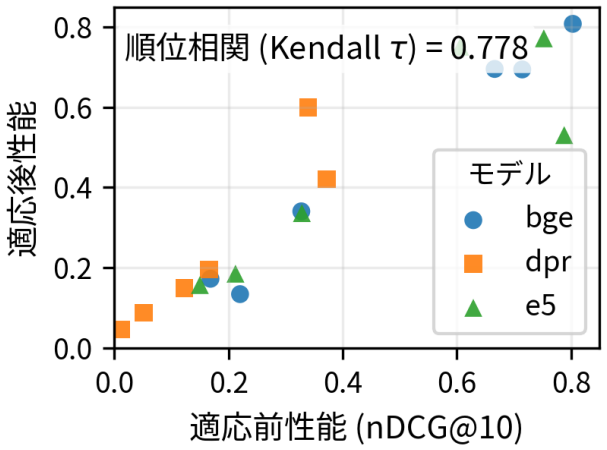

図 4　未適応状態（$r = 0$）の代理スコア（c_{val} 上の nDCG@10）と，学習用データの全てを用いた追加学習（$r = 100$）後の最終性能（c_{test} 上の nDCG@10）の散布図.

$r = 0$ では $\tau_{\mathrm{all}}(0) = 0.778$ であり，未適応状態の代理スコアが高いモデルほど，学習用データの全てを用いた追加学習後にも高い最終性能を示す傾向が見られた．図 5 に $\tau_{\mathrm{all}}(r)$ を示す.

また，$r = 10$ では $\tau_{\mathrm{all}}(10) = 0.843$ となり，$r = 0$ よりも最終性能の順位と強く相関した．一方で，$r \in \{25, 50, 75\}$ では $\tau_{\mathrm{all}}(r) = 0.830$，$r = 90$ では $\tau_{\mathrm{all}}(90) = 0.778$ であり，r の増加に伴って $\tau_{\mathrm{all}}(r)$ が単調に改善するとは限らない．一方で，$\tau_{\mathrm{all}}(r)$ はテストコレクション間の性能差も含むため，表 4 に示した各テストコレクション内の平均順位相関（平均 τ）とは解釈が異なる点に注意が必要であると考えられる．実験 1 の結果より，未適応状態での評価に基づくモデル選択は，テストコレクションによっては真に最適な候補モデル $m^*(c)$ を正しく推定できる一方で，本実験範囲では 6 つのテストコレクション中 4 つ

のテストコレクションで一致しなかった．特に trec-covid では，未適応状態では e5 が最良であると推定されるが，全量追加学習（$r = 100$）後の最良モデルは bge であり，$\Delta(c) = 0.1638$ の差が生じた．

一方，実験 2 の結果より，学習用データの一部のみを用いた追加学習後の評価に基づく推定は，未適応状態での評価より高い一致率と平均順位相関を示した（表 4）．ただし，推定精度および平均性能損失は r の増加に対して単調に改善するとは限らない（図 1）．

図 2 に示す通り，平均性能損失の増大は，特定のテストコレクションで大きな性能損失が生じる r が存在することに起因する．特に trec-covid では $r = 50, 75, 90$ で性能損失が大きく，この挙動が平均値を上昇させる要因となっている．

なお，trec-covid は本研究の分割で c_{val} および c_{test} のクエリ数が各 5 と小さい（表 1）ため，評価値の不安定さが推定結果に影響する可能性がある．

また，候補モデル数が 3 と少ないため，順位相関 τ は取り得る値が離散的であり，平均順位相関の解釈には注意が必要であると考えられる．

4.7 考　　察

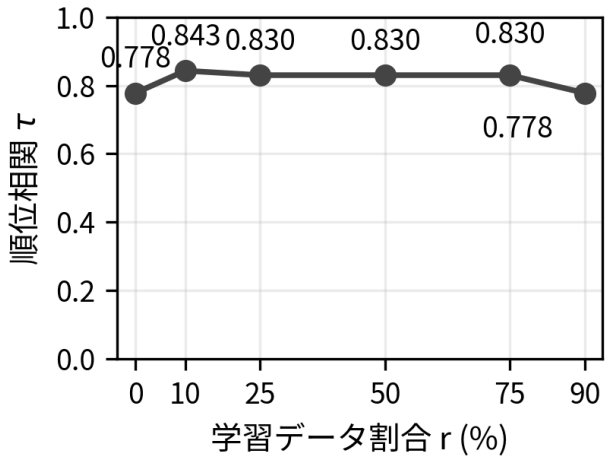

図 5　各 r における，全点順位相関 $\tau_{\text{all}}(r)$（Kendall の τ）．

5 ま と め

本研究は，情報検索システム構築におけるモデル選択コストの削減を目的とし，密検索モデルに追加学習を施してドメイン適応を行うことを前提とした候補モデル選択問題を対象とした．近年，密検索モデルが広く用いられている一方，その性能はドメインに強く依存するため，対象ドメインのデータを用いたドメイン適応が不可欠となる．しかし，候補となるすべてのモデルを対象ドメインで追加学習して評価する総当たりは，高い計算資源と時間を要する．そこで，自動ドメイン適応フレームワークの一要素としてモデル選択に焦点を当て，追加学習後に，より高い性能を発揮する有望な候補モデルを早期に推定する枠組みを検討した．

具体的には，テストコレクション c に対し，学習用データの全てを用いた追加学習（$r = 100$）後に評価用データ c_{test} 上での性能を最大化する真に最適な候補モデルを $m^*(c)$ と定義した．さらに，検証用データ c_{val} 上で算出可能な代理指標に基づいて推定最適候補モデル $\hat{m}(c)$ を得る手順を整理し，その推定精度を検証した．代理指標には，追加学習を行わない未適応状態での評価（$r = 0$）と，学習用データの一部である $r\%$ のみを用いて追加学習した後の評価を用い，それぞれが最終性能をどの程度予測できるかを検証した．

実験では，BEIR [15] に含まれる 6 つのテストコレクションを対象とし，3 種の候補モデル bge，dpr，e5 を用いて評価を行った．評価指標には nDCG@10 を採用し，モデル選択品質を一致率，性能損失 $\Delta(c)$，および順位相関により測定した．

実験の結果，未適応状態での評価（$r = 0$）に基づく推定は一致率 33.3%にとどまり，テストコレクションによっては大きな性能損失が生じた．とりわけ trec-covid では，未適応状態での評価により e5 が選択されたのに対し，学習用データの全てを用いた追加学習（$r = 100$）後の最良モデルは bge であり，$\Delta(c) = 0.1638$ の差が生じた．

一方，学習用データの一部 $r\%$ のみを用いた追加学習後の評価に基づく推定は，未適応状態での評価に基づく推定よりも高い精度を示した．本実験では $r = 10$ のときに一致率 83.3%，平均性能損失 0.0085（nDCG@10），平均順位相関 0.889 が得られた．ただし，推定精度は r の増加に対して単調に改善するとは限らず，例えば $r = 50$ では一致率が 33.3%まで低下した．

以上の結果は，未適応状態での評価のみでは追加学習後の性能を十分に推定できない場合があることを示す．これに対し，学習用データの一部を用いた追加学習後の評価は，有望モデルの早期選別に寄与し得る．しかし，推定精度と性能損失は r に対して非単調であり，データセットによっては特定の r において大きな性能損失が生じ得るため，平均値だけに依拠せず，データセット別の挙動を踏まえて r を設定する必要がある．また，本実験では候補モデル数が 3 と少なく，順位相関 τ の取り得る値が離散的となるため，平均順位相関の解釈には注意を要すると考えられる．

今後の課題として，候補モデルおよび評価対象テストコレクションを拡張し，事前学習手法，モデル規模，アーキテクチャが異なるより大規模な候補集合に対して，本枠組みの一般化可能性を検証することが挙げられる．さらに，機械学習分野で提案されてきた転移指標 LEEP [12] や LogME [19]，ならびに大規模モデル選択の枠組み PARC [2] や B-Tuning [20] を検索タスクへ適用することも検討対象となる．これらを未適応状態での評価や少量追加学習後の評価と組み合わせることで，より低コストなモデル推定に向けた可能性を検討する余地がある．

今後は，関連度ラベルが得られない新規ドメインを想定し，評価用テストコレクションの構築から候補モデルの選択・ドメイン適応，さらにリランキングモデルの選択・ドメイン適応までを一貫して扱う自動ドメイン適応フレームワークへ発展させることを目指す．

謝　辞

謝辞　本研究は JSPS 科研費 JP25K03229，JP23K28090，JP24K03048 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] Alessandro Achille, Michael Lam, Rahul Tewari, Avinash Ravichandran, Subhransu Maji, Charless C Fowlkes, Stefano Soatto, and Pietro Perona. Task2Vec: Task embedding for meta-learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6430–6439, 2019.

[2] Daniel Bolya, Rohit Mittapalli, and Judy Hoffman. Scalable diverse model selection for accessible transfer learning. Vol. 34, pp. 19301–19312, 2021.

[3] Zhongxiang Dai, Haibin Yu, Bryan Kian Hsiang Low, and Patrick Jaillet. Bayesian optimization meets bayesian optimal stopping. In *International conference on machine learning*, pp. 1496–1506. PMLR, 2019.

[4] Tobias Domhan, Jost Tobias Springenberg, Frank Hutter, et al. Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves. In *IJCAI*, Vol. 15, pp. 3460–8, 2015.

[5] 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.

[6] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, Vol. 20, No. 4, pp. 422–446, 2002.

[7] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *EMNLP (1)*, pp. 6769–6781, 2020.

[8] 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.

[9] Ekaterina Khramtsova, Shengyao Zhuang, Mahsa Baktashmotlagh, Xi Wang, and Guido Zuccon. Selecting which dense retriever to use for zero-shot search. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, pp. 223–233, 2023.

[10] Ekaterina Khramtsova, Shengyao Zhuang, Mahsa Baktashmotlagh, and Guido Zuccon. Leveraging llms for unsupervised dense retriever ranking. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1307–1317, 2024.

[11] Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2661–2671, 2019.

[12] Cuong Nguyen, Tal Hassner, Matthias Seeger, and Cedric Archambeau. Leep: A new measure to evaluate transferability of learned representations. In *International Conference on Machine Learning*, pp. 7294–7305. PMLR, 2020.

[13] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, Vol. 22, No. 10, pp. 1345–1359, 2009.

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

[15] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In J. Vanschoren and S. Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, Vol. 1, 2021.

[16] 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.

[17] Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, pp. 641–649, 2024.

[18] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? Vol. 27, 2014.

[19] Kaichao You, Yong Liu, Jianmin Wang, and Mingsheng Long. Logme: Practical assessment of pre-trained models for transfer learning. In *International Conference on Machine Learning*, pp. 12133–12143. PMLR, 2021.

[20] Kaichao You, Yong Liu, Ziyang Zhang, Jianmin Wang, Michael I Jordan, and Mingsheng Long. Ranking and tuning pre-trained models: A new paradigm for exploiting model hubs. *Journal of Machine Learning Research*, Vol. 23, No. 209, pp. 1–47, 2022.

[21] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, Vol. 109, No. 1, pp. 43–76, 2020.

地球環境データに対する周辺情報を用いた拡張的メタデータの検討

清水　敏之† 　中原　陽子†† 　島井　博行†††

† 九州大学附属図書館研究開発室　〒 819–0395 福岡県福岡市西区元岡 744
†† 国立情報学研究所コンテンツ科学研究系　〒 101–8430 東京都千代田区一ツ橋 2-1-2
††† 大阪成蹊大学データサイエンス学部　〒 533–0007 大阪府大阪市東淀川区相川 3 丁目 10-62
E-mail: †shimizu.toshiyuki.457@m.kyushu-u.ac.jp, ††nakahara_y@nii.ac.jp, †††shimai@g.osaka-seikei.ac.jp

あらまし　研究データの公開に際し，データを説明するためのメタデータは不可欠であるが，データ提供者により作成されるメタデータは，記述の不足や利用者の検索意図との乖離が生じる場合があり，研究データの発見性や理解可能性を低下させる要因となっている．一方，研究データを利用した論文や解説文書，利用事例といった周辺情報は，利用者の視点を反映した有用な情報源であり，近年では DOI を介したデータ引用の普及により，その活用環境が整いつつある．さらに，大規模言語モデル（LLM）の発展により，これらの非構造化情報を整理・統合して活用することが現実的になってきた．本研究では，従来の構造化メタデータを拡張し，周辺情報を「拡張的メタデータ」として活用する手法を検討する．具体的には，DIAS（データ統合・解析システム）において管理されている地球環境データを対象に，周辺情報としてデータ利用論文中のデータ引用テキストを用い，拡張的メタデータとしてキーワードを取得することを検討する．

キーワード　メタデータ，データセット検索，データ公開，オープンデータ

1 はじめに

研究データの公開と共有は，研究の透明性向上や再現性の確保，さらには新たな知見創出の基盤として重要性を増している．近年では，オープンサイエンスの潮流のもと，研究データに対する DOI の付与やデータ引用の推進が進められており，研究データを学術的成果物として位置づける取り組みが国際的に広がっている [9]．これに伴い，研究データを適切に発見・理解・再利用するための基盤として，メタデータの役割はますます重要となっている．

一般に，研究データのメタデータはデータ提供者によって作成され，データの内容や取得条件，形式などを記述することを目的としている．しかし，実際の運用においては，記述の粒度や詳細度にばらつきが生じやすく，また将来の第三者による利用を十分に想定した記述がなされていない場合も少なくない．その結果，利用者が自身の研究目的に適合するデータを探索・選択する際に，メタデータのみからデータの有用性を判断することが難しいという課題が指摘されている [6]．特に地球環境分野のように，観測条件や解析手法が多様で専門性の高い分野では，この問題は顕著である．

一方で，研究データは論文や技術報告書などにおいて実際に利用され，その文脈の中で解釈・評価されている．これらのデータ利用論文には，データの具体的な使用方法，適用分野，制約条件，さらには有効性や限界に関する知見が記述されており，データ提供者が付与する一次的なメタデータとは異なる，利用者視点の有用な情報が含まれている．DOI を介したデータ引用の普及により，データと論文の対応関係を機械的に把握できる環境が整いつつあり，こうした周辺情報を研究データの理解や検索に活用する可能性が高まっている．

さらに近年では，大規模言語モデル（LLM）の発展により，論文本文のような非構造化テキストから情報を抽出・要約し，構造化された知識として整理することが現実的となってきた [1]．これにより，データ利用論文に含まれる知見を体系的に整理し，研究データの検索等に活用する新たなアプローチが可能となりつつある．

本研究では，研究データを取り巻く論文情報などの周辺情報を，従来の構造化メタデータを補完する「拡張的メタデータ」として位置づけ，これを用いたデータセット検索の可能性を検討する．概念的には多様な周辺情報が拡張的メタデータに含まれ得るが，本論文では初期的検討として，データ利用論文に限定した検証を行う．具体的には，DIAS（データ統合・解析システム）[5][1] において管理されている地球環境データを対象とし，データ利用論文中のデータ引用テキストを用い，拡張的メタデータとしてキーワードを取得することを検討する．

2 関連研究

研究データの発見性および再利用性を高めるため，メタデータの設計や品質に関する研究はこれまで広く行われてきた．研究データリポジトリにおけるメタデータは，データの内容や取得条件，形式などを記述するための基本的な情報基盤として位置づけられており，その記述粒度や一貫性がデータ検索に大きく影響することが指摘されている [6]．また，分野横断的なデータ共有を促進する観点から，標準的なメタデータスキーマや語彙の整備も進められている [2], [3]．しかし，これらの研究の多

1：https://diasjp.net/

くは，主としてデータ提供者が付与する構造化メタデータを対象としており，データの利用文脈や事後的な評価を十分に反映することは難しいという課題が残されている．

一方，研究データの利用実態に着目し，論文情報や引用情報を通じてデータの価値や影響を捉えようとする研究も進められている．データ引用に関する研究では，研究データの共有が論文の被引用数増加と関連することが示されており [7]，データを学術的成果物として評価する動きが広がっている．また，論文中に記載されたデータ利用記述を分析することで，データの再利用可能性や適用分野を把握しようとする試みも報告されている [12]．これらの研究は，論文情報が研究データ理解にとって重要な周辺情報であることを示唆しているが，本研究ではデータセット検索への応用を意識しつつ，大規模言語モデル（LLM）の活用を前提としている点で焦点が異なる．

近年では，自然言語処理技術の発展により，論文本文などの非構造化テキストから情報を抽出し，知識として整理・活用する研究が活発化している．特に，大規模言語モデル（LLM）は，従来手法では困難であった要約や意味理解を可能にし，学術文献分析や検索支援への応用可能性が示されている [1], [8]．研究データ分野においても，論文や報告書を対象とした情報抽出を通じて，データ記述を補完するアプローチが検討されつつあるが，それらを体系的にメタデータとして位置づけ，検索機構に組み込む枠組みは十分に整理されていない．

以上のように，研究データのメタデータ設計，論文情報を用いたデータ理解，および LLM を用いた情報抽出に関する研究はそれぞれ蓄積されているものの，これらを統合し，論文等の周辺情報から抽出された情報を「拡張的メタデータ」として体系的に位置づけた点に本研究の特徴がある．本研究では，研究データを利用した論文に着目し，その記述から抽出されるキーワード等を拡張的メタデータとして捉える枠組みを提示するとともに，具体的なデータ基盤を対象とした初期的検討を行う．

3　拡張的メタデータの概念と対象範囲

3.1　拡張的メタデータ

研究データに付与されるメタデータは，データの内容や取得条件，形式，作成者などを構造化して記述するための基本的な情報として，データ管理や検索において重要な役割を果たしている．一方で，こうしたメタデータは主としてデータ提供者によって作成されるため，記述の粒度や観点が提供者側に依存しやすく，将来の第三者による多様な利用を十分に支援できない場合があることが指摘されている [6]．

このような課題に対し，本研究では，従来の構造化メタデータを補完・拡張する情報を総称して「拡張的メタデータ」と捉える．拡張的メタデータは，必ずしもデータ提供者によって事前に付与される情報に限られず，データの利用過程や解析文脈において新たに得られる情報や，既存メタデータから推論・補完された情報を含み得る．例えば，研究データを利用した論文に記述された利用方法や適用分野に関する知見は，データの理解や再利用を検討する上で有用な情報である [7]．

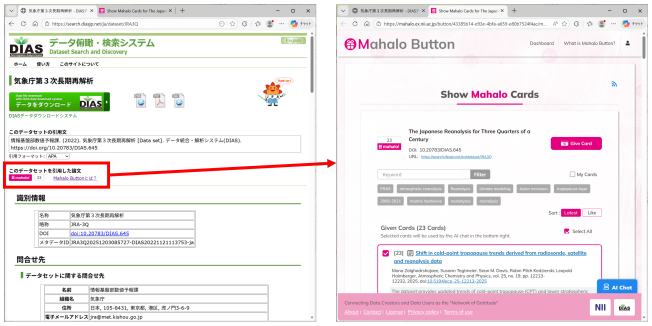

図 1: DIAS における Mahalo Button

また近年では，大規模言語モデル（LLM）の発展により，論文本文やメタデータといった既存情報から，データの特徴や利用可能性に関する示唆を推論的に抽出することも可能になりつつある．このような情報も，研究データの理解や検索を支援する拡張的メタデータとして位置づけることができる．本研究では，このような緩やかな概念として拡張的メタデータを捉え，その具体的な活用可能性について検討する．

3.2　DIAS における地球環境データ

本研究では，拡張的メタデータの検討対象として，DIAS（データ統合・解析システム）において管理・公開されている地球環境データを扱う．DIAS は，気象，海洋，水文，陸域など多様な分野にわたる地球環境データを統合的に管理するデータ基盤であり，観測データ，再解析データ，数値モデル出力など，性質の異なるデータセットが多数登録されている．これらのデータは研究論文において広く利用されており，地球環境分野における代表的な研究データ基盤の一つと位置づけられる．

一方で，DIAS に登録されているデータセットは多様性が高いがゆえに，メタデータの記述内容や詳細度にはばらつきが存在する．利用者がデータセットを検索・選択する際には，観測条件や解析手法との適合性，過去の利用実績といった情報が重要となるが，これらは必ずしも一次的なメタデータから十分に把握できるとは限らない．このような特性は，拡張的メタデータの有効性を検討する対象として DIAS が適している理由の一つである．

DIAS では，データ利用論文に関する情報を収集する仕組みとして，Mahalo Button [10] が導入されている．Mahalo Button は，研究者が論文執筆時に利用したデータを明示的に登録できる仕組みであり，DOI による形式的なデータ引用に限らず，本文中で言及されているデータ利用情報も含めて収集できる点に特徴がある．これにより，従来のデータ引用情報のみでは把握が困難であったデータ利用論文の情報を，体系的に蓄積・活用することが可能となっている．DIAS では，図 1 に示すように，各データセットのメタデータ閲覧ページにおいて Mahalo Button が設置されており，Mahalo Button を通してデータ利用論文に関する情報が取得可能である．

本研究では，このような DIAS の特性を踏まえ，拡張的メタデータの初期的検討として，データ利用論文に着目する．具体的には，図 2 に示すように，DIAS に登録された地球環境デー

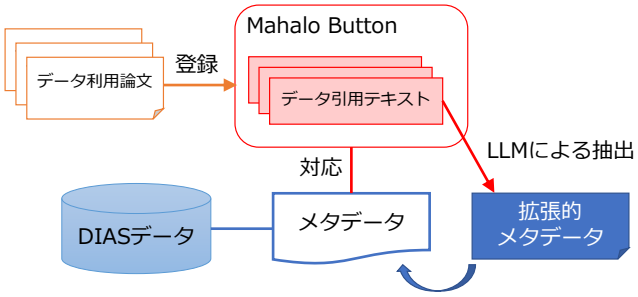

図 2: データ利用論文を用いた拡張的メタデータ

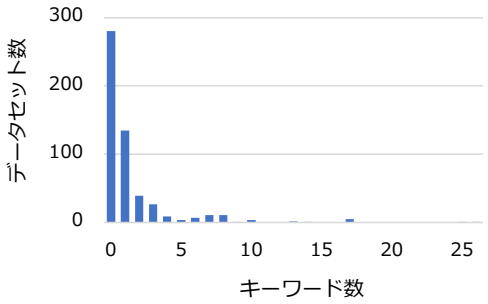

図 3: DIAS データセットに付与された GCMD Science Keywords 数の分布（2026 年 2 月現在）

タと，Mahalo Button 等を通じて収集されたデータ利用論文との対応関係を基に，論文中に含まれるデータ利用に関する情報を利用し，拡張的メタデータとして活用する．論文以外の周辺情報や，メタデータからの推論的情報の活用については，将来的な拡張可能性として位置づけ，本論文では対象外とする．

4 論文情報を用いた拡張的メタデータ生成手法

4.1 データ利用論文およびデータ引用テキスト

本研究では，拡張的メタデータ生成のための情報源として，研究データを利用した論文に関する情報を用いる．DIAS では，研究データの利用状況を把握する仕組みとして，Mahalo Button が導入されており，研究者が論文執筆時に利用したデータを登録することで，データと論文との対応関係を収集している．この仕組みの特徴は，DOI による形式的なデータ引用に限らず，論文本文中で研究データの利用が言及されている箇所も対象としている点にある．

Mahalo Button による登録情報は，現状では人手による判断に基づいており，論文中で研究データの利用が明確に読み取れる文章が抽出・登録されている．これらの文章は，主として段落単位であり，研究データの利用目的，適用分野，解析条件などが集中的に記述されていることが多い．本研究では，このように論文中で研究データの利用が言及されている文章を「データ引用テキスト」と呼ぶ．

データ引用テキストは，論文全文と比較して，研究データとの関連性が人手によって確認されている点に特徴があり，拡張的メタデータ生成の入力として適している．論文全文を対象とした分析では，データと直接関係しない記述も多く含まれるが，データ引用テキストに限定することで，データ利用文脈に関する情報を効率的に抽出できると考えられる．また，人手で登録されているという特性は，本研究における初期的検討において，信頼性の高い入力データとして位置づけることができる．

4.2 データ引用テキストを用いた拡張的メタデータ生成

本研究では，前節で述べたデータ引用テキストを入力として，大規模言語モデル（LLM）を用いた情報抽出を行い，拡張的メタデータを生成する．拡張的メタデータは多様な形式を取り得るが，本論文では初期的検討として，データセット検索への適用が容易な表現として，キーワードの抽出に着目する．

データ引用テキストには，研究データの利用目的や適用分野，解析条件などが自然言語で記述されている．これらの情報を整理せずにそのまま扱うことも可能であるが，拡張的メタデータとして体系的に活用するためには，一定の構造化が求められる．キーワードとして情報を抽出・整理することで，利用側文脈に関する情報を簡潔に表現でき，従来の構造化メタデータと比較可能な形で整理することが可能になると考えられる．

LLM を用いた情報抽出においては，データ引用テキストを入力とし，意味的に整理された語の抽出を試みる．プロンプト設計としては，過去の研究 [11] で設計したプロンプトを参考にしつつ，本研究では地球環境データを対象としていることから，初期的検討として GCMD Science Keywords [4] に着目してキーワードを抽出することを考えた．GCMD Science Keywords は，NASA が提供する地球科学分野の統制語彙であり，階層構造を持つことにより分野横断的なデータ検索を支援する．DIAS ではメタデータとして GCMD Science Keywords の付与を推奨しているが，図 3 に示す通り，多くのデータセットで十分な数のキーワードが付与されておらず，拡張的メタデータとして GCMD Science Keywords を付与することは有用であると考えられる．具体的なプロンプト例を付録の図 A·1 に示す．LLM による推論能力を活用することで，従来のキーワード抽出手法では困難であった，文脈に依存した情報の整理が可能になると考えられる．

4.3 抽出されるキーワードに関する議論

一口にキーワードと言っても，その性質や役割は一様ではない．データ引用テキストからは，対象となる現象や研究主題を表す語，研究データの利用目的や研究観点を示す語，解析手法やモデル名などの技術的語彙，さらには空間的・時間的条件を表す語など，異なる種類の情報が抽出され得る．本論文では初期的検討として，GCMD Science Keywords を取得することを考えたが，取得されたキーワードを一律に扱うのではなく，キーワードの種類という観点から整理することが，拡張的メタデータの設計において重要になる可能性があると考えている．

表 1 に示すように，情報探索における典型的な問い（5W1H）を整理の枠組みとして参照しつつ，既存の研究データメタデータでは十分に表現されていない要素に着目し，データ引用テキストから抽出するキーワードを，主題・研究対象，利用目的・研究観点，解析手法・モデル，空間条件，時間条件の 5 種類に分類

することを検討している．これらの観点のうち，特にデータ利用側の側面からの主題・研究対象や利用目的・解析手法に関する情報は，従来の構造化メタデータでは十分に記述されない場合が多く，拡張的メタデータとしての活用が期待される．なお，データ利用側の文脈に関する情報の抽出が重要になると思われるため，5W1H のうち Who に関する情報については，データ利用論文から抽出する拡張的メタデータの対象には含めないことを考えている．本論文で扱った GCMD Science Keywords は，この中で主に「主題・研究対象」に相当するキーワードになると考えられる．

5 ケーススタディ：キーワード推薦の比較

本節では，拡張的メタデータ生成の初期的検討として，周辺情報の有無が推薦される GCMD Science Keywords にどのような影響を与えるかをケーススタディとして検討する．以下の 2 条件でキーワード推薦を行った結果を議論する．

- 条件 A：データセットのタイトルおよび概要文（メタデータのみ）を入力
- 条件 B：データセットのタイトルおよび概要文に加え，データ引用テキストを入力

いずれの条件においても，同一の LLM および同一のプロンプト構造を用いた．具体的なプロンプトは，条件 B は図 A·1 を用いて出力したものであり，条件 A はそれからデータ引用テキストを除いたものである．出力は GCMD Science Keywords に限定した．なお，結果の提示にあたっては，GCMD Science Keywords のルート概念である「Earth Science」は除外し，表記ゆれ（全大文字など）は統一した上で比較を行った．

我々は，仮説として，周辺情報であるデータ引用テキストを用いることで，以下の優位性があるのではないかと考えた．これらの仮説を探索的に検討する．

- H1：周辺情報を加えることで，推薦キーワードの多様性が増加する可能性がある．
- H2：周辺情報を加えることで，より具体的なキーワードが推薦される可能性がある．

具体的には, 一定数のデータ利用論文がある FORP-NP10 version4（FORP_NP10_version4, doi:10.20783/DIAS.655）データセットおよび Global dataset of historical yield of major crops (version 1.2)（GDHY_v1_2, doi:10.20783/DIAS.528）データセットを用いて検討を行った．ChatGPT 5.2 を用いて得られた具体的な推薦結果を表 2 に示す.

5.1 ケース 1：FORP_NP10_version4

FORP_NP10_version4 に対する推薦結果を比較したところ, 条件 A（メタデータのみ）では, Ocean circulation patterns, Sea surface temperature, Salinity, Ocean heat content, Carbon cycle など，物理海洋学的な上位概念が中心となった．一方，条件 B（データ引用テキストあり）では，Ocean currents, Mixed layer depth, Marine heatwaves, Nutrients, Photosynthetically active radiation などが推薦された．両条件に共通して Sea surface temperature や Carbon cycle といった語は含まれていたが，条件 B では，

- 生物地球化学的側面（Nutrients）
- 指標的概念（Marine heatwaves）
- より具体的な物理量（Mixed layer depth）

など，利用文脈に依存した語が追加される傾向が見られた．階層の深さについて明確な差は確認できなかったものの，概念の粒度という観点では，条件 B においてより具体的な利用文脈を反映した語が出現している可能性が示唆される．

5.2 ケース 2：GDHY_v1_2

GDHY_v1_2 に対する推薦結果では，条件 A では Croplands，Agricultural lands，Food security，Atmospheric temperature，Precipitation など，比較的広い概念が中心であった．これに対し，条件 B では Agricultural productivity，Crop yield, Drought indices, Extreme weather, Climate variability などが推薦された．

特に，

- Croplands → Crop yield
- Agricultural lands → Agricultural productivity

のように，より具体的な成果指標や利用目的に関係する語への変化が観察された．また，Drought indices や Extreme weather といった，利用側の研究課題を直接反映する語が出現している点も特徴的である．この結果は，周辺情報を入力することで，データの利用文脈に即したより具体的かつ多様な概念が推薦される可能性を示唆していると考えている．

5.3 考　　察

本ケーススタディの範囲では, 周辺情報の導入により, 推薦キーワードの内容が変化することが確認された．特に GDHY_v1_2 の例では，メタデータのみでは得られなかった具体的な利用目的や成果指標に関する語が出現している．周辺情報の導入は，少なくとも一部のデータセットにおいて，利用文脈を反映したキーワード推薦に影響を与える可能性があることが示唆された．一方で，階層の深さという観点で明確な差を定量的に示すことはできなかった．GCMD Science Keywords は階層構造を有しているため，将来的には各キーワードの階層深度を用いた定量的比較を行うことを検討している．詳細な統計的検証は今後の課題とする．

6 データセット検索への適用に関する議論

拡張的メタデータはデータセットを検索する際に有用な情報になることが考えられる．本節では，前節までで検討した拡張的メタデータを，データセット検索に適用した場合の評価計画について述べる．特に，データ引用テキストから抽出した「利用側の文脈」に基づく情報が，検索においてどのような役割を果たし得るかを検討することが重要だと考えられる．評価に用いる検索要求の構築方法および評価の観点を中心に整理する．

表 1: データ引用テキストから抽出するキーワードの観点（利用側の文脈）

観点	説明	5W1H
主題・研究対象	研究データが論文中でどのような研究対象や主題として扱われたかを表す語．データの利用文脈における位置づけを示す観点である．	What
利用目的・研究観点	研究データがどのような研究目的や課題意識のもとで利用されたかを表す語．データの利用意図を理解するための観点である．	Why
解析手法・モデル	研究データが論文においてどのような解析手法やモデル，指標と組み合わせて利用されたかを表す語．	How
空間条件	論文中で解析対象として選択された地理的範囲や地点，空間スケールを表す語．	Where
時間条件	論文中で解析対象として選択された期間や時間分解能を表す語．	When

表 2: GCMD Science Keywords の推薦結果の比較

データセット	条件 A（メタデータのみ）	条件 B（データ引用テキストあり）
FORP_NP10_version4	Oceans > Ocean Circulation > Ocean Circulation Patterns	Oceans > Ocean Temperature > Sea Surface Temperature
	Oceans > Ocean Temperature > Sea Surface Temperature	Oceans > Ocean Circulation > Ocean Currents
	Oceans > Salinity/Density > Salinity	Oceans > Ocean Chemistry > Nutrients
	Oceans > Ocean Temperature > Ocean Heat Content	Oceans > Ocean Optics > Photosynthetically Active Radiation
	Oceans > Ocean Chemistry > Carbon Cycle	Oceans > Ocean Climate Indicators > Marine Heatwaves
	Oceans > Ocean-Atmosphere Interaction > Air-Sea Interaction	Oceans > Ocean Structure > Mixed Layer Depth
	Oceans > Ocean Waves > Coastal Processes	Climate Indicators > Atmospheric/Ocean Indicators > Sea Surface Temperature
	Oceans > Ocean Climate > Decadal Variability	Oceans > Ocean Biogeochemistry > Carbon Cycle
		Oceans > Ocean Dynamics > Air-Sea Interaction
		Oceans > Coastal Processes > Coastal Waters
GDHY_v1_2	Biosphere > Vegetation > Croplands	Biosphere > Agricultural Science > Agricultural Productivity
	Biosphere > Vegetation > Agricultural Lands	Biosphere > Agricultural Science > Crop Yield
	Human Dimensions > Food Security	Climate Indicators > Drought Indices
	Atmosphere > Atmospheric Temperature	Biosphere > Agricultural Science > Food Security
	Atmosphere > Atmospheric Precipitation	Land Surface > Land Use/Land Cover
	Biosphere > Ecosystems > Agroecosystems	Hydrosphere > Water Quality/Water Resources
	Land Surface > Land Use/Land Cover > Agricultural Lands	Climate Indicators > Climate Variability
	Biosphere > Vegetation Indices	Atmosphere > Weather Events > Extreme Weather
		Biosphere > Ecosystems > Terrestrial Ecosystems
		Human Dimensions > Natural Hazards > Drought

6.1 利用側文脈に基づく検索ニーズの把握に関する課題

研究データ検索において，利用者がどのような情報を手がかりとしてデータを探索しているかを把握することは重要である．特に，本研究が対象とする「研究データがどのように利用されたか」という利用側の文脈に基づく検索ニーズが，どの程度存在するかを把握することは，拡張的メタデータの意義を検討する上で不可欠である．

一方で，既存のデータ検索システムでは，解析目的や利用された手法といった利用側の文脈を明示的に表現する検索インタフェースが十分に整備されていない場合が多い．そのため，検索ログに記録されている検索語は，データセット名や変数名，地名や期間といった情報に偏りがちであり，利用側の文脈に基づく検索ニーズを直接的に反映していない可能性がある．この点は，検索ログを用いたニーズ分析の限界として認識しておく必要がある．このような制約を踏まえ，拡張的メタデータを検索に組み込んだ場合に，どのような検索が新たに可能になるかを検討することが重要である．

6.2 評価用検索要求の構築方法

拡張的メタデータを用いた検索の有効性を評価するためには，評価用の検索要求をどのように構築するかが重要となる．本研究では，実際の検索ログに完全に依存するのではなく，複数の情報源を組み合わせて検索要求を構築することを検討している．

一つの方法として，検索ログ中の検索語を分析し，解析目的や研究対象を示唆する語が含まれている検索要求を抽出することが考えられる．ただし，前節で述べたように，既存システムでは利用側文脈が十分に表現できていない可能性があるため，この方法は補助的な位置づけとする．

もう一つの方法として，データ利用論文の記述に基づいて検索要求を構築する方法が考えられる．例えば，論文の導入部や研究目的の記述，あるいはデータ引用テキストそのものから，「この研究で必要とされたデータを検索するとしたらどのような表現になるか」という観点で検索要求を生成する．この方法により，利用側の文脈を明示的に含む検索要求を評価用に用意することが可能となる．

6.3 拡張的メタデータを用いた検索の評価方法

データ利用論文の記述に基づいて構築した検索要求を用いる場合，評価の一つの方法として，検索結果に元の論文で利用されたデータセットが含まれるかどうかを確認することが考えられる．ただし，同一論文から抽出された拡張的メタデータを用いて検索を行うことは評価として適切ではないため，検索対象には当該論文由来の拡張的メタデータを含めない設計が必要と

なる．

この条件のもとで，検索結果に元の論文で利用されたデータセット，あるいは同様の利用文脈を持つ別のデータセットが含まれるかどうかを確認する．同様の利用文脈を持つ別のデータセットの判定には人手による評価が必要になるため，限定的な検索要求に対する妥当性確認にとどめることが想定される．

7 おわりに

本研究では，研究データの検索において，従来の構造化メタデータでは十分に表現されてこなかった「研究データがどのように利用されたか」という利用側の文脈に着目し，データ利用論文に含まれる記述を拡張的メタデータとして活用する可能性について検討した．特に，データ引用テキストを入力として，大規模言語モデルを用いて利用文脈に基づくキーワードを抽出し，データセット検索への適用を検討した点に本研究の特徴がある．

本研究の検討を通じて，利用側の文脈に関する情報は，解析目的や研究観点，解析手法や条件といった形で，データ引用テキスト中に明示的に記述されている場合が多く，これらを整理・構造化することで，従来のメタデータ検索では捉えにくかった検索軸を提供できる可能性が示唆された．このことは，研究データの再利用を支援する上で，利用事例に基づく情報が有用であることを示すものと考えられる．

今後の課題としては，拡張的メタデータをデータセット検索に活用し，有効性を検証したいと考えている．その際，6 節で議論した通り，利用側の文脈に基づく検索ニーズそのものが，既存のデータ検索システムや検索インタフェース上で十分に表現されていないという問題がある．検索ログに記録されている検索語は，データセット名や地名，期間といった一次的なメタデータに対応する語に偏る傾向があり，解析目的や研究手法といった利用側の文脈を直接反映していない可能性が高い．このため，検索ログのみを用いて，利用文脈検索のニーズや効果を評価することには限界がある．また，データ利用論文の記述に基づいて検索要求を構築する評価方法についても，評価の解釈には注意が必要である．より直接的には，利用側の文脈に基づく検索ニーズをより適切に把握するための仕組みを実装することも考えられる．例えば，解析目的や利用条件を明示的に入力できる検索インタフェースの設計や，検索時の意図をより詳細に記録できるログの収集が挙げられる．これらが整備されることで，拡張的メタデータの効果を定量的に評価するための基盤が整うと考えられる．

また，本研究では拡張的メタデータの初期的表現としてキーワード抽出に着目したが，より高次の表現形式についても検討の余地がある．例えば，利用目的や解析手法の関係性を記述する構造化表現や，複数のデータ利用事例を統合した利用パターンの抽出などが考えられる．さらに，データ利用論文以外の周辺情報を拡張的メタデータとして取り込む可能性についても，今後検討を進める必要がある．

以上のように，本研究は，研究データ検索における利用側の文脈の重要性を明示し，拡張的メタデータという観点からその活用可能性を整理した点に意義がある．評価や実装に関する課題は残されているものの，本研究で示した枠組みは，今後の研究データ管理および検索支援の高度化に向けた基礎的検討として位置づけられる．

謝　辞

本研究では，文部科学省の補助事業「地球環境データ統合・解析プラットフォーム事業」（JPMXD0721453504）により開発・運用されているデータ統合・解析システム（DIAS）の下で収集されたメタデータを利用した．

文　献

[1] Rishi Bommasani, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.
[2] DataCite. Datacite metadata schema 4.6. `https://schema.datacite.org/`, 2024.
[3] Dublin Core Metadata Initiative. Dublin core metadata element set, version 1.1. `https://www.dublincore.org/specifications/dublin-core/dces/`, 2012.
[4] Global Change Master Directory (GCMD). GCMD keywords. `https://forum.earthdata.nasa.gov/app.php/tag/GCMD+Keywords`.
[5] Eiji Ikoma and Masaru Kitsuregawa. DIAS—earth environment data integration and analysis system. *Communications of the ACM*, Vol. 66, No. 7, pp. 85–86, jun 2023.
[6] Jung-Ran Park. Metadata quality in digital repositories: A survey of the current state of the art. *Cataloging & Classification Quarterly*, Vol. 47, No. 3-4, pp. 213–228, 2009.
[7] Heather A Piwowar, Roger S Day, and Douglas B Fridsma. Sharing detailed research data is associated with increased citation rate. *PLoS ONE*, Vol. 2, No. 3, p. e308, 2007.
[8] Karan Singhal, et al. Large language models encode clinical knowledge. *Nature*, Vol. 620, pp. 172–180, 2023.
[9] Mark D. Wilkinson, et al. The fair guiding principles for scientific data management and stewardship. *Scientific Data*, Vol. 3, p. 160018, 2016.
[10] 北本朝展, 中原陽子, 清水敏之, 島井博行, 吉川正俊. Data citation and mahalo button: Collecting and sharing dataset usage in dias. 日本地球惑星科学連合 (JpGU)2023 年大会, No. MGI27-04, 2023.
[11] 篠原桐, 清水敏之, 冨浦洋一. 流通を考慮した研究データのメタデータ付与支援. 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, 2024.
[12] 角掛正弥, 松原茂樹. 研究データ検索における論文上の引用文脈の利用. 言語処理学会第 27 回年次大会, 2021.

付　　録

```
#Instruction
Please follow the steps in #Procedure to suggest keywords.
However, please follow the #Constraints restrictions.

#Dataset_Information
- Dataset name
(Insert dataset name here)

- Explanation
(Insert dataset description here)

#Dataset_Citation_Context
(Insert concatenated dataset citation texts here)

#Constraints
- Follow the specified #Output_Format.
- Don't use sentences for Suggested keywords.
- Do not use proper noun for Suggested keywords.
- Don't use words included in #Dataset_Information for
    Suggested keywords.

#Procedure
- Based on #Dataset_Information and #Dataset_Citation_Context,
    consider research using this dataset.
- #Dataset_Information is information about the dataset you
    want to search.
- #Dataset_Citation_Context is the concatenation of citation
    texts for the dataset.
- Extract as many keywords as possible from
    #Dataset_Information.
- Suggest keywords that exclude previously extracted keywords
    and are suitable for searching research using this
    dataset.
- List keywords necessary for searching this dataset for this
    research.
- If a keyword in Suggested keywords exactly matches a GCMD
    Science Keyword, retrieve the corresponding GCMD Science
    Keyword with its hierarchical structure.

#Output_Format
- Research using this dataset (10 in total)
- All extracted keywords
- Suggested keywords (10 in total)
- GCMD Science Keywords (up to 10)
```

図 A· 1: GCMD Science Keyword を推薦するプロンプト例

一般発表 | Track 3: 情報検索・情報推薦・ソーシャルメディア

2026年2月28日(土) 13:00 ~ 15:10 | F会場

[2F] 情報検索

座長:金子 邦彦(福山大学)　コメンテータ:渡辺 知恵美(筑波技術大学)　ジュニアコメンテータ:橋口 友哉(兵庫県立大学)

14:40 ~ 15:05

[2F-05] [技術報告] 生成AI時代の情報マネジメントにおける検索の役割 ― 非構造化業務データを対象とした設計と運用の知見

*清田 陽司[1,2] (1. 株式会社FiveVai、2. 麗澤大学)

発表者区分：スポンサー
種別：技術報告
インタラクティブ発表：あり

キーワード：検索モデル（言語モデル／ランキング学習）、インタフェース・インタラクション、人間中心情報マネジメント

生成AIの発展により、自然言語による情報アクセスやRAGを用いたシステム構築が広く普及しつつある。一方で、実務で扱われる業務データは、形式や粒度が不均一な非構造・半構造データであることが多く、生成AIやベクトル検索のみでは安定した情報活用が困難な場面も少なくない。本報告では、株式会社FiveVaiにおける業務システム開発の実践事例をもとに、生成AI時代における情報マネジメントの観点から、検索が果たす役割を再考する。具体的には、業務文脈を反映したメタデータ設計、検索方式の選択、生成AIとの役割分担といった設計判断、および非構造業務データを対象とした運用上の工夫について報告する。検索と生成AIを代替関係として捉えるのではなく、両者を補完的に組み合わせる設計の重要性を示し、データ工学および情報マネジメント分野における実践的示唆と今後の課題を提示する。