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.