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

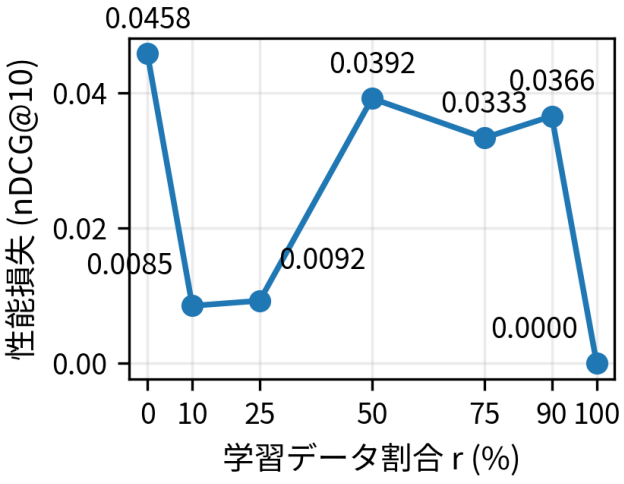

図 1　学習用データの割合 r に対する平均性能損失の推移．

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

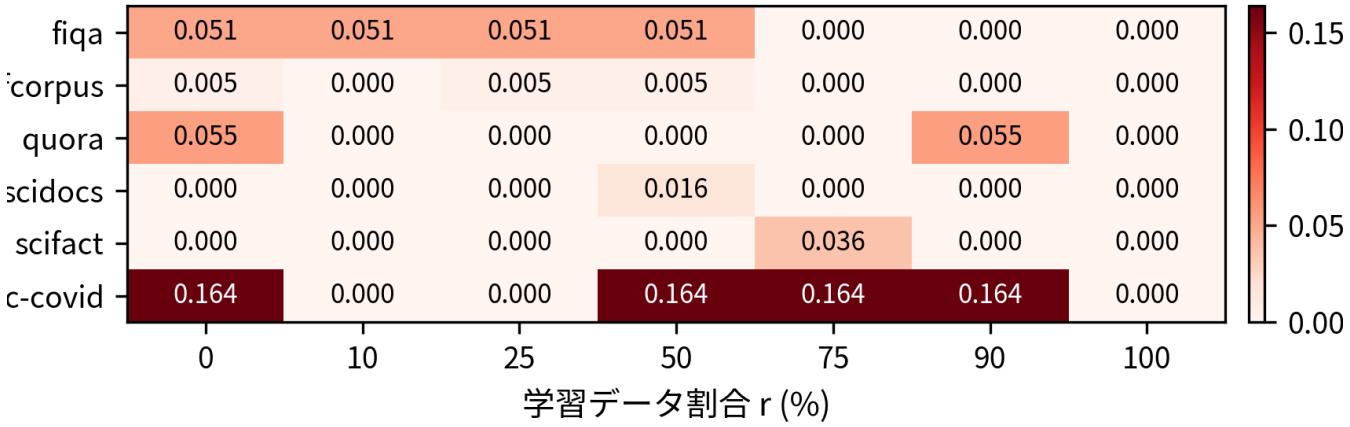

図 2　テストコレクション c と学習用データの割合 r ごとの性能損失 $\Delta(c)$（nDCG@10 の差）.

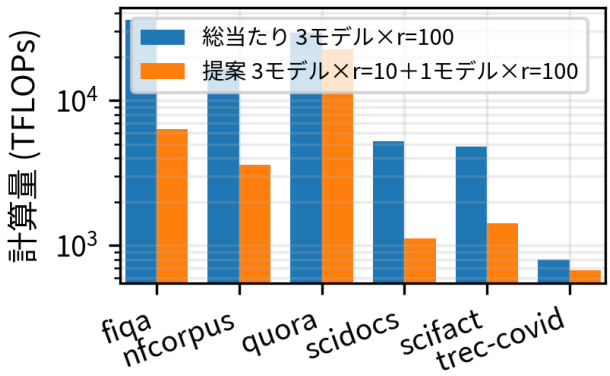

図 3　総当たり 3 モデル $\times r = 100$ と，実験 2 の推定に基づく手順 3 モデル $\times r = 10$ に加えて選択後 1 モデル $\times r = 100$ まで含めた際の追加学習に要する計算量の比較.

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

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.

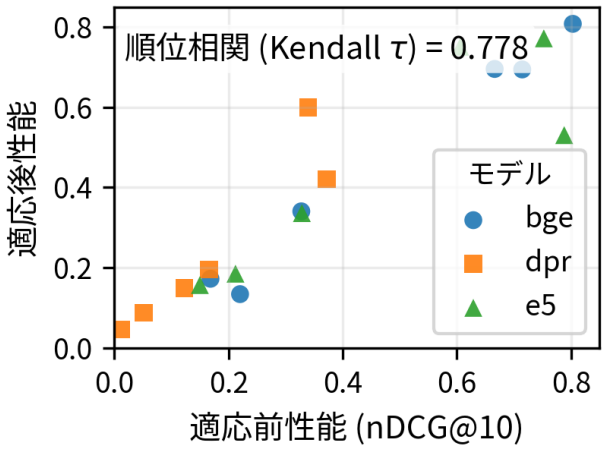

図 4　未適応状態（$r = 0$）の代理スコア（c_{val} 上の nDCG@10）と，学習用データの全てを用いた追加学習（$r = 100$）後の最終性能（c_{test} 上の nDCG@10）の散布図.

$r = 0$ 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

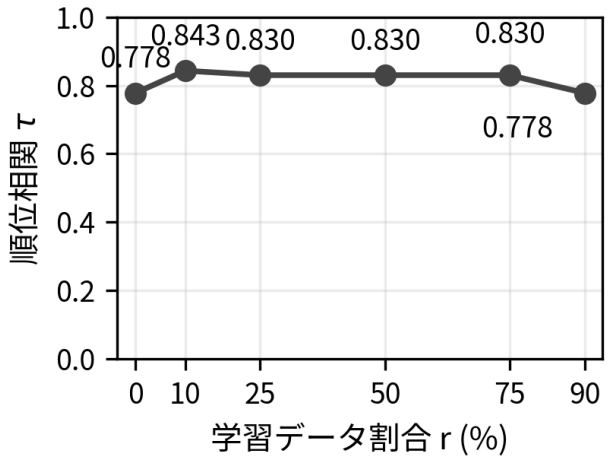

図 5　各 r における，全点順位相関 $\tau_{\text{all}}(r)$（Kendall の τ）．

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.