大規模言語モデルを用いた情報検索のための coarse-tuning 手法の提案

欅 惇志 [1]　田中 リベカ [2]
[1] 一橋大学 ソーシャル・データサイエンス教育研究推進センター
[2] お茶の水女子大学 文理融合 AI・データサイエンスセンター
a.keyaki@r.hit-u.ac.jp　tanaka.ribeka@is.ocha.ac.jp

概要

大規模言語モデルを用いた情報検索 (LLM-IR) の fine-tuning では，タスクに特化した学習に加えて，クエリの埋め込み表現（クエリ表現）とクエリ–文書の関係を学習する必要がある．本研究では，pre-training と fine-tuning を繋ぐ中間段階の学習として coarse-tuning を導入する．coarse-tuning においてクエリ表現とクエリ–文書の関係を学習することで，fine-tuning の負荷を軽減して学習効果の向上を目指す．その際，クエリ–文書ペアの適切性を推定する Query–Document Pair Prediction (QDPP) を提案する．評価実験の結果，提案手法によって情報検索タスク Robust04 の検索性能 (nDCG@20) が 6% 向上した．

1　はじめに

BERT [1] の登場によって情報検索コミュニティにおいてパラダイムシフトが生じた．BERT 以前にも深層学習ベースの情報検索手法（pre-BERT 深層学習）[2, 3, 4] は多数提案されていたが，文献 [5] における徹底的な実験によって，適切にハイパーパラメータ・チューニングされた古典的手法 [6] に対して pre-BERT 深層学習ベースの手法は実質的な性能改善はないということが示された．そのような中，BERT の登場により，大規模言語モデルを用いた情報検索 (LLM-IR) 手法の性能は大幅に改善した [7].

ただし，本来は大規模言語モデルを用いた fine-tuning では軽微な追加学習で高性能の達成が期待されるにも関わらず，実情は情報検索データセットを用いて fine-tuning をするだけでは大幅な性能改善はもたらされない [8]．高い性能を発揮するためには高コストな fine-tuning（例えば，BERT の出力を情報検索特化型のネットワークに入力して行う追加の学習 [9, 10] や大量データ [11] を用いた fine-tuning）が必要である．非事前学習型の Transformer ベースの検索手法 [12, 13] よりも BERT などの事前学習をベースとした検索手法がより高性能を示すことから，大量の言語資源を用いた事前学習の効果は明らかであるものの，高コストな fine-tuning を前提とした LLM-IR では事前学習の恩恵を最大限に受けられているとはいえない．

高コストな fine-tuning が必要となる一因として，入力データ構造の違いが考えられる．大規模言語モデルの pre-training の入力は自然文のテキストである．対して，情報検索タスクの入力は，数語程度の単語列である**クエリ**と数十から数百語程度の自然文である**文書**という不均衡な二つ組から構成されるデータ構造を持つ．更に，クエリは自然文の文法に従わない [14] ため，自然文を用いて訓練されたモデルとは齟齬がある[1)]．BERT の pre-training においても，クエリの性質を反映した埋め込み表現（クエリ表現）を適切に学習できていない可能性が高い．また，クエリと文書は文法や語数に大きな相違があるものの，出現語には一定の傾向がある．すなわち，クエリに適合する候補文書中にはクエリ語やその関連語が出現し[2)]，無関係な文書には出現しない．BERT の pre-training では，このようなクエリ–文書の関係も学習されていない．従って，LLM-IR の fine-tuning では，後段タスク（クエリに対する適合度順の検索結果作成）に特化した学習に加えて，クエリ表現とクエリ–文書関係の学習を全て同時に行う必要がある．以上の理由から LLM-IR における fine-tuning が高コストになっていると推測される．

そこで，本研究では，pre-training と fine-tuning を繋ぐ中間段階の学習である coarse-tuning を提案す

1)　例えば，自然文で学習した品詞推定器をクエリの品詞付与に適用した場合には性能が低下する [14, 15, 16] ことが報告されている．
2)　文書中の語をサンプリングして擬似クエリを生成して性能改善を行う研究も多数存在する [17, 18, 19].

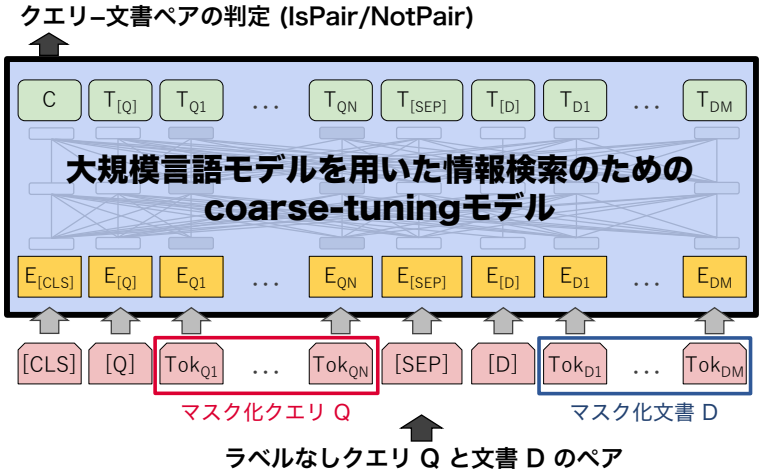

図 1　coarse-tuning モデルの概要図

る．coarse-tuning においてクエリ表現とクエリ–文書関係を学習することで，fine-tuning ではクエリに対する文書の適合度の推定に注力し，検索性能を改善することを目指す．

BERT の事前学習では単語の穴埋め問題を解く Masked Language Model (MLM) と文の連続性を予測する Next Sentence Prediction (NSP) が採用されている．本研究においても，クエリ表現の学習に MLM を用いる．加えて，クエリ–文書関係の学習では，NSP に着想を得て，クエリ–文書ペアの適切性を推定する **Query–Document Pair Prediction (QDPP)** を提案する．本稿では，クリックログ（検索エンジンに発行されたクエリとクリックされた検索結果中の文書の情報）からクエリ–文書ペアを作成する．

情報検索データセットである Robust04 における性能評価の結果，fine-tuning に先立ち coarse-tuning を適用することで，nDCG@20 が 6%向上した．また，文書からクエリを推定するタスクにおいて，coarse-tuning によってクエリ表現やクエリ–文書関係が学習されたことを示唆する結果が得られた．

2　提案手法

coarse-tuning の学習モデルの概要を図 1 に示す．モデルのアーキテクチャは，BERT と同様に，複数層の双方向 Transformer で構成される．coarse-tuning では MLM と QDPP が同時に行われることで，クエリ表現とクエリ–文書の関係が学習される．

2.1　モデルの入力

coarse-tuning の学習モデルは情報検索タスク特化型であるため，クエリと文書が入力される．クエリと文書はそれぞれトークン化され，特殊トークンとともに単一のシーケンスとして結合される．その際，文書トークンの末尾が最大トークン長を超える場合には，最大トークン長以降の文書トークンは除去する．また，一般的にクエリ長よりも文書長が大きい傾向があるが，稀に長文クエリが存在する．長文クエリは無意味な文字列であったり，そうでなくても学習への悪影響が懸念されるため，経験則的に最大トークン長の半分よりも大きなトークン長を持つクエリはノイズとして除外する．なお，特殊トークンには，BERT で用いられる `[CLS]` `[SEP]` `[PAD]` に加えて，`[Q]` と `[D]` が含まれる．`[Q]` と `[D]` は LLM-IR の fine-tuning で用いられる特殊トークン [20] である．`[Q]` はクエリトークンの前に挿入され，クエリを表す．`[D]` は文書トークンの前に挿入され，文書を表す．

2.2　Masked Language Model

深い双方向のクエリ表現の学習のため MLM による学習を行う．まず，クエリと文書それぞれをトークン化する．その後，一定割合のトークンをランダムにマスク化（`[MASK]` トークンと置換）する．マスク化されたトークンに対して，マスク化前のオリジナルのトークンを推定することで学習が行われる．

2.3　Query–Document Pair Prediction

QDPP では，入力されたクエリと文書ペアが適切か (`IsPair`) 不適切か (`NotPair`) を推定するタスクを通して，クエリ–文書の関係を学習する．クエリ–文書ペアの適切性判断として最も信頼性の高い情報源は，情報検索データセット中の qrel（クエリ–文書に

対する適合性判定）である．しかしながら qrel のアノテーションコストは高く，大量に作成することは難しいため，汎用的なクエリ–文書関係の学習に用いるデータセットとしては適切とはいえない．これを踏まえると，coarse-tuning 用のデータが満たす条件は下記の通りである．

- クエリと文書を含む
- クエリ–文書の何らかの関係性を保持する
- （研究用途で）されている

これらの条件を満たすデータとして，ORCAS (Open Resource for Click Analysis in Search) [21] が該当する．ORCAS は研究用途で利用可能なクリックログデータセットであり，1,900 万のクエリ–文書ペア（1,000 万種類のクエリ）が含まれる．Bing で収集された膨大な量のログからノイズを除去したうえで，k-匿名化と不適切なクエリの除去を行っている．

クエリ–文書のクリック関係は適合性とも関連を持ち，クリックされた文書を擬似適合文書として扱うことで性能改善が実現した事例も多数存在する [22, 2, 23]．従って，本研究では，クリック関係を持つクエリ–文書を適切なクエリ–文書ペアとする．

2.4 学習手順

coarse-tuning はクエリ表現とクエリ–文書関係の学習に主眼を置いており，汎用的な言語表現の獲得は目的としていない．従って，既に言語表現が学習されている事前学習済みモデルに対して，coarse-tuning を適用することを想定する．下記の手順にて後段の情報検索タスク用モデルが学習される．

1. 事前学習済みの大規模言語モデルを取得する
2. coarse-tuning （MLM と QDPP) を行う
3. 情報検索データセットで fine-tuning を行う

3 評価実験

3.1 実験設定

比較手法は下記の 4 手法である．

1. **`pre-trained`** 事前学習済みモデルを用いる
2. **`coarse-tuned`** 事前学習済みモデルに対して coarse-tuning を行う
3. **`fine-tuned (baseline)`** 事前学習済みモデルに対して fine-tuning を行う
4. **`coarse+fine (proposed)`** 事前学習済みモデルに対して coarse-tuning 後に fine-tuning を行う

表 1 情報検索タスク Robust04 の性能評価

	nDCG@20	P@20
`pre-trained`	0.062	0.066
`coarse-tuned`	0.045	0.048
`fine-tuned (baseline)`	0.327	0.307
`coarse+fine (proposed)`	**0.347**	**0.322**

なお，本稿では，単純化のため，fine-tuning は coarse-tuning と同一のモデルを用いてクエリ–文書ペアの適合度ラベルを推定する分類タスクとして学習を行い，適合ラベルの推定確率を文書のスコアとした．

検索性能評価では単一の設定に対して 3 回の試行を行った平均値を用いる．事前学習済みモデルには `prajjwal1/bert-tiny`[3] (L=2, H=128) を用いて，最大トークン長は 256 とした．その他の設定については A.1 を参照されたい．予備実験の結果，最適な設定は，ORCAS のサンプリング率 5%，coarse-tuning の epoch 数 1，fine-tuning の epoch 数 2 となった．予備実験結果の一部も A.1 に掲載する．次節の実験では最適な設定を用いた際の性能を報告する．

3.2 情報検索データセット

情報検索データセットは Robust04 [24] を用いる．Robust04 はキーワードマッチに基づく古典的な検索手法では性能が低い高難易度のクエリから構成されているため，近年の LLM-IR の評価でも多用される．250 個のクエリ，50 万件のニュース記事，31 万件の qrel（クエリ–文書の適合性判定）を含み，平均的な情報検索データセットと比較してクエリあたりの qrel 数は大きく，リッチな評価が行われている．qrel の適合度ラベルは，**非適合**（94.4%），**適合**（5.3%），**強く適合**（0.3%) の 3 段階であり，評価実験では少数ラベルである**強く適合**は**適合**に変換して用いた．fold1–fold4 の 200 クエリを訓練，fold5 の 50 クエリを評価クエリとして用いる．評価指標は，関連研究 [17] に準拠して nDCG@20 と P@20 を採用する．nDCG@k は順位を考慮した指標であり，より上位により多くの適合文書を提示することで高い値を示すため，情報検索の評価では P@k より重視される．

3.3 後段の情報検索タスクに関する評価

Robust04 による評価実験結果を表 1 に示す．比較手法のなかで最も高精度な検索精度を示した手法は `coarse+fine (proposed)` であり，`fine-tuned`

3) https://huggingface.co/prajjwal1/bert-tiny

表 2　クエリ表現とクエリ–文書関係の獲得に関する評価

	pre-trained			coarse-tuned			fine-tuned (baseline)			coarse+fine (proposed)		
	Tok_{Q1}	Tok_{Q2}	Tok_{Q3}	Tok_{Q1}	Tok_{Q2}	Tok_{Q3}	Tok_{Q1}	Tok_{Q2}	Tok_{Q3}	Tok_{Q1}	Tok_{Q2}	Tok_{Q3}
Top1	##lty	##lty	##lty	data	data	data	##ify	##ify	##ify	tv	tv	show
Top2	##nty	##tness	##tness	how	information	information	sure	##now	##now	show	show	tv
Top3	##gles	##rked	##rked	what	search	search	##now	sure	afford	oclc	chart	com
Top4	##tness	##gles	##rky	information	of	citations	guess	guess	guess	chart	oclc	chart
Top5	##rked	##nty	##lish	computer	.	wikipedia	##qui	##pass	sure	com	written	written

(baseline) と比較して nDCG@20 は 6% 改善した．後段タスクに特化した学習を行っていない pre-trained と coarse-tuned では想定の通り低い検索性能が確認された．fine-tuning による性能の改善が認められ，さらに，先立って coarse-tuning を行うことで更に高精度な検索性能を達成した．

3.4　クエリ表現とクエリ–文書関係の獲得に関する評価

coarse-tuning によってクエリ表現やクエリ–文書関係が獲得できたかどうかを評価するため，文書からクエリを予測するタスクを行った．具体的には，与えられた文書からシーケンスを作成する際に，クエリトークンの位置に [MASK] トークンを挿入する．その後各モデルで [MASK] トークンを予測する．ORCAS クエリの平均は 3.27 語のため，予測するクエリのトークン数は 3 とする．

表 2 は，英語版 Wikipedia 記事の “Information retrieval”[4] の冒頭から最大トークン長まで入力した際に予測されたクエリトークン (Tok_{Q1}, Tok_{Q2}, Tok_{Q3}) の上位 5 件である．pre-trained と fine-tuned (baseline) ともにサブワードを含むランダムなトークン群が予測されている．coarse-tuned では，クエリの先頭トークンである Tok_{Q1} において “how”や “what”，２つ目のトークンである Tok_{Q2} において “of” が出現しており，ある程度のクエリ表現が学習できていると考えられる．また，“data”, “information”, “search” は入力中に出現する語であり，“computer” と “citations” は入力された範囲よりも後の文書中に出現する語である．このことから，予測されたクエリが文書中の語やその関連語から構成されるため，クエリ–文書関係も学習されている形跡が認められた．なお，Tok_{Q3} の “wikipedia” は入力中に含まれないが，ORCAS クエリの中には末尾に “Wikipedia” を含むクエリが多数存在するため，その特徴が現れていると考察される．

また，coarse+fine (proposed) では，予測されているトークン群に一貫性があるものの，入力された文書とは異なるトピックに関するトークン群が出力されている．これは，fine-tuning の過程で，coarse-tuning で学習されたクエリ表現やクエリ–文書関係の一部が失われたためであると考えられる．

今後の課題として，詳細な分析を通したより高性能な coarse-tuning 手法の提案や，クエリ表現やクエリ–文書関係を損失しない fine-tuning 手法の提案を行う予定である．また，より大きな BERT モデルを用いた場合の振る舞いの差異の評価や，より高性能な fine-tuning 手法の適用も今後の課題である．

4　関連研究

文献 [25] では後段タスクに応じた事前学習の必要性が主張されている．LLM-IR に関する研究の主流は fine-tuning であるが [26, 27, 28]，事前学習に着目した研究も存在する [17, 18, 19, 29]．文献 [17, 18, 19] ではいずれも文書から生成した疑似クエリを事前学習に利用しているため，実クエリ–文書ペアからクエリ表現やクエリ–文書関係を学習する本研究とはアプローチが異なる．WebFormer [29] は事前学習に Web 文書の構造を利用する手法であり，本研究とは着眼点が異なる．文書からクエリを生成する doc2query [30] や docTTTTTquery [31] は実クエリ–文書ペアを用いて学習を行うという点で共通点するが，クエリ–文書ペアが相互にインタラクションを行う本研究とは目的が異なる．効率性向上のためクエリと文書のインタラクションを遅延させる ColBERT [20] は本研究と対極に位置するといえる．

5　おわりに

本研究では，大規模言語モデルを用いた情報検索において，pre-training と fine-tuning を繋ぐ coarse-tuning を提案した．coarse-tuning は，クエリ表現を学習する MLM とクエリ–文書関係を学習する QDPP から構成される．coarse-tuning によって，後段の情報検索タスクの検索性能向上が確認された．また，文書からクエリを予測するタスクにおいて，クエリ表現とクエリ–文書関係の獲得が示唆された．

4)　https://en.wikipedia.org/wiki/Information_retrieval

謝辞

本研究の一部は，学術研究助成基金助成金研究活動スタート支援（課題番号 22K21303）の助成を受けて遂行された．また，本研究を遂行するうえで必要不可欠であった計算機を提供頂いた一橋大学ソーシャル・データサイエンス教育研究推進センター教授の七丈直弘氏に謝意を表する．

参考文献

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In **Proc. of the NAACL 2019**, 2019.

[2] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. Learning Deep Structured Semantic Models for Web Search using Clickthrough Data. In **Proc. of the CIKM 2013**, 2013.

[3] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. A Deep Relevance Matching Model for Ad-hoc Retrieval. In **Proc. of the CIKM 2016**, 2016.

[4] Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. End-to-End Neural Ad-hoc Ranking with Kernel Pooling. In **Proc. of the SIGIR 2017**, 2017.

[5] Wei Yang, Kuang Lu, Peilin Yang, and Jimmy Lin. Critically Examining the "Neural Hype": Weak Baselines and the Additivity of Effectiveness Gains from Neural Ranking Models. In **Proc. of the SIGIR 2019**, 2019.

[6] Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. Okapi at TREC-3. In **Proc. of the TREC 1994**, 1994.

[7] Jimmy Lin. The Neural Hype, Justified! A Recantation. **ACM SIGIR Forum**, Vol. 53, pp. 88–93, 2019.

[8] Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. Pretrained Transformers for Text Ranking: BERT and Beyond. arXiv:2010.06467, 2020.

[9] Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. CEDR: Contextualized Embeddings for Document Ranking. In **Proc. of the SIGIR 2019**, 2019.

[10] Canjia Li, Andrew Yates, Sean MacAvaney, Ben He, and Yingfei Sun. PARADE: Passage Representation Aggregation for Document Reranking. arXiv:2008.09093, 2020.

[11] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. arXiv:1611.09268, 2018.

[12] Sebastian Hofstätter, Markus Zlabinger, and Allan Hanbury. TU Wien @ TREC Deep Learning '19 – Simple Contextualization for Re-ranking. arXiv:1912.01385, 2019.

[13] Bhaskar Mitra, Sebastian Hofstätter, Hamed Zamani, and Nick Craswell. Conformer-Kernel with Query Term Independence for Document Retrieval. arXiv:2007.10434, 2020.

[14] Cory Barr, Rosie Jones, and Moira Regelson. The Linguistic Structure of EnglishWeb-Search Queries. In **Proc. of the EMNLP 2008**, 2008.

[15] Kuzman Ganchev, Keith Hall, Ryan McDonald, and Slav Petrov. Using Search-Logs to Improve Query Tagging. In **Proc. of the ACL 2012**, 2012.

[16] Atsushi Keyaki and Jun Miyazaki. Part-of-speech Tagging for Web Search Queries Using a Large-scale Web Corpus. In **Proc. of the SAC 2017**, 2017.

[17] Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Yixing Fan, Xiang Ji, and Xueqi Cheng. PROP: Pre-Training with Representative Words Prediction for Ad-Hoc Retrieval. In **Proc. of the WSDM 2021**, 2021.

[18] Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. Latent Retrieval for Weakly Supervised Open Domain Question Answering. In **Proc. of the ACL 2019**, 2019.

[19] Wei-Cheng Chang, Felix X. Yu, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. Pre-training Tasks for Embedding-based Large-scale Retrieval. In **Proc. of the ICLR 2020**, 2020.

[20] Omar Khattab and Matei Zaharia. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. In **Proc. of the SIGIR 2020**, 2020.

[21] Nick Craswell, Daniel Campos, Bhaskar Mitra, Emine Yilmaz, and Bodo Billerbeck. ORCAS: 18 Million Clicked Query-Document Pairs for Analyzing Search. arXiv:2006.05324, 2020.

[22] Filip Radlinski and Thorsten Joachims. Query Chains: Learning to Rank from Implicit Feedback. In **Proc. of the KDD 2005**, 2005.

[23] Xuanhui Wang, Michael Bendersky, Donald Metzler, and Marc Najork. Intent Based Relevance Estimation from Click Logs. In **Proc. of the SIGIR 2016**, 2016.

[24] Ellen M. Voorhees. Overview of the TREC 2004 Robust Retrieval Track. In **Proc. of the TREC 2004**, 2004.

[25] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. In **Proc. of the ACL 2020**, 2020.

[26] Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. Multi-Stage Document Ranking with BERT. arXiv:1910.14424, 2019.

[27] Zeynep Akkalyoncu Yilmaz, Wei Yang, Haotian Zhang, and Jimmy Lin. Cross-Domain Modeling of Sentence-Level Evidence for Document Retrieval. In **Proc. of the EMNLP-IJCNLP 2019**, 2019.

[28] Zhuyun Dai and Jamie Callan. Deeper Text Understanding for IR with Contextual Neural Language Modeling. In **Proc. of the SIGIR 2019**, 2019.

[29] Qifan Wang, Yi Fang, Anirudh Ravula, Fuli Feng, Xiaojun Quan, and Dongfang Liu. WebFormer: The Web-page Transformer for Structure Information Extraction. In **Proc. of the WWW 2022**, 2022.

[30] Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. Document Expansion by Query Prediction. arXiv:1904.08375, 2019.

[31] Rodrigo Nogueira and Jimmy Lin. From doc2query to docTTTTTquery, 2019.

A　付録 (Appendix)

A.1　実験の設定と環境，予備実験

予備実験では ORCAS サンプリング率，coarse-tuning epoch 数，fine-tuning epoch 数の 3 項目に対してグリッドサーチで性能評価を行ったが，各項目に対して独立にパラメータチューニングを行った際の挙動と，他の項目を変化させた場合の挙動とは概ね同じであった．そのため，チューニング対象の項目以外はデフォルト値 (1) を用いた結果を掲載する．

- **mlm_prob**　MLM におけるマスク化されるトークンの割合は，BERT の pre-training にならい 0.15 とした．
- **qdpp_prob**　QDPP における学習データ生成において `isPair` のサンプルが生成される確率は，BERT の pre-training にならい 0.5 とした．
- **ORCAS サンプリング率**　ORCAS は膨大なクエリ–文書ペアを含むため，全データを用いた場合には学習が完了しなかった．そこで 1%–10% まで 1% 刻みでデータセットからクエリ–文書ペアのランダムサンプリングを行った．その結果，サンプリング率 5% の場合に最も高い検索性能を示した（表 3 参照）．
- **coarse-tuning epoch 数**　coarse-tuning では最大 5 epoch まで学習を行った．1 epoch 目で最も高性能を示した（表 4 参照）．
- **fine-tuning epoch 数**　fine-tuning では最大 10 epoch まで学習を行った．単独での実験では 1 epoch 目で最も高性能を示した（表 5 参照）が，他項目を変化させたときに 2 epoch 目で最高精度となる設定が多数を占めた．
- **バッチサイズ**　coarse-tuning では 80，fine-tuning では 128 を設定した．
- **開発**　システムの実装と評価には Hugging Face の Transformers[5] を用いた．
- **計算機**　スペックは CPU：AMD Ryzen 9 5900X 12-Core Processor，GPU：GeForce RTX 3060 VENTUS 2X 12G OC，メモリ：128GB である．

A.2　追加のクエリ推定結果

`coarse-tuned` から推定された「沖縄美ら海水族館」の記事[6] のクエリを表 6 に記載する．

5）https://huggingface.co/docs/transformers/index
6）https://www.okinawatraveler.net/en/feature/special01

表 3　ORCAS のサンプリング率と検索性能

サンプリング率	nDCG@20	P@20
1	0.283	0.269
2	0.305	0.288
3	0.301	0.268
4	0.290	0.267
5	**0.339**	**0.310**
6	0.323	0.303
7	0.293	0.271
8	0.321	0.296
9	0.307	0.283
10	0.318	0.298

表 4　coarse-tuning における epoch 数と検索性能

epoch 数	nDCG@20	P@20
1	**0.344**	**0.319**
2	0.299	0.269
3	0.293	0.279
4	0.293	0.271
5	0.287	0.282

表 5　fine-tuning における epoch 数と検索性能

epoch 数	nDCG@20	P@20
1	**0.344**	**0.319**
2	0.341	0.320
3	0.319	0.307
4	0.283	0.269
5	0.279	0.258
6	0.257	0.234
7	0.251	0.243
8	0.268	0.248
9	0.261	0.240
10	0.269	0.247

表 6　「沖縄美ら海水族館」の記事から推定されたクエリ

	Tok_{Q1}	Tok_{Q2}	Tok_{Q3}
Top1	sea	beach	beach
Top2	the	sea	islands
Top3	japan	islands	japan
Top4	hawaii	japan	island
Top5	japanese	island	hawaii