大規模言語モデルを用いた検索モデルの中間学習のためのコーパス作成手法

柴田幸輝 [1]　加藤誠 [2]　百合草陽介 [3]
[12] 筑波大学　[2] 株式会社ミスミグループ本社
[1]s2221646@u.tsukuba.ac.jp　[2]mpkato@acm.org　[3]yosuke.gjdt.yurikusa@misumi.co.jp

概要

商品検索などのドメイン特化の検索タスクにおいては，言語モデルが事前学習するような一般的なコーパスには無い，ドメイン独自の意味を持つ語が存在し，語彙の意味のギャップが存在することが懸念される．このようなドメイン間の語彙の意味のギャップを埋めるために，事前学習を用いた転移学習手法が提案されている．しかしながら，実務においては十分なサイズのコーパスを用意することが難しいような場合がある．その場合，事前学習を用いた転移学習手法の適用は難しい．そこで本研究では比較的小さな対象ドメインのコーパスが存在する場合に，言語モデルの MLM loss と二値分類モデルに基づいて汎用大規模コーパスから語彙の意味のギャップを埋めるようなコーパスを作成する手法を提案する．提案手法の性能を評価するため, 提案手法とベースライン手法を用いて複数のドメインを対象に大規模汎用コーパスからコーパスを作成し検索タスクにて実験を行った．実験を行った結果，全てのドメインで提案手法が高い性能を示した．

1　はじめに

商品検索などのドメイン特化の検索タスクにおいては，言語モデルが事前学習するような一般的なコーパスには無い，ドメイン独自の意味を持つ語が存在し，語彙の意味のギャップが存在することが懸念される．このようなドメイン間の語彙の意味のギャップを埋めるために，事前学習を用いた転移学習手法が提案されている．事前学習に基づいて言語モデルの転移を行う手法 [1, 2, 3, 4, 5] とは転移対象のドメインのコーパスを用いて言語モデルによる継続的な事前学習を実施する手法である．代表的な手法として，AdaLM[1] が挙げられる．

しかしながら，実務においては十分なサイズのコーパスを用意することが難しいような場合がある．その場合，事前学習を用いた転移学習手法の適用は難しい．この問題に対処するため，新たにドメイン固有のコーパスを作成する必要があると考えられる．コーパスの作成には，Web 文書や既存のコーパスを用いて，対象ドメインと類似している文書を取集する手法 [6] が提案されているが，収集した文書によっては，語彙が対象ドメインとは異なる意味で使われている文書を収集してしまうことも考えられ，語彙のギャップが拡大する可能性も存在する．

本研究では比較的小さな対象ドメインのコーパスが存在する場合に，言語モデルに基づいて汎用大規模コーパスから語彙の意味のギャップを埋めるようなコーパスを作成する手法を提案する．

提案手法において，初めに転移対象ドメインのコーパスを用いて，対象ドメインのコーパスと言語モデルが事前学習時に使用したコーパスとのギャップのある単語を推定し，その単語をクエリとして汎用的な大規模コーパスの文書を検索して文書を収集する．汎用的な大規模コーパスにて検索し文書を収集した後，各文書に対して転移対象と予測されるスコアを計算しスコアが高い文書を対象コーパスに加えることで，新たにコーパスを作成する．

提案するコーパス作成手法を評価するため, 提案手法とベースライン手法を用いて複数のドメイン (生物医学, 科学, E コマース) を対象に C4[7] データセットからコーパスを作成し実験を行った．実験ではそれぞれの手法を用いてコーパスを作成し，言語モデルに AdaLM を適用した後，検索タスクにおいて，ベースライン手法と提案手法を比較し，提案手法の有効性を検証した．実験を行った結果，全てのドメインで提案手法が高い性能を示した．

この論文における我々の貢献を次に示す：(1)MLM loss と二値分類モデルに基づいて汎用コーパスから語彙の意味のギャップを埋めるようなコーパスを作

成する手法を提案した．(2) 複数のドメインを対象に評価実験を行い，提案するコーパス作成手法の有効性を示した．

2 提案手法

本章では本研究が取り組む問題設定に関して説明し，本研究の問題に対してのアプローチを述べる．

本研究では転移対象のドメインのコーパスを対象コーパス D_t，コーパス作成時に文書を抽出するためのコーパスを参照コーパス D_s とする．この時，転移対象のドメインのコーパス D_t は数百から数千件の文書からなる小規模な文書集合であり，参照コーパス D_s は数千万から数億万件の文書からなる非常に大きな文書集合とする．本研究では入力として対象コーパス D_t，参照コーパス D_s, 対象ドメインに偏っていない汎用的な言語モデル m が与えられたとする．これらが与えられた時に参照コーパス D_s から文書を抽出することによって，対象ドメインに関連する文書を豊富に含む，新たなコーパス $D^\star$ を作成することを目的とする．

提案手法の概要をアルゴリズム 1 に示す．提案手法では汎用的な大規模コーパスから文書を検索することで文書を収集する．そのため，初めにコーパスから文書の検索時に，クエリとして用いるための単語の集合 W を作成する．単語集合 W は，対象コーパス D_t の文書に対して，言語モデル m を用いて計算した各語彙の MLM loss が高い語彙に基づいて単語集合 W を作成する (PrepareQueryWord(D_t, m)，アルゴリズム 1 行目)．単語集合 W を作成した後，単語集合 W の各単語 w をクエリとして参照コーパス D_s から文書を検索することで，単語集合 W 中の各単語 $w \in W$ に対応する文書集合 D_w を取得する (SearchDocument(W, D_s)，アルゴリズム 3 行目)．この際，各単語 w で文書を検索することで取得した文書集合 D_w に対して，二値分類器 g を用いて文書 $d \in D_w$ ごとに対象ドメインに当てはまると予測されるスコア s_d を求める (PredictDomainScore(g, d_i)，アルゴリズム 5 行目)．その後，文書集合 D_w の文書を対象ドメインに当てはまると予測されるスコア s_d が高い順に並び替えた上で，対象ドメインに当てはまると予測されるスコア s_d が高い上位 n 件の文書から，新たに単語 w に対応する D'_w を作成する (SortDocumentByScore(D_w, S, n)，アルゴリズム 8 行目)．その後各単語の文書集合 D'_w からコーパス D_{new} を作成する．最終的には新たに作成したコーパス D_{new} を対象コーパス D_t に加えていくことで新たにコーパス $D^\star$ を作成する．

2.1 検索に利用する語彙の選択

本節ではコーパスから文書の検索時に，クエリとして用いるための単語の集合 W を作成する手法の詳細について述べる．

本研究の目的は，語彙の意味のギャップを埋めるようなコーパスを作成することである．そのため，コーパスから文書の検索を行う際に，クエリとして用いる単語には意味のギャップがある単語であることが望ましい．意味のギャップのある単語の傾向として対象コーパスにおいて一般的な使われ方とは異なるような単語が考えられる．そのため，言語モデルに基づいて対象コーパス D_t から単語集合 W を作成する際は，初めに対象コーパス D_t にて言語モデル m の語彙集合 V_m 中の語彙 $v \in V_m$ に対する MLM loss を計算する．対象コーパス D_t にて言語モデルを用いて語彙 v の MLM loss を計算する際は，対象コーパス D_t から言語モデル m 内の語彙 v に対応する文書集合 D_v を取得する．文書集合 D_v は対象コーパス中の文書で，語彙 v を含んでいる文書のうち，10 件ランダムで抽出した文書集合である．またこの際，D_t の内，語彙 v の登場する文書の数が 10 件に満たないような語彙は計算対象から除いた．

対象コーパス D_t 中の，言語モデルの語彙 $v \in V_m$ の MLM Loss である $\text{loss}_{\text{mlm}}(D_t, v)$ は，文書集合 D_v を用いて次のように求められる．

$$\text{loss}_{\text{mlm}}(D_t, v) = -\frac{1}{n} \sum_{d \in D_v} \log P(v|d^{\text{Masked}}) \qquad (1)$$

$P(v|d^{\text{Masked}})$ は，文書 d に対して語彙 v が登場する箇所のうちランダムで一箇所だけマスクした上で，マスクされた位置に語彙 v が出現する確率である．D_t における単語 w の MLM loss である $\text{loss}_{\text{mlm}}(D_t, w)$ は次のように求められる．

$$\text{loss}_{\text{mlm}}(D_t, w) = \frac{1}{|V_w|} \sum_{v \in V_w} \text{loss}_{\text{mlm}}(D_t, v) \qquad (2)$$

V_w は単語 w を構成する言語モデル m の語彙の集合である．単語集合 W は転移対象のドメインのコーパス D_t に登場した各単語の MLM loss が高い上位 l 件の単語の集合である．

Algorithm 1 提案手法の概要

Require: m, D_t, D_s

$W \leftarrow \text{PrepareQueryWord}(D_t, m)$ ▷ 言語モデル m に基づいて検索クエリとして用いる単語を選択
for $w \in W$ **do**
  $D_w \leftarrow \text{SearchDocument}(W, D_s)$ ▷ 単語 w を検索クエリとして文書を検索
  for $d_i \in D_w$ **do**
    $s_i \leftarrow \text{PredictDomainScore}(g, d_i)$ ▷ 二値分類モデル g からスコアを計算
  end for
  $S \leftarrow (s_1, s_2, \ldots, s_n)$
  $D'_w \leftarrow \text{SortDocumentByScore}(D_w, S, n)$ ▷ スコアが高い n 件から新たな文書集合 D'_w を作成
  $D_{\text{new}} \leftarrow D_{\text{new}} \cup D'_w$ ▷ 文書集合 D'_w の文書を文書集合 D_{new} に追加
end for
$D^\star \leftarrow D_t \cup D_{\text{new}}$ ▷ 文書集合 D_{new} と文書集合 D_t の文書で文書集合 $D^\star$ 作成
return $D^\star$

3 語彙を用いた文書の検索

本節では各単語 $w \in W$ をクエリとしてコーパス D_s から文書を検索して単語 $w \in W$ に対する文書集合 D_w を取得する手法の詳細について述べる．

単語 $w \in W$ に対する文書集合 D_w を取得するには単語 $w \in W$ をクエリに BM25[8] のスコアを基準にコーパス D_s の文書を検索する．本研究では単語 $w \in W$ をクエリとしてコーパス D_s から各文書の BM25 のスコアを用いて検索して，上位 1,000 件の文書を単語 $w \in W$ に対する文書集合 D_w とした．

4 文書選択

本節では単語 $w \in W$ に対する文書集合 D_w から AdaLM を適用するためのコーパス $D_\star$ を作成する手法の詳細について述べる．単語 $w \in W$ に対する文書集合 D_w には単語 w を含んでいる文書のみが存在するが，単語 w が対象ドメインとは異なる意味で使われている文書を含む可能性がある．そのため，提案手法では文書集合 D_w をそのまま用いて $D^\star$ を作成せず，対象ドメイン中の文書と汎用ドメイン中の文書の二値分類を学習したモデル g を用いて，対象ドメインらしい文書を D_w から抽出して AdaLM を適用するためのコーパス $D_\star$ を作成する．

初めに文書集合 D_w から文書集合 D_{new} を作成する．コーパス D_{new} は次のように示される．

$$D_{\text{new}} = \bigcup_{w \in W} \{D'_w\} \tag{3}$$

文書集合 D'_w は D_w 中の文書のうち，二値分類モデル g が予測する，対象ドメインに分類されるスコアが高い上位 n 件の文書からなる文書集合である．

AdaLM を適用するためのコーパス $D_\star$ は対象コーパス D_t と D_{new} を組み合わせることで作成する．二値分類器には，対象コーパス D_t の文書と参照コーパス D_s とは異なる汎用的なコーパス D_g の文書の二値分類を学習させた．また，その他の二値分類の学習の詳細は付録に記載した．

5 実験設定

本説では初めに実験に使用したデータセットについて述べた後，実験手順に関して述べ，その後ベースライン手法を含む実験設定について紹介する．

本研究では，生物医学, 科学, E コマースの 3 つの異なるドメインで実験を行った．本研究では生物医学に TREC-COVID データセットを，科学に SciFact データセットを，E コマースに Amazon ESCI データセットを使用した．さらに，参照コーパス D_s として C4 コーパス [7] を使用した．

実験の手順を次に述べる．各手法でコーパスを作成した後，そのコーパスを用いて言語モデルに AdaLM を適用した．その後，言語モデルを用いた検索モデルを用いて検索用データセットにてファインチューニングを行い，転移対象のデータセットでランキングを作成し，ランキングの評価を行った．

我々は実験におけるベースライン手法として大規模コーパスからランダムで文書を選択する手法を設定した．この手法では参照コーパス D_s からランダムで文書を選択して対象コーパス D_t に加えることでコーパスを作成した．本手法では提案手法のコーパスサイズに合わせて作成した．また本研究では提案手法のうち，検索のみを用いた手法と二値分類のみを用いた手法とも比較を行った．検索のみとは提案手法のうち，検索を通じて収集した文書をそのまま利用してコーパス作成する手法である．二値分類のみとは二値分類モデルを用いてコーパス中のす

表 1 TREC-COVID での各コーパス作成手法の NDCG@10

手法	3,000 単語	6,000 単語	12,000 単語
AdaLM なし	0.597 (-)	-	-
ランダム	0.626 (624MB)	0.633 (1.2GB)	0.550 (2.3GB)
2 値分類のみ	0.595 (624MB)	0.581 (1.2GB)	**0.666** (2.3GB)
検索のみ	0.602 (990MB)	0.650 (1.8GB)	0.659 (3.3GB)
提案手法	0.626 (624MB)	0.652 (1.2GB)	**0.666** (2.3GB)

表 2 SciFact での各コーパス作成手法の NDCG@10

手法	3,000 単語	6,000 単語	12,000 単語
AdaLM なし	0.686 (-)	-	-
ランダム	0.661 (595MB)	0.653 (1.4GB)	0.671 (3.0GB)
2 値分類のみ	0.679 (595MB)	0.676 (1.2GB)	0.688 (3.0GB)
検索のみ	0.677 (1.2GB)	0.694 (2.2GB)	0.694 (4.2GB)
提案手法	0.685 (595MB)	0.696 (1.4GB)	**0.697** (3.0GB)

べての文書に対してスコア計算を行い，スコアが高かった文書を用いてコーパス作成する手法である．

ランキングの評価は，Iida ら [2] と同様 NDCG@10[9] を用いて評価した．その他の実験の詳細は付録に記載した．

6 実験結果

本節では実験結果について述べる．初めにベースラインとの比較を行い，その後対象ドメインと同じドメインから文書を抽出した結果について述べる．

6.1 ベースラインとの比較

表 1, 表 2, 表 3 に各コーパス作成手法によって作成されたコーパスを用いて AdaLM を適用した検索モデルの NDCG@10 の値を示す．これらの表のヘッダーは文書の検索の際，クエリとして利用した単語数を表記している．表中の NDCG@10 の値の横にはそれぞれ作成したコーパスのサイズを示している．AdaLM なしとは AdaLM を適用せずに，ファインチューニングのみを行った手法である．実験の結果，提案手法はその他の手法と比べて全体的に NDCG@10 の値が高い値を示した．この結果から，提案手法は特定のドメインでは有効であると考えられる．また，提案手法のコーパス作成手法のうち，文書の検索のみを使用した手法はランダムで文書を選択する手法と二値分類モデルのみを使用した手法と比べて TREC-COVID， SciFact にて NDCG@10 が高い値を示した．この結果から，コーパス内の文書

表 3 Amazon ESCI での各コーパス作成手法の NDCG@10

手法	3,000 単語	6,000 単語	12,000 単語
AdaLM なし	0.123 (688MB)	-	-
ランダム	0.110 (688MB)	0.116 (1.1GB)	0.112 (1.9GB)
2 値分類のみ	0.107 (688MB)	0.113 (1.1GB)	0.126 (1.9GB)
検索のみ	0.112 (927MB)	0.114 (1.5GB)	0.117 (2.6GB)
提案手法	**0.155** (688MB)	0.119 (1.1GB)	0.112 (1.9GB)

表 4 Pubmed コーパスを参照コーパスとした TREC–COVID での NDCG@10

手法	NDCG@10	コーパスサイズ
AdaLM なし	0.597	-
ランダム	0.626	1.4GB
2 値分類のみ	0.642	1.4GB
検索のみ (12,000 クエリ)	0.656	1.3GB
提案手法 (12,000 クエリ)	**0.660**	1.4GB

を選ぶ際には，収集する文書の語彙に着目することは重要であると考えられる．

6.2 対象ドメインと同じドメインコーパスから文書を抽出した結果

6.1 節では参照コーパスとして汎用コーパスである C4 コーパスを利用して実験を行ったが，参照コーパスとして対象ドメインと同じドメインのコーパスを利用した場合でも実験も行った．本研究では Iida らを参考に，生物医学ドメインで実験を行い，参照コーパスとして Pubmed コーパス[1)] 中の論文の要旨を使用した．表 4 に各コーパス作成手法によって作成されたコーパスを用いて AdaLM を適用した検索モデルの NDCG@10 の値を示す．実験の結果，提案手法はその他の手法と比べて全体的に NDCG@10 が高い値を示した．この結果から，対象ドメインとドメインが近いコーパスを参照コーパスとして利用した場合でも提案手法は有効であると考えられる．

7 結論

本研究では汎用大規模コーパスから語彙の意味のギャップを埋めるようなコーパス作成手法を提案した．提案手法の性能を評価するため, 提案手法とベースライン手法を用いて複数のドメインを対象に C4 データセットからコーパスを作成し実験を行った．実験の結果，全てのドメインでベースライン手法と比べて提案手法が高い性能を示した．

1) https://pubmed.ncbi.nlm.nih.gov/

謝辞

本研究は JSPS 科学研究費助成事業 JP22H03905, および，22H03905A による助成を受けたものです．ここに記して謝意を表します．また，論文執筆にあたりご助言をいただきました株式会社ミスミグループ本社の皆様に感謝に深く御礼申し上げます．

参考文献

[1] Yunzhi Yao, Shaohan Huang, Wenhui Wang, Li Dong, and Furu Wei. Adapt-and-Distill: Developing Small, Fast and Effective Pretrained Language Models for Domains. In **Findings of the Association for Computational Linguistics**, pp. 460–470, 2021.

[2] Hiroki Iida and Naoaki Okazaki. Unsupervised Domain Adaptation for Sparse Retrieval by Filling Vocabulary and Word Frequency Gaps. In **Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing**, pp. 752–765, 2022.

[3] Shitao Xiao, Zheng Liu, Yingxia Shao, and Zhao Cao. RetroMAE: Pre-Training Retrieval-oriented Language Models Via Masked Auto-Encoder. In **Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing**, pp. 538–548, 2022.

[4] Tao Shen, Xiubo Geng, Chongyang Tao, Can Xu, Xiaolong Huang, Binxing Jiao, Linjun Yang, and Daxin Jiang. LexMAE: Lexicon-Bottlenecked Pretraining for Large-Scale Retrieval. In **The Eleventh International Conference on Learning Representations**, 2022.

[5] 飯田大貴, 岡崎直観. 事前学習済みモデルに基づく検索モデルにおけるドメイン適応手法の比較と相乗効果の検証. 言語処理学会 第 29 回年次大会 発表論文集, 2023.

[6] 本浦庄太, 秋元康佑, 槇尾純太, 定政邦彦. 超大規模コーパスからの抽出コーパスによる言語モデルのタスク適応. 人工知能学会全国大会論文集 第 37 回, pp. 3Xin403–3Xin403, 2023.

[7] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. **The Journal of Machine Learning Research**, Vol. 21, No. 1, pp. 5485–5551, 2020.

[8] 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.

[9] 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.

[10] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. **the Journal of machine Learning research**, Vol. 9, pp. 1871–1874, 2008.

[11] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. **arXiv preprint arXiv:1611.09268**, 2016.

[12] 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.

[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] Sebastian Hofstätter, Sophia Althammer, Michael Schröder, Mete Sertkan, and Allan Hanbury. Improving Efficient Neural Ranking Models with Cross-Architecture Knowledge Distillation. **arXiv preprint arXiv:2010.02666**, 2020.

A 付録 (Appendix)

A.1 より詳細な実験設定

表 5 中間学習時のハイパーパラメータ

Parameter	Value
Batch size	10
Max document length	512
Learning rate	5×10^{-5}
Epoch	1
Warmup steps	0
Weight decay	0.0

表 6 ファインチューニング時のハイパーパラメータ

Parameter	Value
Batch size	32
Max document length	256
Learning rate	2×10^{-5}
Epoch	30
Warmup steps	1,000
Weight decay	0.01

本節では，より詳細な実験設定について述べる．

本研究では二値分類器 g には線形 SVM（サポートベクトル・マシン）を使用した [10]．二値分類器の学習は転移対象のドメインのコーパス D_t と汎用的なドメインのコーパス D_g の二値分類を学習させる．この際，汎用的なドメインのコーパス D_g では多種多様な文書が揃っており，文書のドメインが偏っていない必要がある．そのため二値分類器の学習には，汎用的なドメインのコーパス D_g として検索用のデータセットである MSMARCO-Document データセット [11] の文書コーパスを用いた．二値分類器の学習では転移対象のドメインのコーパスの文書のラベルを 1， MSMARCO-Document データセットの文書のラベルを 0 として訓練を行った．二値分類器の学習に用いる文書の特徴ベクトル $\mathbf{v_d}$ は，one-hot ベクトルを採用した．

本研究では Iida ら [2] に従って検索モデルとして SParse Lexical AnD Expansion(SPLADE)[12] を利用した．SPLADE はクエリと文書を疎な埋め込みで表現し，最終的にはこれらのクエリ，文書の疎な埋め込みの内積を取ることでにクエリ，文書の適合度を予測する手法である．

ファインチューニングには TREC-COVID データセットと SciFact データセットの転移学習時は Iida[2] に従って MSMARCO[11] データセットでファインチューニングを行った．Amazon ESCI データセットでの転移学習時は Amazon ESCI データセットの Train データを用いてファインチューニングを行った．

AdaLM での語彙拡張では Iida[2] らの実験設定を参考に，拡張する語彙数は 40,000 語彙とした．また，AdaLM での中間学習では distilbert-base-uncased[13][2] をベースモデルとして Masked Language Modeling を行った．Masked Language Modeling でのハイパーパラメータを表 5 に示す．Masked Language Modelling では最大入力長 512 トークン Batch サイズは 10，Learning rate:$5 * 10^{-5}$，Warmup steps:0，Weight decay:0.0 で 8 つの NVIDIA A100 40 GB を用いて 1 エポックで学習をおこなった．

SPLADE の損失関数は Iida を参考に Margin-MSE[14] と FLOPS 正則化の和とした．クエリ側の FLOPS の正則化重み λ_Q とドキュメント側の λ_D は，それぞれ $\lambda_Q = 0.08, \lambda_D = 0.1$ と設定した．Margin-MSE で使用されるハードネガティブは，MSMarco のデータセットを用いる場合は Iida ら [2] に従って，BM25 や他の検索方法によって文書を検索することで収集した文書ハードネガティブとした．ESCI データセットを用いる場合，ESCI データセットにおいては適合度は Exact, Substitute, Complement, Irrelevant の 4 段階のラベルが存在するため，このラベルのうち一番上の Exact をポジティブとして，他 Substitute, Complement, Irrelevant の 3 ラベルをハードネガティブとした．また Iida らの研究 [2] ではベースの言語モデルとして bert-base-uncased[3] を用いている．本研究では distilbert-base をベースの言語モデルとしているため，本研究で使用するベースモデルはパラメーター数が半分ほどとなっている．モデルのオーバーフィッティングを防ぐために本研究では訓練データのうち 2 割を検証データとして，early stopping を採用した．SPLADE のファインチューニングでは入力長 256 トークン Batch サイズは 10， Learning rate:$2 * 10^{-5}$， Warmup steps:1,000, Weight decay:0.01 で 4 つの NVIDIA V100 16 GB GPU で 30 エポックで学習をおこなった．

2） https://huggingface.co/distilbert-base-uncased
3） https://huggingface.co/bert-base-uncased