情報処理学会論文誌　データベース　Vol.15　No.3　20–35 (Oct. 2022)

Zero-shotニューラル検索のための語彙一致と文脈の類似度による関連度スコアリング

飯田 大貴[1,a)]　岡崎 直観[1,b)]

受付日 2022年3月10日, 採録日 2022年4月28日

概要：BERT をエンコーダとしてクエリと文書を密ベクトルで表現し，その類似度を関連度スコアとする密ベクトル検索は，BM25 などの語彙一致をベースとした検索アルゴリズムを大きく上回る性能を示している．しかし，密ベクトルに変換するエンコーダを訓練するためには，訓練データとして大量のクエリと適合文書の正解ペアが必要となる．企業内の文書検索などは，アノテーションコストからこの正解ペアを作成することが困難であり，検索システムを使用するドメインで訓練データが存在しない Zero-shot と呼ばれる場合で，検索性能を向上させることが求められる．そこで，我々は，Zero-shot 時に検索性能を向上させるスコアリング方法として，Contextualized BM25（C-BM25）を提案する．C-BM25 は，クエリと文書で語彙が一致したトークンごとに文脈の類似度を計算し，BM25 をトークンの重要度として，類似度と重要度の重み和を関連度スコアとする．我々は，Zero-shot 設定の検索ベンチマークデータセットである BEIR を用いて実験を行い，BM25 と比較して約 20%の性能改善をした．本手法を用いることで，Zero-shot でもエンコーダが学習した文脈情報を検索で有効に利用することが可能である．

キーワード：Zero-shot 検索，ニューラル検索，文脈化語彙一致検索，密ベクトル検索

Relevance Scoring with Lexical Match and Context Similarity for Zero-shot Neural Retrieval

Hiroki Iida[1,a)]　Naoaki Okazaki[1,b)]

Received: March 10, 2022, Accepted: April 28, 2022

Abstract: Dense retrieval, in which queries and documents are encoded to dense vectors through BERT and the similarity of these vectors is used as relevance scores, has shown significantly better performance than ranking algorithms based on lexical matching, such as BM25. However, to train the encoder which convert queries and documents to dense vectors, we need a large amount of pairs of queries and matching documents to the queries as training data. In some cases like enterprise search, it is difficult to create such training data due to the annotation cost, and it is necessary to improve performance in what is called a "zero-shot" case where no training data exists in the target domain. Thus, we propose Contextualized BM25 (C-BM25) as a scoring method to improve retrieval performance in the case of zero-shot. The scoring of C-BM25 is weighted sum of the BM25 importance for the tokens and the context similarity of lexically matching tokens between a query and a document. We experiment with BEIR, a benchmark dataset for zero-shot settings, and show that C-BM25 improves the performance by about 20% compared to BM25. Our method enables us to utilize context information learned by the encoder for information retrieval even with zero-shot case.

Keywords: Zero-shot retrieval, neural retrieval, contextualized lexical retrieval, dense retrieval

[1] 東京工業大学
Tokyo Institute of Technology, Meguro, Tokyo 152–8550, Japan
[a)] hiroki.iida@nlp.c.titech.ac.jp
[b)] okazaki@c.titech.ac.jp

1. はじめに

BERT [6] [*1] などの事前学習済み言語モデルをエンコーダとして，クエリと文書を密ベクトルで表現し，そのベクトルの内積やコサイン類似度を関連度スコアとする密ベクトル検索（Dense Retrieval）は，BM25 [30] などの語彙一致検索（Lexical Retrieval）を大きく上回る性能を示している [13], [16], [20], [40], [43].

しかし，密ベクトルに変換するエンコーダを訓練するためには，訓練データとしてクエリと適合文書からなる正解ペアが大量に必要である．また，検索において，対象ドメインで訓練データを用いることができない場合は多く見られる．実際，企業内検索や専門家向けの検索においては，アノテーションコストの問題から，訓練データとなるクエリと適合文書のペアを作成できない場合がある．また，代替手段として，検索システムの対象ドメインと異なるドメインで訓練した密ベクトル検索を使用すると，大幅に性能が劣化し BM25 に劣る結果となる場合がある [32]．このように，対象ドメインで訓練データが存在しない場合を Zero-shot と呼ぶ [32] が，実応用において Zero-shot で検索性能を向上させることが求められる場合がある．

Zero-shot 時に，密ベクトル検索がドメインの変化に弱い一因として，低頻度語や訓練時にはなかった単語の重要度を低くとらえているという問題が指摘されている [8]．その結果，クエリ中の単語が含まれない文書の方が上位に出現する場合がある．よって，Zero-shot 時は，語彙一致と検索コーパス中の低頻度語への重み付けによって，検索上位の文書が，より確実にクエリ中の単語を含むようにすれば良いと考えられる．さらに，その一致した単語が出現する文脈の類似度を用いることで，密ベクトル検索のエンコーダが学習した文脈情報を活用し，検索性能を向上することができると考えられる．

そこで，このようなスコアリングを実現する方法として，Contextualized BM25（C-BM25）を提案する．C-BM25 は，クエリと文書で一致する単語，すなわち語彙が一致するトークンごとに文脈の類似度を計算し，BM25 をトークンの重要度として，類似度と重要度の重み和を関連度スコアとする方法である．また，このようにクエリと文書で一致した単語の文脈類似度を用いる検索方法を，文脈化語彙一致検索（Contextualized Lexical Retrieval）[9] と呼ぶ．

本論文ではまず，C-BM25 が Zero-shot 時に広く有効であることを確認するために，Zero-shot の検索ベンチマークデータセットである BEIR [32] において，BM25 の上位 100 件を並べ替えるリランキングタスクを実施した．検索データセットである MS MARCO [26] にて訓練を行った密ベクトル検索のエンコーダを用いて，検索性能の評価指標である nDCG@10 で評価した．結果，BM25 や密ベクトル検索を上回るだけでなく，他の Zero-shot 時に有効な手法を上回るまたは同等な値となった．また，広く検索タスクのベンチマークデータセットとして用いられている Robust04 [35] にて，C-BM25 が文書全体の検索を実用的な遅延時間で検索を行えることを確認した．

さらに，C-BM25 における各要素の影響分析を行い，C-BM25 が BM25 によるトークンへの重み付けなしで BM25 を上回ることを確かめた．そして，事例を用いて，文脈化語彙一致検索が Zero-shot での検索でなぜ有効であるのかを単語頻度の観点から考察した．

まとめると，本論文の貢献は以下である．

- Zero-shot 時でも低頻度語を重視し，文脈情報を利用できるスコアリング方法として C-BM25 を提案した．
- C-BM25 が，Zero-shot 時の検索で有効であることを示した．また，応用上妥当な遅延時間で対象文書全体を検索できることを示した．
- Zero-shot での検索において，文脈化語彙一致検索が有効な理由を単語頻度の観点から考察した．

本論文は以下のように構成される．2 章では，関連研究について述べる．3 章では，提案手法について述べる．4 章では，有効性を確かめるための実験設定について述べる．5 章では，実験結果について述べる．6 章では，提案手法を構成する各要素の影響分析を行う．7 章では，事例を用いた本手法の傾向について述べる．

2. 関連研究

2.1 ニューラルネットワークを用いた検索

ニューラルネットワークを用いた検索（Neural Retrieval）に関する研究は，BERT が提案される以前から多数提案されていた [11], [14], [29]．中でも，Guo ら [11] は，検索タスクにおけるクエリと文書のマッチング要因を指摘し，それをニューラルネットワークの構成に組み込んだ．C-BM25 は，その要因を考慮したスコアリング方法と見なせる．詳細は，3.4 節にて述べる．

BERT が提案された後，当初は，クエリと文書を結合し，同時に BERT に入力とするクロスエンコーダが使用されていた [23], [27]．しかし，この手法は計算コストが高いため，検索結果の上位を並べ替えるリランキングのみに使用されていた．その後，対象文書全体を検索できるモデルとして，密ベクトル検索 [13], [16], [20], [40], [43] が提案され

[*1] BERT とは，Bidirectional Encoder Representations from Transformers と呼ばれる手法の通称であり，Transformer というニューラルネットワークのユニットを利用して学習した，テキストをベクトルに変換するエンコーダである．BERT は，大規模なコーパスで事前学習を行うことで，幅広い自然言語処理タスクにおいて良好な性能を示すことが知られている．事前学習は，2 種類の自己教師あり学習によって行われる．1 つは，テキストの一部を [MASK] と呼ばれるトークンに置換し，置換前のトークンをあてる Masked Language Model というタスクである．もう一方は，2 つのテキストを結合して入力し，その 2 つが連続するかそうでないかをあてる Next Sentence Prediction というタスクである．

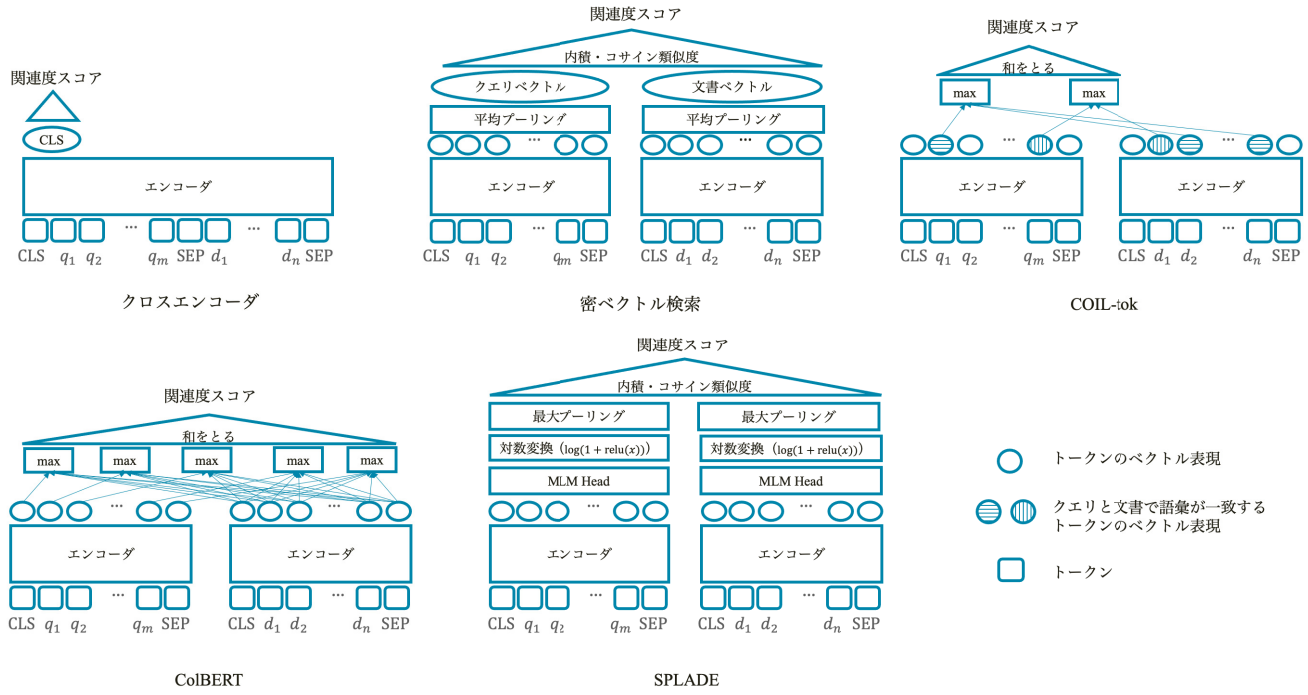


図 1　BERT を用いた既存のニューラル検索手法の模式図
Fig. 1　An illustration of neural retrieval methods with BERT.

た．しかし，密ベクトル検索は，使用するドメインが訓練したデータセットと異なる場合に，多くのデータセットでBM25 に劣るということが指摘されている [32]．本研究は，密ベクトル検索のエンコーダをトレーニングしたドメイン以外でも有効に使用する手法を提案するものである．

C-BM25 と同様に，文脈化語彙一致検索の手法として，COIL-tok [9] がある．COIL-tok は，エンコーダを用いてクエリと文書のトークンをベクトルで表現し，クエリと文書で語彙が一致したトークンごとに，トークンを表現するベクトルの内積で文脈の類似度をとる．そして，その類似度の和を関連度スコアとする．本論文では，C-BM25 がCOIL-tok を上回ることを確認する．また，その要因について，6.1 節と A.1 節で考察している．

これら以外の手法として，ColBERT [17] や SPLADE [7] が提案されている．ColBERT はクエリの各トークンのベクトルごとに，最も類似する文書のトークンのベクトルとの内積をとり，それらの和を検索結果の関連度スコアとする手法である．SPLADE は，クエリと文書をエンコーダの語彙次元のベクトルに変換し，それらの内積を関連度スコアとする手法である．ともに，Zero-shot での検索性能向上を目指したものではなかったが，Zero-shot 時の検索性能が BM25 を上回ることが示されている．そのため，本論文ではこれらとの比較を行い，C-BM25 の有効性を明らかにする．

2.2　Zero-shot のニューラルネットワーク検索

Zero-shot におけるニューラル検索は，現在のニューラル検索の主たるテーマの 1 つである．

Ma ら [22] は，Web から質問と回答のペアを収集し，それを用いて疑似クエリ生成器を学習し，その生成器で検索対象文書から生成したクエリで正解ペアを作り，密ベクトル検索のエンコーダを訓練した．Ma らはさらに，密ベクトル検索の関連度スコアと BM25 など語彙一致検索の関連度スコアを足し合わせることで，Zero-shot 時の検索精度を向上させた．このように異なる検索方法の関連度スコアを足し合わせる方法を，本研究では，ハイブリッド検索（Hybrid Retrieval）と呼び，これとも比較を行う．なお，生成した疑似クエリを使用して密ベクトル検索用のエンコーダをさらに訓練すると，密ベクトル検索は多数のデータセットの性能指標の平均において，劣化することがThakur ら [32] によって報告されているため，本研究では，疑似クエリ生成による訓練は実施していない．

Zero-shot 時の検索における密ベクトル検索のエンコーダの改善として，Xin ら [39] は，ドメイン敵対的訓練を用いて Zero-shot 時の密ベクトル検索の精度向上を行った．また，Izacard ら [15] は，文書の一部から語彙一致が一致するように取り出したスニペットのペアを正例として対照学習を行うことで Zero-shot 時の密ベクトル検索の精度向上を行った．これらはベクトル表現自体を改善するものであり，本研究とは補完的関係にあると考えられる．

3.　提案手法

本研究の目的は，Zero-shot 時の検索性能向上をスコアリングで実施できることを示すことにある．Zero-shot 時

の密ベクトル検索の問題点として，低頻度語や訓練時にはなかった単語の重要度を低くとらえているという問題が指摘されている [8]．その結果，クエリで指定された単語が含まれない文書の方が上位に出現する場合がある．一方で，クエリ中の単語が上位文書に含まれるように，BM25 などの語彙一致による検索を行うと，密ベクトル検索のように文脈をとらえた検索を行うことができない．

そこで，Zero-shot 時でも，クエリ中の単語が含まれるようにしつつ，文脈をとらえた検索が行えるようにすることで，検索性能を上げられると考えられる．このような検索を行う方法として文脈化語彙一致検索（Contextualized Lexical Retrieval）[9] がある．文脈化語彙一致検索は，クエリと文書で一致した単語，つまり語彙が一致したトークンごとに文脈の類似度を算出し，それを関連度スコアに用いる方法である．これに加えて，BM25 のスコアでトークンを重み付けることによって，検索対象コーパス中で低頻度な単語を重要視することが可能となる．このように，BM25 でトークンを重み付ける文脈化語彙一致検索の手法として，Contextualized-BM25（C-BM25）を提案する．

さらに，C-BM25 の関連度スコアと密ベクトル検索の関連度スコアを足し合わせる，ハイブリッド C-BM25（HC-BM25）を提案する．C-BM25 は，クエリと文書で語彙が一致したトークンの文脈類似度を計算するため，クエリと文書の一部がマッチングした場合と考えられる．HC-BM25 は，密ベクトル検索による関連度スコアを，クエリと文書全体のマッチングととらえ，これを加味することで，さらなる性能向上を目指したものである．

本章ではまず，BM25 と既存の文脈化語彙一致検索である COIL-tok を説明する．次に提案手法である，C-BM25 について説明する．最後に，HC-BM25 について説明する．

なお，以下では，検索対象文書全体を $\mathcal{D}$，語彙集合を $\mathcal{V}$ と表し，クエリ Q が m 個のトークンの列からなるとし，$Q = (q_1, q_2, \ldots, q_m) \in \mathcal{V}^m$ と表す．文書 D は n 個のトークンの列からなり，$D = (d_1, d_2, \ldots, d_n) \in \mathcal{V}^n$ と表す．クエリ，文書の各トークンはともに BERT などの言語モデルを使用して l 次元ベクトルにエンコードされる．エンコードしたクエリ Q を $\boldsymbol{Q} = (\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_m) \in \mathbb{R}^{l \times m}$，エンコードした文書 D を $\boldsymbol{D} = (\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_n) \in \mathbb{R}^{l \times n}$ と表す．

3.1　前提となる既存手法

3.1.1　BM25

BM25 は，現在最も広く使用されている検索のランキングアルゴリズムである．クエリと文書に出現する語彙が一致するトークンをベースにスコアリングを行う方法であり，語彙一致による検索アルゴリズムとして分類される．BM25 は，トークン $t \in \mathcal{V}$ について，文書 D での頻度 $\mathrm{TF}(t, D)$ と，トークン t を含む文書集合 $\mathcal{D}_t$ の検索対象コーパス $\mathcal{D}$ における逆頻度 $\mathrm{IDF}(t)$ をベースしたスコアリングであり，以下で関連度スコア $S_{\mathrm{BM25}}(Q, D)$ が計算される．

$$S_{\mathrm{BM25}}(Q, D) = \sum_{i=1}^{n} \mathrm{BM25}(q_i, D), \tag{1}$$

$$\mathrm{BM25}(t, D) = \frac{\mathrm{IDF}(t)\mathrm{TF}(t, D) \cdot (1 + k_1)}{\mathrm{TF}(t, D) + k_1\{1 + b(\frac{n}{\mathrm{DA}} - 1)\}}, \tag{2}$$

$$\mathrm{IDF}(t) = \begin{cases} \log(|\mathcal{D}|/|\mathcal{D}_t|) & |\mathcal{D}_t| \neq 0 \\ 0 & |\mathcal{D}_t| = 0 \end{cases} \tag{3}$$

ここで，k_1 と b は，BM25 のハイパーパラメータである．また，DA は，$\mathcal{D}$ 中の文書の文書長の平均である．なお，本論文では式 (2) をトークンの BM25 スコアと呼ぶ．また，式 (3) をトークンの IDF スコアと呼ぶ

3.1.2　COIL-tok

COIL-tok は，語彙一致が持つ解釈性・操作可能性を残しつつ，TF などのヒューリスティクスより効果的なマッチングを実現するために文脈を考慮することを目指して，Gao ら [9] により提案された，文脈化語彙一致検索の手法である．文脈を考慮するために，BERT などの学習済み言語モデルによってトークンをエンコードしたベクトルを使用している．具体的には，クエリと文書で語彙が一致するトークンごとに，そのエンコードしたベクトル表現どうしの内積を計算し，その内積の和を関連度スコアとしている．COIL-tok の関連度スコアを $S_{\text{COIL-tok}}(Q, D)$ とすると，

$$S_{\text{COIL-tok}}(Q, D) = \sum_{i=1}^{n} \max_j (\delta(q_i, d_j) \cdot \boldsymbol{q}_i^\top \boldsymbol{d}_j). \tag{4}$$

と表される[*2]．ここで，$\delta(q_i, d_j)$ は，クエリと文書のトークンの語彙が一致する場合，つまり $q_i = d_j$ の場合に 1 をとる関数であり，

$$\delta(q_i, d_j) = \begin{cases} 1 & q_i = d_j \\ 0 & \text{それ以外の場合} \end{cases} \tag{5}$$

としている．これによって，クエリ中の各トークンのスコアは，文書のトークンと語彙が一致した場合は，その語彙のトークンを表現するベクトルとの内積の最大値となり，一致する語彙がない場合は，0 となる．また，内積の最大値を利用することで，最も重要な信号に焦点を当てている．なお，検索タスク用に言語モデルを訓練する際も同様のスコアリングを用いている．

3.2　Contextualized-BM25

Zero-shot のニューラル検索では，検索対象文書で低頻度な語や訓練用の正解ペアに存在しない語の重要度が低くなることが指摘されている．そのため，クエリ中のトークンで指定される語彙を含む文書より，含まない文書が

[*2] 実際には，インデクスサイズ削減のため，トークンをエンコードしたベクトル $\boldsymbol{q}_i$，$\boldsymbol{d}_j$ に対して線形変換を施して次元を小さくしている．

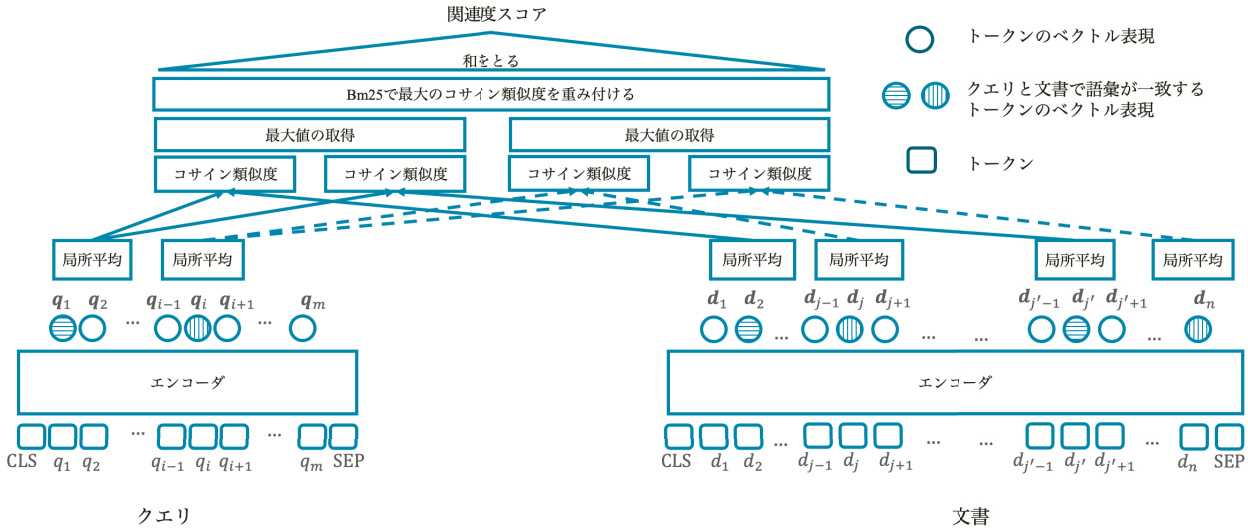


図 2　C-BM25 の概要図
Fig. 2　An illustration of C-BM25.

上位になることがある．COIL-tok のような文脈化語彙一致検索は，語彙一致を基本としているため，文脈を考慮しつつ，クエリ中の検索対象文書中で低頻度なトークンを含む文書が上位に来ることが期待できる．さらに，BM25 でトークンを重み付けることによって，よりその期待が高まる．このようなスコアリングを実現する方法として，Contextualized-BM25（C-BM25）を提案する．クエリと文書で語彙が一致するトークンごとに文脈類似度を計算し，その類似度をトークンの BM25 スコアで重み付けて和をとることで関連度スコアを算出する．C-BM25 の関連度スコア $S_{\text{C-BM25}}(Q, D)$ は，

$$S_{\text{C-BM25}}(Q, D) = \sum_{i=1}^{n} \text{BM25}(q_i, D) \max_j (\delta(q_i, d_j) \cdot \cos(\boldsymbol{h}_{q_i}, \boldsymbol{h}_{d_j})) \quad (6)$$

である．COIL-tok 同様，クエリの各トークンごとに最大の類似度を使用している．COIL-tok は TF などの重みも含むことを意図しているため内積を用いているが，我々は重みは BM25 によって実現できているとして，文脈類似度を計算するため，文類似度タスク [3] の実践に倣いコサイン類似度を用いている．コサイン類似度はベクトル $\boldsymbol{u}$, $\boldsymbol{v}$ に対し，

$$\cos(\boldsymbol{u}, \boldsymbol{v}) = \frac{\boldsymbol{u}^\top \boldsymbol{v}}{\|\boldsymbol{u}\| \|\boldsymbol{v}\|} \quad (7)$$

で計算される．

$\boldsymbol{h}_{q_i}$ と $\boldsymbol{h}_{d_j}$ は，トークンの文脈を表現するベクトルである．その際，語彙が一致したトークン周辺の単語も活用するため，周辺のトークンのベクトルで平均を計算する．今，$q_i = d_j$ とすると，式としては以下である．

$$\boldsymbol{h}_{q_i} = \frac{1}{B} \sum_{k=i-o}^{i+o} \boldsymbol{q}_k, \quad (8)$$

$$\boldsymbol{h}_{d_j} = \frac{1}{B} \sum_{k=j-o}^{j+o} \boldsymbol{d}_k. \quad (9)$$

o は窓幅であり，$B = 2o + 1$ である．また，このように窓幅を用いた平均を本論文では局所平均と呼ぶ．

C-BM25 は，COIL-tok 同様にクエリと文書に共通して現れるトークンごとにスコアを計算するため，本論文では文脈化語彙一致検索の手法と位置付ける．

3.3　ハイブリッド C-BM25

C-BM25 では，文脈類似度としてトークンごとに最大のコサイン類似度を使用しているため，最も一致した箇所のみがスコアに反映される．そのため，クエリと文書全体が適合していることをスコアに反映することで，より性能向上が見込まれる．そこで，密ベクトル検索の関連度スコアをクエリと文書全体のマッチングと見なし，これを利用する方法を提案する．具体的には，密ベクトル検索の関連度スコア $S_{\text{Dense}}(Q, D)$ と C-BM25 の関連度スコア $S_{\text{C-BM25}}(Q, D)$ を足し合わせる，ハイブリッド C-BM25（HC-BM25）を提案する．HC-BM25 のスコア $S_{\text{HC-BM25}}(Q, D)$ は

$$S_{\text{HC-BM25}}(Q, D) = S_{\text{C-BM25}}(Q, D) + S_{\text{Dense}}(Q, D), \quad (10)$$

$$S_{\text{Dense}}(Q, D) = \cos\left(\frac{1}{m}\sum_{i=1}^{m} \boldsymbol{q}_i, \frac{1}{n}\sum_{j=1}^{n} \boldsymbol{d}_j\right) \quad (11)$$

と計算される．密ベクトル検索の関連度スコアと他の関連度スコアを足し合わせる点において，Ma ら [22] が実施しているハイブリッド検索と類似しているため，本論文では HC-BM25 をハイブリッド検索の一部であると位置付ける．

3.4　検索タスクにおけるマッチング要因と提案手法の関係

C-BM25 および HC-BM25 は，Guo ら [11] が指摘した，検索タスクにおけるクエリと文書のマッチング要因を考慮したスコアリング手法と見なせる．Guo らは要因として，クエリと文書の単語の一致，クエリ中の単語の重要度判別，多様なマッチング要件の 3 点を指摘している．多様なマッチング要件とは，クエリと文書の全体がマッチングする場合とクエリと文書の一部のみがマッチングする場合の両方があるとしている．C-BM25 は，文脈化語彙一致検索を使用しているため，クエリと文書の語彙が一致するトークンを重視しつつ，文脈類似度によってクエリと文書の一部のみがマッチングしている場合に対応する．加えて，BM25 によるトークンの重み付けによって，クエリ中のトークンの重要度を判別している．また，HC-BM25 は，密ベクトル検索の関連度スコアを利用し，クエリと文書全体がマッチングした場合についても考慮している．

4.　実験設定

本章では，C-BM25 の有効性を示す実験に用いるデータセットと比較に用いたベースライン手法について述べる．また，C-BM25 やベースライン手法に用いるエンコーダとその訓練方法について合わせて述べる．本手法の実装は，https://github.com/nlp-titech/LSS_FUNC にて公開している．

4.1　データセット

本論文では，C-BM25 が広く多様なデータセットで有効であることを示すため，BEIR [32] を用いる．BEIR は，Zero-shot 時の検索ベンチマークデータセットであり，既存の検索ベンチマークデータセットをまとめたものである．C-BM25 の有効性は，リランキングタスクによって検証する．リランキングタスクとは，あらかじめ BM25 などの高速な検索アルゴリズムを用いて，上位 k 件を取り出し，その k 件を並べ替えるタスクである．今回は，Thakur ら [32] にならい，$k = 100$ とした．なお，BM25 の関連度スコアは上位 100 件を検索した際と同じものを使用する．性能指標も Thakur らと同様に，nDCG@10 を使用する．

次節にてベースライン手法を述べるが，BM25 を除くすべての手法においてモデルの訓練を実施する．その訓練には検索タスクのデータセットである MS MARCO [26] を用いる．BEIR は MS MARCO を含むため，Zero-shot 時のリランキング性能評価では，MS MARCO を除いている．

一方，対象文書全体に対してスコアリングを行う検索タスクについては，Robust04 [35] を用いて有効性を示す．Robust04 は ad-hoc 検索にて広く用いられているデータセットである．クエリには，title という短いクエリと descripotin という長いクエリがあるが，title をクエリとして使用する．

BEIR において使用するデータセットは，Formal ら [7] 同様に Thakur ら [32] が web 配布しているデータセット[*3]と，Robust04 を用いる．すべてのデータセットで test とされているものを用いた．BEIR 中で用いる Robust04 は，desctiption をクエリとして用いているため，リランキングの実験では description をクエリとしている．

4.2　ベースライン手法

ベースライン手法として，前述の BM25 と COIL-tok に加えて，密ベクトル検索を用いる．クエリと文書を表現するベクトルを作成する方法として，トークンベクトルの平均とトークンベクトルの重み付き平均を用いる．さらに，密ベクトル検索を用いる手法として，ハイブリッド検索をベースラインとする．

加えて，他の Zero-shot 検索で有効な手法との比較として，クロスエンコーダ [27]，ColBERT [17]，SPLADE [7]，の実験結果を記載する．

4.2.1　BM25

BM25 は，語彙一致を基本とした関連度スコア算出方法である．現在，最も広く使われるランキングアルゴリズムの 1 つであるため，ベースラインとした．関連度スコアは，式 (2) で計算される．

4.2.2　COIL-tok

COIL-tok は，クエリと文書で語彙が一致したトークンごとに文脈類似度を計算し，それを関連度スコアとして用いる文脈化語彙一致検索の手法である．関連度スコアは，式 (4) で表される．C-BM25 と同様に文脈化語彙一致検索の方法であるため，ベースラインとした．

4.2.3　密ベクトル検索

密ベクトル検索はクエリと文書をベクトルで表現し，内積またはコサイン類似度で検索結果を順序づける検索手法である．本研究では，コサイン類似度を用いている．ベクトル表現の作成方法としては，クエリと文書の各トークンをエンコードしたベクトルの平均を用いる**平均ベクトル**と，各トークンについて BM25 のスコアで重み付けを行い，平均をとった**重み平均ベクトル**を用いている．Formal ら [8] は，密ベクトル検索において，低頻度語の重要度が過少評価されているという問題を指摘しているため，BM25 で重み付けることによってこれを補えることが期待できる．そのため，平均ベクトルに加えて，重み平均ベクトルもベースラインとした．なお，重み平均ベクトルでは，重みの和でベクトルを割ることで，正規化している．

4.2.4　ハイブリッド検索

ハイブリッド検索（Hybrid Retrieval）は，密ベクトル

[*3] つまり，DBPedia [12], FiQA [24], Natural Question（NQ）[18], HotPotQA [42]，NFCorpus [2]，TREC-COVID [34]，Touché-2020 [1]，ArguAna [36]，Climate-FEVER [19]，FEVER [33]，Quora，SCIDOCS [4]，SciFact [37] である．

検索と語彙一致検索のそれぞれのスコアを用いるものである．語彙一致検索のアルゴリズムとしては，BM25 が用いられることが多い．また，密ベクトル検索と語彙一致検索が相互補完的であることが，Gao ら [10] や Luan ら [21] で示されている．本論文では，Zero-shot の場合であるため，ハイブリッド検索のベースラインとして，Ma ら [22] 同様，BM25 のスコアと密ベクトル検索のスコアの和を使用した．密ベクトルの表現方法は平均ベクトルを使用した．これを本論文では**ハイブリッド BM25**（**H-BM25**）と呼ぶ．

4.2.5　クロスエンコーダ

クロスエンコーダ（CrossEncorder: CE）はクエリと文書の間に文末トークン（BERT の場合は [SEP]）を入れて結合し，学習済み言語モデルに入力し，関連の有無を 2 値の分類タスクとして学習する方法である．関連度スコアは，モデルが出力するロジットの値を使用している．Thakur ら [32] によると，BEIR においてリランキングの場合，nDCG@10 で最も良い値を示している．そのため，性能の参考値としてベースラインとした．

4.2.6　ColBERT

ColBERT [17] は，クエリの各トークンのベクトルと最も類似する文書のトークンのベクトルとの内積の和を関連度スコアとする手法であり，クロスエンコーダが使用されていた際の高速化方法として提案された．ColBERT の関連度スコア S_{ColBERT} は，式としては以下のとおりである．

$$S_{\mathrm{ColBERT}}(Q, D) = \sum_{i=1}^{n} \max_{j}(\boldsymbol{q}_i, \boldsymbol{d}_j) \tag{12}$$

クロスエンコーダと異なり，文書のトークンをあらかじめエンコードできる．そのため，検索時はより短いクエリのエンコードのみでよいため，クロスエンコーダよりも高速に実行可能である．Thakur ら [32] が示したとおり，ColBERT は，BEIR において，多くのデータセットで BM25 を nDCG@10 で上回る方法である．そのため，性能の参考値としてベースライン手法とした．

4.2.7　SPLADE

SPLADE [7] は，クエリと文書を語彙次元のベクトルで表現し，その内積を関連度スコアとする方法である．また，疎ベクトル検索のアプローチの一種として分類される．SPLADE は BEIR において，BM25 を上回る性能を示している．そのため，性能の参考値としてベースライン手法とした．今回，BEIR の中で用いた実験データセットは SPLADE の著者らにならって使用している．

4.3　学習設定と実装

本研究では，検索のデータセットである MS MARCO で訓練したエンコーダを使用している．クロスエンコーダは後述のとおり，学習済みの配布されているモデルを使用している．その他の方法は，負の対数尤度を損失関数として用いて学習を行っている．各手法によるクエリと文書の関連度スコアを $S(Q, D)$ とし，正例となる文書を D^+，負例となる文書を $\{D_1^-, D_2^- ..., D_r^-, ...\}$ とすると，負の対数尤度は，

$$\mathcal{L} = -\log \frac{\exp(S(Q, D^+))}{\exp(S(Q, D^+)) + \sum_r \exp(S(Q, D_r^-))} \tag{13}$$

と表される．負例としては，クエリに対して BM25 による検索結果のうち正例ではないものとバッチ中の他のクエリの正例を使用する．BM25 による負例は Hard Negative，バッチ中の他の正例は In-Batch Negative などと呼ばれる [16]．

密ベクトル検索，ハイブリッド検索，C-BM25 のエンコーダとして，密ベクトル検索として訓練したエンコーダを使用する．つまり，式 (13) において，関連度スコアとして密ベクトル検索を用いたものである．事前学習済み言語モデルは，MPNet [*4] [31] を使用する．具体的には，microsoft/mpnet-base [*5] を使用している．損失関数の実装は，HardNegativeRankingLoss [*6] を使用している．

COIL-tok のエンコーダは，COIL-tok として訓練したエンコーダを使用する．つまり，Gao ら [9] と同様に，訓練時に使用する式 (13) 中の関連度スコアは式 (4) を用いて計算している．こちらも MPNet を事前学習済みモデルとしている．COIL-tok ではトークンを表現するベクトルの次元を線形変換によって小さくしているが，Gao らと同様にその次元数を 32 としている．学習の実装は Gao らの実装を用いる [*7]．

SPLADE [*8]，ColBERT [*9] では，事前学習済みモデルとして BERT [*10] を用いている．具体的には，bert-base-uncased [*11] を使用している．学習の実装はそれぞれの著者の実装を用いている．

上記の訓練を行う際のハイパーパラメータは，A.3 節に示す．なお，バッチサイズとトークンの最大長さ以外は，それぞれの著者らがデフォルトで設定した値を使用している．また，学習は NVIDIA V100 16 GB が 4 枚ついたサーバーにて実施する．

クロスエンコーダについては，事前学習済みモデルとして，Thakur ら [32] と同様に MiniLM-6L [38] を用いている．これを MS MSRCO を用いて訓練したモデルとして，

[*4] MPNet については，Sentence Transformers の公式サイト https://www.sbert.net/docs/pretrained_models.html にて，精度が良いとされているため，こちらを使用した．
[*5] https://huggingface.co/microsoft/mpnet-base
[*6] https://www.sbert.net/docs/package_reference/losses.html#multiplenegativesrankingloss
[*7] https://github.com/luyug/COIL
[*8] https://github.com/naver/splade
[*9] https://github.com/stanford-futuredata/ColBERT
[*10] ColBERT，SPLADE についても，MPNet を事前学習済みモデルとして訓練を実施したが，BERT の性能に及ばなかったため，これらについては BERT を使用した場合の結果を載せている．
[*11] https://huggingface.co/bert-base-uncased

表 1　BEIR における語彙一致・密ベクトル検索・文脈化語彙一致・ハイブリッド検索のリランキング結果．評価指標は nDCG@10．最も良かった結果を**太字**にしている．また，BM25 を上回った結果をイタリック体にしている．提案手法名をセリフ体にしている．Zero-shot 平均は，MS MARCO 以外のデータセットの nDCG@10 を平均している

Table 1　The results of re-ranking task with lexical retrieval, dense retrieval, contextualized lexical retrieval, hybrid retrieval on BEIR. The evaluation index is nDCG@10. The best results are labeled in **bold** and the results outperforming BM25 are labeld in *italic*. The names of our proposed methods are labeled in serif.

	語彙一致検索	密ベクトル検索		文脈化語彙一致検索		ハイブリッド検索	
	BM25	平均ベクトル	重み平均ベクトル	COIL-tok	C-BM25	H-BM25	HC-BM25
In Domain							
MS MARCO	0.5058	*0.6802*	**0.6822**	*0.6632*	*0.6687*	*0.5673*	*0.6726*
Zero-shot							
Arguana	0.2754	*0.3885*	*0.3859*	0.2245	**0.4487**	*0.3812*	*0.4485*
Climate-FEVER	0.1578	*0.2159*	*0.2187*	*0.1703*	*0.2440*	*0.2189*	**0.2457**
DBPedia	0.2846	*0.2939*	*0.2952*	*0.3219*	*0.3631*	*0.3250*	**0.3671**
FEVER	0.5768	*0.6156*	*0.6092*	*0.7417*	*0.7539*	*0.6485*	**0.7555**
FiQA	0.2361	0.2284	0.2333	*0.2553*	*0.3217*	*0.2826*	**0.3245**
HotpotQA	0.5674	0.4222	0.4221	*0.6050*	**0.6627**	*0.5845*	*0.6622*
NFCorpus	0.3301	0.2564	0.2596	0.3045	*0.3428*	0.3150	**0.3472**
NQ	0.2428	*0.3470*	*0.3450*	*0.3530*	*0.4086*	*0.3056*	**0.4110**
Quora	0.7886	*0.8238*	*0.8236*	0.7430	*0.8437*	*0.8185*	**0.8450**
SCIDOCS	0.1399	0.1058	0.1058	0.1332	*0.1593*	0.1364	**0.1596**
SciFact	0.6639	0.4614	0.4674	*0.6715*	*0.7110*	0.6554	**0.7141**
TREC-COVID	0.5302	*0.6376*	*0.6363*	*0.6961*	*0.7241*	*0.6805*	**0.7365**
Robust04	0.4088	0.3481	0.3655	0.4075	*0.4662*	*0.4427*	**0.4684**
Touché-2020	0.4536	0.2292	0.2697	0.2425	0.3608	**0.4546**	0.3608
Zero-shot 平均	0.4040	0.3838	0.3884	*0.4193*	*0.4865*	*0.4464*	**0.4890**

ms-marco-MiniLM-L-6-v2 [*12] を用いる[*13]．

上位 100 件の検索には pyserini [*14] を使用している．pyserini は anserini [41] の python ラッパーである．pyserini は内部で BM25 を使用している．BM25 のパラメータは，Thakur ら [32] と同じく，$k_1 = 0.9$，$b = 0.6$ とする．C-BM25 も同様に $k_1 = 0.9$，$b = 0.6$ を使用する．C-BM25 において，ベクトルの平均をとる窓幅 $o = 3$ とする．

遅延計測の実施時には，Gao ら [9] と同様にあらかじめすべてのトークンを表現するベクトルを転置インデックスにしている．密ベクトル検索や C-BM25 の遅延計測の実装においては，pytorch [28] を使用し，jupyter notebook の timeit コマンドを用いて実行している．遅延計測を行ったサーバーは，Intel Xeon Gold 6132 Processor 2.6 GHz の 8 コアを使用し，RAM は 600 GB である．

5. 結果

本章ではまず，C-BM25 の有効性について，BEIR を用いたリランキングタスクにて示す．次に，全文書を対象とした検索タスクで実行可能であることを示すため，Robust04 を用いた検索タスクの結果を示す．

5.1 BEIR を用いたリランキングタスクによる検証

本節では，BEIR でリランキングタスクを行った結果を用いて，C-BM25 の有効性を示す．**表 1** に結果を記す．また，Zero-shot 検索で有効な他のアプローチと比較した結果は**表 2** に記す．MS MARCO は，エンコーダを訓練するデータセットであるため，In-Domain としている．他のデータセットについては，そのデータセットで訓練をしていないので，Zero-shot としている．MS MARCO 以外のデータセットの nDCG@10 の平均を Zero-shot 平均としている．

まず，表 1 の結果から述べる．C-BM25 と BM25・密ベクトル検索を比較すると，C-BM25 は Zero-shot 平均の nDCG@10 で BM25 と密ベクトル検索を大きく上回っている．また，Touché-2020 を除く全データセットで BM25 を上回り，MS MARCO を除くすべてのデータセットで密ベクトル検索を上回っている．よって，C-BM25 が密ベクトル検索の課題であるドメインの変化に弱いという点

[*12] https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-6-v2
[*13] MPNet に対して，クロスエンコーダを訓練してリランキングタスクを実施したが，ms-marco-MiniLM-L-6-v2 の方が良い結果であったため，こちらを記載する．
[*14] https://github.com/castorini/pyserini/

表 2　BEIR における他の Zero-shot で有効な手法との比較．評価指標は nDCG@10 を用いている．最も良い結果を太字，提案手法名をセリフ体にしている．Zero-shot 平均は，MS MARCO 以外のデータセットの nDCG@10 を平均している

Table 2　The results of re-ranking task on BEIR with C-BM25, HC-BM25 and other methods which are effective in zero-shot cases. The evaluation index is nDCG@10. The best results are labeled in **bold**. The names of our proposed methods are labeled in serif.

	ColBERT	SPLADE	CE	C-BM25	HC-BM25
In Domain					
MS MARCO	0.6761	0.6853	**0.7267**	0.6687	0.6726
Zero-shot					
Arguana	0.2961	0.2627	0.3023	**0.4487**	0.4485
Climate-FEVER	0.1630	0.1115	0.2434	0.2440	**0.2457**
DBPedia	0.3693	0.3417	**0.4344**	0.3631	0.3671
FEVER	0.7512	0.7166	**0.7854**	0.7539	0.7555
FiQA	0.3006	0.1857	**0.3474**	0.3217	0.3245
HotpotQA	0.6109	0.5010	**0.7058**	0.6627	0.6622
NFCorpus	0.2886	0.2797	**0.3658**	0.3428	0.3472
NQ	0.4439	0.3603	**0.4589**	0.4086	0.4110
Quora	**0.8528**	0.7681	0.8252	0.8437	0.8450
SCIDOCS	0.1459	0.1182	**0.1619**	0.1593	0.1596
SciFact	0.6337	0.5411	0.6814	0.7110	**0.7141**
T-COVID	0.6618	0.5545	0.7357	0.7241	**0.7365**
Robust04	0.3973	0.3598	**0.4761**	0.4662	0.4684
Touché-2020	0.3163	0.2345	0.3205	**0.3608**	**0.3608**
Zero-shot 平均	0.4451	0.3811	0.4889	0.4865	**0.4890**

を解決していることが分かる．また，BM25 を上回っており，エンコーダが保持する文脈情報をうまく活用できている．なお，重み平均ベクトルはほとんど改善が見られず，Zero-shot 平均で BM25 を下回っている．この結果から，重み平均ベクトルは低頻度語の対策としてうまく機能していないことが分かる．

次に，C-BM25 と，同じ文脈化語彙一致検索である COIL-tok を比較する．COIL-tok も BM25 を Zero-shot 平均で上回ることができている．よって，文脈化語彙一致検索は，Zero-shot 時に有効であると考えられる．一方，C-BM25 はすべてのデータセットで COIL-tok を上回っており，同じ文脈化語彙一致検索でも，C-BM25 の方が優れていることが分かる．この要因については，6.1 節でスコアリング方法の違いを通じて考察を行う．

さらに，Zero-shot 時に有効といわれているハイブリッド BM25（H-BM25）と C-BM25 を比較する．H-BM25 は BM25 を Zero-shot 平均で上回っており，訓練データからドメインが変化した場合に H-BM25 が有効であることが分かる．ただし，H-BM25 は，MS MARCO で密ベクトル検索を大きく下回っており，単純なスコアの和では必ずしも補完的に機能しないことが分かる．一方，C-BM25 は Zero-shot 平均および Touché-2020 を除く全データセットで，H-BM25 を上回っている．よって，C-BM25 は H-BM25 よりも，密ベクトル検索で訓練されたエンコーダをより有効に活用できている．

最後に，HC-BM25 と C-BM25 を比較すると，HC-BM25 がわずかながら C-BM25 を上回っており，クエリと文書全体のマッチングとしての密ベクトル検索の関連度スコアが，C-BM25 においても寄与していることが分かる．

次に，他の Zero-shot 検索で有効なアプローチと比較した表 2 の結果について述べる．C-BM25 は，Zero-shot 平均で ColBERT や SPLADE も上回っている．なお，SPLADE については，Formal ら [7] と異なり BM25 を下回っている．これは，我々は負例に BM25 のみを用いているが，Formal らは密ベクトル検索での不正解文書も負例として使用したためと考えられる．クロスエンコーダと比較すると C-BM25 はこれに迫る性能を出し，HC-BM25 はわずかながら上回っている．このように，既存手法と同等あるいは上回る性能を出すことができている．

最後に，Touché-2020 において，C-BM25 が BM25 より劣る理由について考察する．Thakur ら [32] は，論拠検索である Arguana や Touché-2020 は，MS MARCO とは大きくタスクが異なるタスクであるとしている．そのため，MS MARCO で訓練したモデルは，これらのデータセットで良い性能を発揮しないとしている．我々も他の Zero-shot 検索で有効なアプローチにおいて，表 2 で同様な結果を得ている．よって，MS MARCO で学習した関連度をより反映させる C-BM25 によるスコアリングは，BM25 に劣ったに結果となったと考えられる．一方，同じ論拠検索タスクである Arguana は C-BM25 が BM25 を上回っている．Arguana はある主張の反対主張を関連ありとするタスクであり，論拠であれば関連文書であるとする Touché-2020 とは異なる．また，Wachsmuth ら [36] によると，反対主張は似た内容を反対の立場から述べるものであり，文類似度で有効なスコアリング方法が有効なタスクであるとされている．そのため，C-BM25 のように文脈を考慮するスコアリングが有効であったと考えられる．なお，上記のように，検索は何を関連文書とするかで大きく正解が異なる場合がある．どのようなタスクがどのような関連度を持つかは，今後の研究課題の 1 つと思われる．

5.2　Robust04 を用いた検索タスクによる検証

次に，検索タスクにおける有効性を確認するため，Robust04 を用いて，nDCG@10 による評価とサンプルしたクエリに対する遅延時間計測を実施した．結果を**表 3** に示す．検索タスクにおいても，C-BM25 が BM25 と平均ベクトルを nDCG@10 で上回っていることが分かる．遅延時間については，BM25 よりは遅いものの，平均ベクトルよりは速くすることができているため，検索タスクでも使用可能な方法と考えられる．文脈化語彙一致検索は，クエリと文書で一致した語彙を持つトークンを表現するベクトルとの内積計算になるため，内積を計算するベクトルを減らす

表 3　Robust04 における検索タスクの結果．最も良い結果は太字にしている

Table 3　The results of retrieval task on Robust04. The best results are labeled in **bold**.

	nDCG@10	遅延時間/ms
BM25	0.4484	**9.13**
平均ベクトル	0.3564	90.9
C-BM25	**0.4793**	61.9

ことができ，高速になることが Gao ら [9] により示されている．今回は，Gao らよりもベクトルの次元が大きいが，Gao らと同様に密ベクトル検索よりも高速に検索できている．

6. 各要素の影響分析

本章では，C-BM25 における狙いがどの程度性能向上に寄与していたのかを分析する．まず，語彙一致とトークンの BM25 スコアによる重み付けの効果を調べるため，語彙一致のみの場合のスコアリング方法と比較を行った．次に，周辺のトークンを活用するために局所平均を用いた効果を調べるために，窓幅のサイズを変更して実験を行った．

6.1 語彙一致と BM25 によるトークン重要度の影響

本節では，BM25 によるトークン重要度の影響を調査する．そのために，C-BM25 において，トークンの BM25 スコアで重み付けない場合について BEIR で実験を行った．これは，COIL-tok のスコアリングを内積からコサイン類似度に変更した場合と同等である．さらに，Zero-shot での課題は低頻度語であるので，IDF のみでも有効であると考えられる．そのため BM25 から TF の項を除いた場合，つまりトークンの IDF スコアを重要度として使用した場合についても実施した．比較は，密ベクトル検索モデルとして訓練を行ったエンコーダを用いて実施した．結果を**表 4** に記す．

まず，トークンの BM25 スコアによる重要度なしで BM25 を上回ることができている．一方，C-BM25 よりは Zero-shot 平均で低い性能となっているため，BM25 によるトークンの重み付けが有効であることが分かる．これが，C-BM25 が COIL-tok を Zero-shot 平均で上回った一因と考えられる．また，表 1 の COIL-tok の結果と比べると，トークンの BM25 スコアなしの方が Zero-shot 平均で良い結果となっている．よって，エンコーダの学習方法も C-BM25 との差に影響を与えていると考えられる．この点について，A.1 節でさらに調査を行っている．

続いて，TF の重みを用いない場合であるが，BM25 の重みなしを Zero-shot 平均で上回っており，IDF によるトークンのスコアも効果があることが分かる．一方，C-BM25 は BM25 スコアなしの場合も TF なしの場合も Zero-shot 平均で上回っている．よって，トークン重要度として TF も寄与していることが分かる．Gao ら [9] の論文中では，内積による文脈類似度は TF の代替として使用されている．しかし，Robertson ら [30] は，TF を eliteness というドキュメントが指すトピックへの関連度合いを指すと仮定していた．Zero-shot 時では，文脈情報は TF を代替するものではなく，検索において変わらず有効であると考えられる．

表 4　C-BM25 において，BM25 および TF による重み付けを行わなかった場合の結果．結果は，BEIR におけるでリランキング結果である．最も良い結果は太字にしている

Table 4　The results of re-ranking on the BEIR with scoring methods in which BM25 and TF weights are excluded from C-BM25. The best results are labeled in **bold**.

	BM25	BM25 スコアなし	TF なし	C-BM25
In Domain				
MS MARCO	0.5058	0.6591	0.6566	**0.6687**
Zero-shot				
Arguana	0.2754	0.3406	0.4047	**0.4487**
Climate-FEVER	0.1578	0.2294	0.2176	**0.2440**
DBPedia	0.2846	0.3303	0.3391	**0.3631**
FEVER	0.5768	0.7837	**0.7875**	0.7539
FiQA	0.2361	0.2609	0.2948	**0.3217**
HotpotQA	0.5674	0.5961	0.6492	**0.6627**
NFCorpus	0.3301	0.3283	0.3305	**0.3428**
NQ	0.2428	0.3472	0.3637	**0.4086**
Quora	0.7886	0.7936	0.8333	**0.8437**
SCIDOCS	0.1399	0.1416	0.1493	**0.1593**
SciFact	0.6639	0.6741	0.6906	**0.7110**
TREC-COVID	0.5302	0.6766	0.6562	**0.7241**
Robust04	0.4088	0.4380	0.4492	**0.4662**
Touché-2020	0.4536	0.3044	0.3256	**0.3608**
Zero-shot 平均	0.4040	0.4460	0.4637	**0.4865**

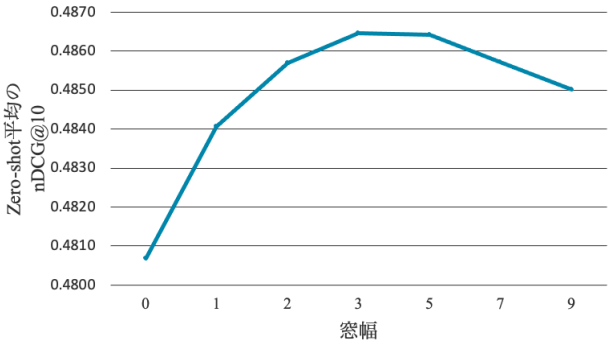


図 3　窓幅が及ぼす性能への影響．値は Zero-shot 平均での nDCG@10．エンコーダは密ベクトル検索で訓練したものを使用している

Fig. 3　The window size effect for performance. The vertical axis value is nDCG@10 of Zero-shot average. The encoder is trained as a dense retriever.

6.2 窓幅の影響

本節では，窓幅 o による性能への影響を見る．そのた

表 5　成功事例における各検索手法による 1 位文書の結果．C-BM25 のみが適合と判定された結果を出力している

Table 5　The first ranked document by each retrieval method. Only the document retrieved by C-BM25 is a relevant document.

クエリ	Give me all movies directed by Francis Ford Coppola.
BM25 の 1 位文書	Don't Box Me In Don't Box Me In is a collaboration between Stewart Copeland and Stan Ridgway. It was recorded as part of the soundtrack for the Francis Ford Coppolaś movie Rumble Fish and was subsequently released as a single. Copeland plays guitar, drums, bass and keyboards, and Ridgway sings and plays harmonica.
平均ベクトルの 1 位文書	All of Me (1984 film) All of Me is a 1984 fantasy comedy film directed by Carl Reiner and starring Steve Martin and Lily Tomlin. This film is based on the novel Me Two by Edwin Davis.
C-BM25 の 1 位文書	The Godfather The Godfather is a 1972 American crime film directed by Francis Ford Coppola and produced by Albert S. Ruddy from a screenplay by Mario Puzo and Coppola.

め，BEIR において，C-BM25 の窓幅を変化させてリランキングタスクを実施した．エンコーダは，密ベクトル検索モデルとして訓練を行ったエンコーダを用いている．結果を**図 3** に示す．結果を見ると，窓幅 $o = 3$ の場合に最も良く，それより短い場合も長い場合も性能が下がっている．トークンをエンコードしたベクトルをそのまま用いるより，局所平均をとる方が良いことから，クエリと文書で語彙一致したトークンの周辺のトークンを利用することが性能向上に寄与していると考えられる．また，長い窓幅になると性能が劣化しており，局所的な文脈の方が性能に寄与することが推察される．

7.　事例観察

本章では，事例を用いて C-BM25 の成功要因と課題を明らかにする．比較対象として，BM25 と平均ベクトルを用いる．成功事例としては，両者よりも C-BM25 が nDCG@10 で上回った事例を用いる．失敗事例として，両者よりも C-BM25 が nDCG@10 で下回った事例を用いる．また，事例は各クエリに対するそれぞれの 1 位文書を用いる．成功事例では，C-BM25 の 1 位文書のみが適合であり，失敗事例では，C-BM25 の 1 位文書のみが不適合である．なお，データセットとしては，DBpedia [12] を用いる．

7.1　成功事例

成功事例分析として，“Give me all movies directed by Francis Ford Coppola.” というクエリ対する分析を行った．BM25，平均ベクトル，C-BM25 の各手法の 1 位文書を**表 5** に示す．C-BM25 はクエリに対して，Francis Ford Coppola が監督した作品を出力できている．一方，平均ベクトルの 1 位文書は映画の話であり，誰が監督しているかまで記載されているが，Francis Ford Coppola が監督した作品ではない．また，BM25 の 1 位文書は Francis Ford Coppola が文書に含まれているが，彼が監督した映画ではなく，採用されたサウンドトラックの話になっている．

より細かい分析を行うために，BM25 の 1 位文書と C-BM25 の 1 位文書をクエリの各トークンのスコアを用いてを比較する．**表 6** にクエリの各トークンの IDF スコアと BM25 スコアおよび C-BM25 の文脈類似度を示す．BM25 と C-BM25 の双方において，期待どおり Francis Ford Coppola に該当するトークンの BM25 スコアが相対的に大きくなっている．まず，なぜ C-BM25 の 1 位文書が BM25 では 1 位文書になれなかったかを考察する．pyserini のトークナイザにおける IDF スコアを見ると，direct のスコアが相対的に小さい．これが，BM25 では C-BM25 で 1 位であった文書を 1 位に出すことができない一因と考えられる．次に，なぜ BM25 の 1 位文書が C-BM25 文書ではより下位になったのかを考察する．BM25 の 1 位文書と C-BM25 の 1

表 6　表 5 のクエリにおける各トークンの IDF と BM25 スコアと文脈類似度．BM25 と C-BM25 の 1 位文書を対象としている．なお，BM25 は pyserini の標準のトークナイザを使用しており，C-BM25 は MPNet のトークナイザを使用しているため，トークン化された結果は異なっている

Table 6　IDF scores, BM25 scores and context similarity of each term in the query of Table 5 for top ranked documents by BM25 and C-BM25. Notice that the pyserini default tokenizer is used in BM25 and the tokenizer of MPNet is used in C-BM25. Thus, the resulted tokens are different.

pyserini のトークナイザ	
クエリの各トークンの IDF スコア	give: 5.55, me: 5.71, all: 3.39, movi: 5.1, direct: 3.57, franci: 5.77, ford: 6.51, coppola: 9.72
BM25 の 1 位文書における クエリの各トークンの BM25 スコア	give: 0.0, me: 7.54, all: 0.0, movi: 5.16, direct: 0.0, franci: 5.85, ford: 6.59, coppola: 9.84
MPNet のトークナイザ	
クエリの各トークンの IDF スコア	give: 6.51, me: 4.48, all: 3.3, movies: 6.38, directed: 3.78, by: 1.22, francis: 5.71, ford: 6.43, cop: 6.87, ##pol: 6.01, ##a: 2.42, .: 0.01
BM25 の 1 位文書における クエリの各トークンの BM25 スコア	give: 0.0, me: 5.83, all: 0.0, movies: 0.0, directed: 0.0, by: 0.0, francis: 5.66, ford: 6.38, cop: 6.81, ##pol: 5.96, ##a: 2.4, .: 0.01
C-BM25 の 1 位文書における クエリの各トークンの BM25 スコア	give: 0.0, me: 0.0, all: 0.0, movies: 0.0, directed: 4.38, by: 1.91, francis: 6.6, ford: 7.44, cop: 9.88, ##pol: 8.64, ##a: 3.48, .: 0.01
BM25 の 1 位文書における クエリの各トークンの文脈類似度	give: 0.0, me: 0.27, all: 0.0, movies: 0.0, directed: 0.0, by: 0.0, francis: 0.34, ford: 0.37, cop: 0.38, ##pol: 0.37, ##a: 0.35, .: 0.22
C-BM25 の 1 位文書における クエリの各トークンの文脈類似度	give: 0.0, me: 0.0, all: 0.0, movies: 0.0, directed: 0.48, by: 0.51, francis: 0.53, ford: 0.54, cop: 0.54, ##pol: 0.51, ##a: 0.51, .: 0.49

位文書で各トークンの文脈類似度を比べると，C-BM25 の方が全体的に大きいことが分かる．C-BM25 の 1 位文書では directed by Francis Ford Coppola と出てくるため，クエリのフレーズと合致しており，文脈類似度が大きくなったと推察される．これより，エンコーダが文脈の類似度をよく判別できていると考えられる．また，direct のような比較的頻度が高い単語も，文脈類似度を通じて影響を与えていることが推察される．

一方で，平均ベクトルの 1 位文書に注目すると，クエリにある Francis Ford Coppola ではなく，Carl Reiner や Steve Martin や Lily Tomlin など他の人名が 1 位文書に含まれていることが分かる．また，重み平均ベクトルの 1 位文書も同じ結果であった．そこで，各人名のトークンとそのトークンの IDF スコアをみる．結果を**表 7** に示す．Francis Ford Coppola 以外の人名のトークンについても，Francis Ford Coppola と同等の IDF スコアとなっている．このことから，語彙一致が重要である一因が推察される．検索タスクにおいては，知りたい情報を識別するために使用される単語は低頻度になることが想定される．また，この低頻度語と同義と判定される語彙は少ないことが予想される．よって，このような低頻度語を用いた識別は，ほとんど完全にトークンが一致することと一致した低頻度語に重みを付けることが最も確実な手段となると考えられる．よって，語彙一致が重要であったと考えられる．つまり，検索では低頻度語の識別が重要であり，その確実な方法が語彙の一致であるということである．このようなマッチングは，密ベクトル検索では現状困難と考えられる．

以上から，この事例では，C-BM25 は Francis Ford Coppola という監督名を語彙一致とトークン重要度で判別しつつ，directed by についても文脈で考慮できたため，正解文書を 1 位にすることができたと考えられる．また，スコアリング方法の性質からも，他のクエリも同様に語彙一致とトークン重要度による低頻度語の識別と局所的な文脈の考慮によって性能が向上したと推察される．

表 7　表 5 の平均ベクトルの 1 位文書およびクエリ中の人名とそのトークンの IDF スコア

Table 7　Person's names and their token's IDF-score in the query and the top ranked document retrieved by Average Vector in Table 5.

文書中の表記	MPNet のトークナイザーによるトークナイズ結果と各トークンの IDF スコア
Francis Ford Coppola	francis: 5.71, ford: 6.43, cop: 6.87, pol: 6.01, a: 2.42
Carl Reiner	carl: 5.98, rein: 6.74, ##er: 3.34
Steve Martin	steve: 5.93, martin: 5.26
Lily Tomlin	lily: 7.8, tom: 5.35, ##lin: 5.65

7.2　失敗事例

失敗事例分析として，“vietnam food recipes” というクエリ対する分析を行った．BM25，平均ベクトル，C-BM25 の各ランキング手法の 1 位文書を**表 8** に示す．C-BM25 の 1 位文書は料理のレシピとまったく関係ない．一方，密ベクトル検索はベトナム料理の文書を返しており，BM25 も料理のレシピの文書を返している．

平均ベクトルの 1 位文書に注目し，これを D_{a1} とおく．D_{a1} では vietnam 以外にクエリと語彙が一致するトークンがない．実際，クエリの各トークンの BM25 スコアが，$\mathrm{BM25}(\mathrm{vietnam}, D_{a1}) = 5.98$, $\mathrm{BM25}(\mathrm{food}, D_{a1}) = 0.0$, $\mathrm{BM25}(\mathrm{recipes}, D_{a1}) = 0.0$ となっていた．文書を見ると foods があり，vietnammese が vietnam より高頻度に発生している．このように，クエリのトークンから語形変化した語彙は存在している．C-BM25 が語彙の完全一致をベー

表 8　失敗事例における各検索手法による 1 位文書の結果．C-BM25 以外は適合と判定された文書を出力した

Table 8　The first ranked document by each retrieval method. All of them except for the one retrieved by C-BM25 are relevant documents.

クエリ	vietnam food recipes
BM25 の 1 位文書	Raw Food Made Easy for 1 or 2 People Raw Food Made Easy for 1 or 2 People is a recipe book by raw food chef Jennifer Cornbleet. The best-selling book was published in 2005 and promotes the raw food diet, a dietary movement that encourages the consumption of uncooked foods to obtain maximum health benefits. The book features 115 recipes, including 21 breakfast recipes, 64 lunch and dinner recipes and 30 dessert recipes. Each recipe yields servings for one or two people.
平均ベクトルの 1 位文書	Vietnamese cuisine Vietnamese cuisine encompasses the foods and beverages of Vietnam, and features a combination of five fundamental tastes (Vietnamese: nguvi) in the overall meal. Each Vietnamese dish has a distinctive flavor which reflects one or more of these elements. Common ingredients include fish sauce, shrimp paste, soy sauce, rice, fresh herbs, and fruits and vegetables.
C-BM25 の 1 位文書	Luke Nguyen's Vietnam Luke Nguyen's Vietnam is an Australian television series first screened on SBS One in 2010. The series follows chef, Luke Nguyen, as he tours Vietnam seeking culinary delights and adventure. It is regularly broadcast on Good Food, a UK food-orientated TV channel.

スとしているため，これらを評価できなかったと考えられる．

なお，C-BM25 の 1 位文書と BM25-1 位文書を比較した結果，文脈類似度が期待と異なる値を示す事例であった．詳細は，A.2 節にて記載する．

8. 結論

本論文では，Zero-shot 時の検索手法として C-BM25 および HC-BM25 を提案した．そして，これらが Zero-shot の場合に BM25 や密ベクトル検索の性能を上回り，他の Zero-shot 時に有効な手法も上回るまたは同等の性能となることを示した．また，C-BM25 が文書全体を検索する場合も実用可能な速度で検索しうることを示した．

次に，C-BM25 における各要素の影響分析を通じて，C-BM25 は，BM25 によるトークン重みなしでも，BM25 を上回ることを確認した．また，局所平均が有効であることも確認した．この分析に加えて，事例観察を行い，Zero-shot 検索において文脈化語彙一致が有効になる理由について単語頻度の観点から考察を行った．

一方で，語彙一致をベースとすることから，従来の語彙一致検索で発生していた課題と同様の課題が生じることが明らかになったため，この解消は今後の課題としたい．

最後に，本手法を検索で用いる際は，全トークンのベクトルをあらかじめ転置インデックスとして保持する．同様に全トークンのベクトルをインデックスする COIL-tok や ColBERT と比較して，ベクトルの次元が大きく非常に大きなメモリ容量が必要となる．実応用可能な範囲を広げるためには，よりインデックスの容量を減らすことが今後の課題であるが，解決方法としては Product Quantization [25] などによる離散化などが候補となると考えられる．

謝辞　この成果は，国立研究開発法人新エネルギー・産業技術総合開発機構（NEDO）の委託業務（JPNP18002）の結果得られたものです．また，本研究では，産総研の AI 橋渡しクラウド（ABCI）を利用した．

参考文献

[1] Bondarenko, A., Fröbe, M., Beloucif, M., Gienapp, L., Ajjour, Y., Panchenko, A., Biemann, C., Stein, B., Wachsmuth, H., Potthast, M. and Hagen, M.: Overview of Touché 2020: Argument Retrieval, *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pp.384–395 (2020).

[2] Boteva, V., Gholipour, D., Sokolov, A. and Riezler, S.: A Full-Text Learning to Rank Dataset for Medical Information Retrieval, *Advances in Information Retrieval*, pp.716–722 (2016).

[3] Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I. and Specia, L.: SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation, *Proc. 11th International Workshop on Semantic Evaluation* (*SemEval-2017*), pp.1–14 (2017).

[4] Cohan, A., Feldman, S., Beltagy, I., Downey, D. and Weld, D.: SPECTER: Document-level Representation Learning using Citation-informed Transformers, *Proc. 58th Annual Meeting of the Association for Computational Linguistics*, Online, pp.2270–2282 (2020).

[5] Dai, Z. and Callan, J.: Context-Aware Term Weighting For First Stage Passage Retrieval, *Proc. 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, *SIGIR '20*, pp.1533–1536 (2020).

[6] Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, *Proc. 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).

[7] Formal, T., Lassance, C., Piwowarski, B. and Clinchant, S.: SPLADE v2: Sparse Lexical and Expansion Model for Information Retrieval, arXiv:abs/2109.10086 (2021).

[8] Formal, T., Piwowarski, B. and Clinchant, S.: Match Your Words! A Study of Lexical Matching in Neural Information Retrieval, arXiv:abs/2112.05662 (2021).

[9] Gao, L., Dai, Z. and Callan, J.: COIL: Revisit Exact Lexical Match in Information Retrieval with Contextualized Inverted List, *Proc. 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Online, pp.3030–3042 (2021).

[10] Gao, L., Dai, Z., Chen, T., Fan, Z., Durme, B.V. and Callan, J.: Complement Lexical Retrieval Model with Semantic Residual Embeddings, *Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021*, Vol.12656, pp.146–160 (2021).

[11] Guo, J., Fan, Y., Ai, Q. and Croft, W.B.: A Deep Relevance Matching Model for Ad-hoc Retrieval, *Proc. 25th ACM International on Conference on Information and Knowledge Management*, *CIKM '16*, pp.55–64 (2016).

[12] Hasibi, F., Nikolaev, F., Xiong, C., Balog, K., Bratsberg, S.E., Kotov, A. and Callan, J.: DBpedia-Entity v2: A Test Collection for Entity Search, *Proc. 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, *SIGIR '17*, pp.1265–1268 (2017).

[13] Hofstätter, S., Lin, S.-C., Yang, J.-H., Lin, J. and Hanbury, A.: Efficiently Teaching an Effective Dense Retriever with Balanced Topic Aware Sampling, *Proc. 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.113–122 (2021).

[14] Hui, K., Yates, A., Berberich, K. and de Melo, G.: CO-PACRR: A Context-Aware Neural IR Model for Ad-hoc Retrieval, *WSDM*, pp.279–287 (2018).

[15] Izacard, G., Caron, M., Hosseini, L., Riedel, S., Bojanowski, P., Joulin, A. and Grave, E.: Towards Unsupervised Dense Information Retrieval with Contrastive Learning (2021).

[16] Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D. and Yih, W.-T.: Dense Passage Retrieval for Open-Domain Question Answering, *Proc. 2020 Conference on Empirical Methods in Natural Language Processing* (*EMNLP*), Online, pp.6769–6781 (2020).

[17] Khattab, O. and Zaharia, M.: ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT, *Proc. 43rd International ACM SI-*

GIR Conference on Research and Development in Information Retrieval, *SIGIR '20*, pp.39–48 (2020).

[18] Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Devlin, J., Lee, K., Toutanova, K., Jones, L., Kelcey, M., Chang, M.-W., Dai, A.M., Uszkoreit, J., Le, Q. and Petrov, S.: Natural Questions: A Benchmark for Question Answering Research, *Trans. Association for Computational Linguistics*, Vol.7, pp.452–466 (2019).

[19] Leippold, M. and Diggelmann, T.: Climate-FEVER: A Dataset for Verification of Real-World Climate Claims, *NeurIPS 2020 Workshop on Tackling Climate Change with Machine Learning* (2020).

[20] Lin, S.-C., Yang, J.-H. and Lin, J.: In-Batch Negatives for Knowledge Distillation with Tightly-Coupled Teachers for Dense Retrieval, *Proc. 6th Workshop on Representation Learning for NLP* (*RepL4NLP-2021*), Online, pp.163–173 (2021).

[21] Luan, Y., Eisenstein, J., Toutanova, K. and Collins, M.: Sparse, Dense, and Attentional Representations for Text Retrieval, *Trans. Association for Computational Linguistics*, Vol.9, pp.329–345 (2021).

[22] Ma, J., Korotkov, I., Yang, Y., Hall, K. and McDonald, R.: Zero-shot Neural Passage Retrieval via Domain-targeted Synthetic Question Generation, *Proc. 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, Online, pp.1075–1088 (2021).

[23] MacAvaney, S., Yates, A., Cohan, A. and Goharian, N.: CEDR: Contextualized Embeddings for Document Ranking, *Proc. 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, *SIGIR'19*, pp.1101–1104 (2019).

[24] Maia, M., Handschuh, S., Freitas, A., Davis, B., McDermott, R., Zarrouk, M. and Balahur, A.: WWW'18 Open Challenge: Financial Opinion Mining and Question Answering, *Companion Proc. The Web Conference 2018*, *WWW '18*, pp.1941–1942 (2018).

[25] Matsui, Y., Uchida, Y. and Satoh, S.: A survey of product quantization, *ITE Trans. Media Technology and Applications*, Vol.6, No.1, pp.2–10 (2018).

[26] Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R. and Deng, L.: MS MARCO: A Human Generated MAchine Reading COmprehension Dataset, *Proc. Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems* (*NIPS 2016*), Vol.1773 (2016).

[27] Nogueira, R. and Cho, K.: Passage Re-ranking with BERT, arXiv:abs/1901.04085 (2019).

[28] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J. and Chintala, S.: PyTorch: An Imperative Style, High-Performance Deep Learning Library, *Advances in Neural Information Processing Systems 32*, pp.8024–8035 (2019).

[29] Rao, J., Liu, L., Tay, Y., Yang, W., Shi, P. and Lin, J.: Bridging the Gap between Relevance Matching and Semantic Matching for Short Text Similarity Modeling, *EMNLP-IJCNLP*, pp.5370–5381 (2019).

[30] Robertson, S.E. and Walker, S.: Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval, *Proc. 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, *SIGIR '94*, pp.232–241 (1994).

[31] Song, K., Tan, X., Qin, T., Lu, J. and Liu, T.-Y.: MPNet: Masked and Permuted Pre-training for Language Understanding, *Advances in Neural Information Processing Systems*, Vol.33, pp.16857–16867 (2020).

[32] Thakur, N., Reimers, N., Rücklé, A., Srivastava, A. and Gurevych, I.: BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models, *35th Conference on Neural Information Processing Systems Datasets and Benchmarks Track* (*Round 2*) (2021).

[33] Thorne, J., Vlachos, A., Christodoulopoulos, C. and Mittal, A.: FEVER: A Large-scale Dataset for Fact Extraction and VERification, *NAACL HLT 2018*, pp.809–819 (2018).

[34] Voorhees, E., Alam, T., Bedrick, S., Demner-Fushman, D., Hersh, W.R., Lo, K., Roberts, K., Soboroff, I. and Wang, L.L.: TREC-COVID: constructing a pandemic information retrieval test collection, *SIGIR Forum*, Vol.54, No.1, pp.1–12 (2021).

[35] Voorhees, E.M.: Overview of the TREC 2004 robust retrieval track, *TREC* (2004).

[36] Wachsmuth, H., Syed, S. and Stein, B.: Retrieval of the Best Counterargument without Prior Topic Knowledge, *Proc. 56th Annual Meeting of the Association for Computational Linguistics* (*Volume 1: Long Papers*), pp.241–251 (2018).

[37] Wadden, D., Lin, S., Lo, K., Wang, L.L., van Zuylen, M., Cohan, A. and Hajishirzi, H.: Fact or Fiction: Verifying Scientific Claims, *Proc. 2020 Conference on Empirical Methods in Natural Language Processing* (*EMNLP*), Online, pp.7534–7550 (2020).

[38] Wang, W., Wei, F., Dong, L., Bao, H., Yang, N. and Zhou, M.: MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020* (2020).

[39] Xin, J., Xiong, C., Srinivasan, A., Sharma, A., Jose, D. and Bennett, P.N.: Zero-Shot Dense Retrieval with Momentum Adversarial Domain Invariant Representations, arXiv:abs/2110.07581 (2021).

[40] Xiong, L., Xiong, C., Li, Y., Tang, K.-F., Liu, J., Bennett, P.N., Ahmed, J. and Overwijk, A.: Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval, *ICLR* (2021).

[41] Yang, P., Fang, H. and Lin, J.: Anserini: Enabling the use of lucene for information retrieval research, *SIGIR 2017 – Proc. 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.1253–1256 (2017).

[42] Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W., Salakhutdinov, R. and Manning, C.D.: HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering, *Proc. 2018 Conference on Empirical Methods in Natural Language Processing*, pp.2369–2380 (2018).

[43] Zhan, J., Mao, J., Liu, Y., Guo, J., Zhang, M. and Ma, S.: Optimizing Dense Retrieval Model Training with Hard Negatives, *Proc. 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.1503–1512 (2021).

付　　録

A.1　COIL-tok のエンコーダとの比較

6.1 節では，密ベクトル検索として訓練したエンコーダが COIL-tok として訓練したエンコーダより，文脈化語彙一致検索において優れていることが示唆された．しかし，6.1 節の比較では，ベクトルの次元という点で COIL-tok に不利な設定になっている．そこで，より詳細に比較するため，COIL-tok において，線形変換後の次元数を 768 としてトレーニングしたエンコーダを用いた場合についても実験を行った．最良の結果で比較を行うために，スコアリング方法として，C-BM25 を用いた．ただし，COIL-tok をエンコーダで用いる場合は，訓練時と揃えるため文脈類似度としてコサイン類似度ではなく内積を使用した．同様に，トークンを表現するベクトルは，局所平均を用いずにトークンをエンコードしたベクトルをそのまま用いた．なお，コサイン類似度でも実験を行ったが，COIL-tok として訓練したエンコーダを用いた場合，内積の方が良い結果であった．局所平均についても同様であった．結果は**表 A·1** のとおりである．Zero-shot 平均において，COIL-tok で次元数を 768 次元としても，密ベクトル検索のエンコーダが優れていることが分かる．

Gao ら [9] では，TF の代わりとして文脈化語彙一致を持ち出しており，ゆえに内積が使われていると考えられる．よって，トークンの重要度と文脈が混ざった形で学習されていると推察される．Thakur ら [32] の実験では，トークン重要度を学習する DeepCT [5] が Zero-shot 検索で BM25 に劣る結果となっている．また，Formal ら [8] は，あるタスクで学習したトークンの重要度はそのタスク特異なものである可能性を示唆している．このことから，密ベクトル検索のエンコーダが優れた結果を示したと考えられる．一方で，SPLADE は学習を通じて，BM25 より適したトークン重要度とクエリ・文書拡張を実現による手法と考えられ，SPLADE は Zero-shot 時に有効であることが示されている．これは，学習で獲得できるトークン重要度が学習したタスク特有であるという説と矛盾する．ある検索データセットの学習で得られるトークン重要度が他の検索データセットで有効かどうかの明確化については今後の課題としたい．

表 A·1　BEIR における C-BM25 のリランキング結果．COIL-tok，COIL-tok（768 次元），密ベクトル検索をエンコーダとして用いた場合で比較している．最も良い結果は太字にしている

Table A·1　The result of re-ranking on the BEIR with C-BM25 using encoder as COIL-tok, COIL-tok (768dim) and Dense Retriever. The best results are labeled in **bold**.

エンコーダ	COIL-tok	COIL-tok（768 次元）	密ベクトル検索
	C-BM25（内積）	C-BM25（内積）	C-BM25
In Domain			
MS MARCO	0.6677	**0.6744**	0.6687
Zero-shot			
Arguana	0.4445	0.4343	**0.4487**
Climate-FEVER	0.2009	0.2077	**0.2440**
DBPedia	0.3100	0.3511	**0.3631**
FEVER	0.7143	0.7337	**0.7539**
FiQA	0.2836	0.3105	**0.3217**
HotpotQA	0.6070	0.6417	**0.6627**
NFCorpus	0.3144	0.3277	**0.3428**
NQ	0.3812	0.4021	**0.4086**
Quora	0.7595	0.7803	**0.8437**
SCIDOCS	0.1518	0.1540	**0.1593**
SciFact	0.6786	0.6882	**0.7110**
TREC-COVID	**0.7584**	0.7478	0.7241
Robust04	0.4428	0.4604	**0.4662**
Touché-2020	0.2986	0.3132	**0.3608**
Zero-shot 平均	0.4533	0.4681	**0.4865**

A.2　失敗事例における C-BM25 の 1 位文書と BM25 の 1 位文書を比較した詳細分析

C-BM25 の 1 位文書と BM25 の 1 位文書との比較を 7.1 節と同様に行う．**表 A·2** にクエリの各トークンの IDF スコアと BM25 スコアおよび C-BM25 の文脈類似度を示す．BM25 は pyserini の標準のトークナイザを使用しており，C-BM25 は MPNet のトークナイザを使用している．では，なぜ BM25 の 1 位文書が C-BM25 では 1 位文書になれなかったかを考察する．IDF スコアを見ると recipies の idf の値は高く，BM25 の 1 位文書は recipies を含んでいる．しかしながら，文脈類似度を見ると，recipies の文脈類似度について BM25 の文書では低くなっている．一方，C-BM25 の 1 位文書では，vietnam, food のどちらのトークンも文脈類似度が大きくなっている．これらから，この

表 A·2　表 8 のクエリにおける各トークンの IDF と BM25 スコアと文脈類似度．BM25 と C-BM25 の 1 位文書を対象としている．なお，BM25 と C-BM25 で使用するトークナイザが異なるため，トークン化された結果は異なっている

Table A·2　IDF scores, BM25 scores and context similarity of each term in the query of Table 8 for top ranked documents by BM25 and C-BM25. Notice that the tokenized results are different because the tokenizers used in BM25 and C-BM25 are different.

pyserini のトークナイザ	
クエリの各トークンの IDF スコア	vietnam: 6.02, food: 5.62, recip: 8.24
BM25 の 1 位文書におけるクエリの各トークンの BM25 スコア	vietnam: 0.0, food: 8.47, recip: 12.86
MPNet のトークナイザ	
クエリの各トークンの IDF スコア	vietnam: 5.96, food: 5.7, recipes: 8.86
BM25 の 1 位文書におけるクエリの各トークンの BM25 スコア	vietnam: 0.0, food: 8.5, recipes: 13.2
C-BM25 の 1 位文書におけるクエリの各トークンの BM25 スコア	vietnam: 8.92, food: 7.71, recipes: 0.0
BM25 の 1 位文書におけるクエリの各トークンの文脈類似度	vietnam: 0.0, food: 0.25, recipes: 0.27
C-BM25 の 1 位文書におけるクエリの各トークンの文脈類似度	vietnam: 0.51, food: 0.49, recipes: 0.0

クエリについては，文脈類似度が期待している値になっていないことが要因として考えられる．

A.3　各手法の学習時の設定

4.3 節にて言及した各手法の学習時の設定を**表 A·3**，**表 A·4**，**表 A·5**，**表 A·6** に記す．

表 A·3　密ベクトル検索のエンコーダ訓練時の学習パラメータ
Table A·3　Learning parameter for dense retriever.

バッチサイズ	20
最大文書長さ	300
学習率	2e-5
エポック数	5
Warmup ステップ	1,000

表 A·4　COIL-tok の訓練時の学習パラメータ
Table A·4　Learning parameter for COIL-tok.

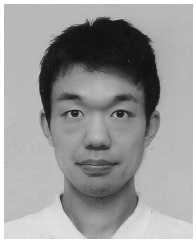
バッチサイズ	16
最大クエリ長さ	64
最大文書長さ	300
学習率	5e-6
エポック数	5
Warmup 率	0.1

表 A·5　ColBERT の訓練時の学習パラメータ
Table A·5　Learning parameter for ColBERT.

バッチサイズ	16
最大文書長さ	512
学習率	3e-6
学習 step 数	400,000
トークンベクトルの次元	128

表 A·6　SPLADE の訓練時の学習パラメータ
Table A·6　Learning parameter for SPLADE.

バッチサイズ	24
最大文書長さ	300
学習率	2e-5
エポック数	5
Warmup ステップ	1,000
クエリの FLOPS による正則化項の重み	0.0006
文書の FLOPS による正則化項の重み	0.0008



飯田 大貴

1989 年生．2012 年東京大学工学部卒業．2014 年東京大学大学院工学系研究科修士課程修了．独立行政法人を経て，2016 年株式会社レトリバ入社．自然言語処理技術の実応用に従事．2020 年東京工業大学情報理工学院博士課程入学，現在に至る．言語処理学会，人工知能学会各会員．

岡崎 直観　（正会員）

1979 年生．2007 年東京大学大学院情報理工学系研究科博士課程修了．2007 年同大学大学院情報理工学系研究科・特任研究員．2011 年東北大学大学院情報科学研究科准教授．2017 年東京工業大学情報理工学院教授．自然言語処理の研究に従事．言語処理学会，人工知能学会，ACL 各会員．

（担当編集委員　野宮 浩揮）