Leveraging Cognitive Complexity of Texts for Contextualization in Dense Retrieval

Effrosyni Sokli[1] **Georgios Peikos**[1] **Pranav Kasela**[1] **Gabriella Pasi**[1]
[1]University of Milano-Bicocca, Italy
{effrosyni.sokli, georgios.peikos, pranav.kasela, gabriella.pasi}@unimib.it

Abstract

Dense Retrieval Models (DRMs) estimate the semantic similarity between queries and documents based on their embeddings. Prior studies highlight the importance of embedding contextualization in enhancing retrieval performance. To this aim, existing approaches primarily leverage token-level information derived from query/document interactions. In this paper, we introduce a novel DRM, namely DenseC3, which leverages query/document interactions based on the full embedding representations generated by a Transformer-based model. To enhance similarity estimation, DenseC3 integrates external linguistic information about the Cognitive Complexity of texts, enriching the contextualization of embeddings. We empirically evaluate our approach across seven benchmarks and three different IR tasks to assess the impact of Cognitive Complexity-aware query and document embeddings for contextualization in dense retrieval. Results show that our approach consistently outperforms standard fine-tuning techniques on lightweight bi-encoders (e.g., BERT-based) and traditional late-interaction models (i.e., ColBERT) across all benchmarks. On larger retrieval-optimized bi-encoders like Contriever, our model achieves comparable or higher performance on four of the considered evaluation benchmarks. Our findings suggest that Cognitive Complexity-aware embeddings enhance query and document representations, improving retrieval effectiveness in DRMs. Our code is available online at: `https://github.com/FaySokli/DenseC3`.

1 Introduction

Information Retrieval (IR) systems aim to retrieve relevant documents in response to users' queries. The current state of the field includes lexicon-based models (e.g., BM25 (Robertson et al., 1994)), neural IR models that exploit semantic similarities, and hybrid architectures that combine lexicon-based with neural models used for re-ranking (e.g., cross-encoders). Based on their architecture, neural IR models can be categorized as follows: (i) Cross-encoders (e.g., Rosa et al. (2022)), (ii) Bi-encoders (e.g., Contriever (Izacard et al., 2022)), and (iii) Late query/document interaction models (e.g., ColBERT (Khattab and Zaharia, 2020)). The latter two constitute key categories of Dense Retrieval Models (DRMs). Depending on their architecture, DRMs often enhance retrieval effectiveness but introduce computational overhead, leading to increased query latency (see Section 5). DRMs project both queries and documents into a shared dense vector space and rank documents based on their similarity to queries (e.g., by using a dot product). In this scenario, contextualization techniques aim to enrich query and document embeddings by injecting contextual knowledge and enhance their representations to improve the performance of DRMs. Prior studies leverage linguistic features to contextualize embeddings with external infor-

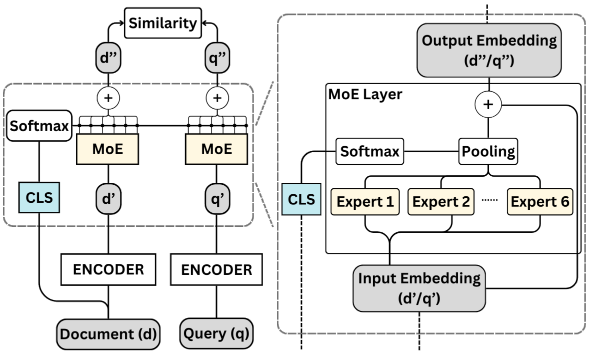

Figure 1: DenseC3. Our architecture employs a bi-encoder that exploits late query/document interactions on a full embedding level, leveraging the Mixture-of-Experts framework. The enhanced part displays DenseC3's main components: (i) the gating mechanism (CLS); (ii) the six experts (one for each Cognitive Complexity level of Bloom's Taxonomy); and (iii) the pooling module, which aggregates the output of the experts and defines the final query or document embedding.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 27072–27085
November 4-9, 2025 ©2025 Association for Computational Linguistics

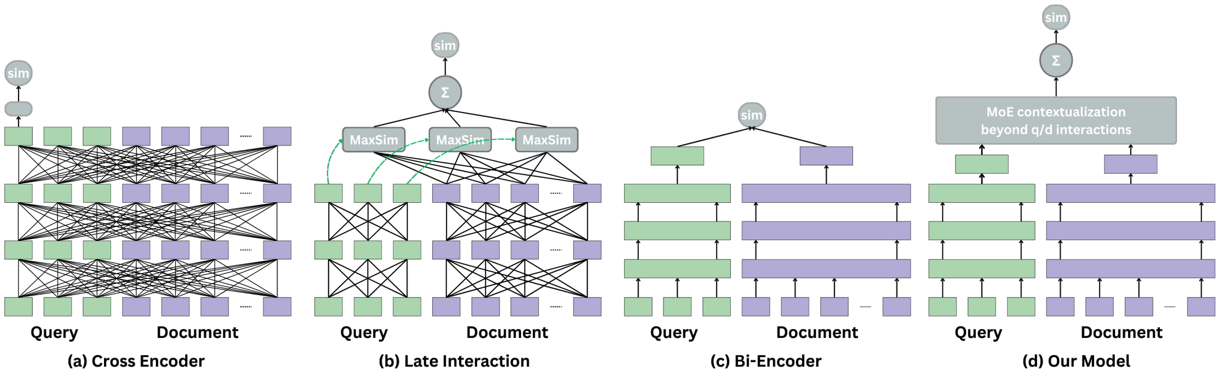

Figure 2: Schematic diagrams illustrating query/document matching neural IR models. The figure contrasts existing approaches (a), (b), and (c) (Khattab and Zaharia, 2020) with the architecture of DenseC3 (d).

mation (Miaschi et al., 2024). Other approaches exploit query/document interactions for embedding contextualization, either at the token-level (Liu et al., 2019; Khattab and Zaharia, 2020; Formal et al., 2021) or using full-text semantics (Pang et al., 2020; Zerveas et al., 2022), enriching representations with contextual and structural information. In this paper, we introduce a novel approach that contextualizes query and document embeddings beyond their interactions, through the notion of Cognitive Complexity.

By Cognitive Complexity of texts, we refer to their characterization based on their understandability and the intended usage of their content (Anderson and Krathwohl, 2001; Bai et al., 2023). To integrate this notion into embedding representations, we rely on Bloom's Taxonomy, originally developed to help educators structure learning materials (e.g., books, documents) and define targeted learning objectives. A learning objective is a clear statement that specifies what a learner is expected to understand or achieve after a learning experience. Consequently, Cognitive Complexity does not reflect a text's structural complexity or linguistic difficulty, but rather the conceptual difficulty of its content. It captures the required level of expertise for a user to understand the content of a document. Bloom's Taxonomy classifies texts into six different levels of increasing Cognitive Complexity, namely, *Remember, Understand, Apply, Analyze, Evaluate, and Create*. The *Remember* category represents the basic level of Cognitive Complexity and refers to texts that provide term definitions or event descriptions. When reading such texts, the learner is expected to only remember specific information. On the highest level of Cognitive Complexity stands the *Create* category, where the learner, after reading such documents, is expected to generate new ideas or produce original work. Academic papers are an example of documents that may belong to this category. To create contextualized query and document representations based on Cognitive Complexity, we introduce a dense retrieval architecture (Figure 1), which leverages a Mixture-of-Experts (MoE) framework (Jacobs et al., 1991) with a *supervised* gating mechanism (CLS). The CLS estimates the likelihood of a given document belonging to each of Bloom's Taxonomy levels and ensures that each expert specializes in a specific level. During training, our model learns to encode the Cognitive Complexity of queries and documents into their embeddings, aligning query representations closer to those of documents with similar complexity. We evaluate our approach on three IR tasks across seven benchmarks, showcasing the effectiveness of Cognitive Complexity-aware embeddings in dense retrieval.

The contributions of this work are twofold. First, we integrate into a DRM a modular MoE framework, which leverages Cognitive Complexity to contextualize query and document embeddings and improve similarity estimation (Section 2.2.1). We show that this approach significantly enhances DRM performance compared to non-contextualized embeddings and existing query/document interaction contextualization methods (Section 4). Second, we introduce a novel query/document matching paradigm that combines a bi-encoder architecture with a late-interaction strategy at the full embedding level. Our approach leverages MoE to enhance the query and document embedding contextualization beyond their interactions, by injecting external information about Cognitive Complexity (Section 2.2.2).

2 The Proposed Methodology

This Section presents the motivations that led to the identification of Cognitive Complexity as a contextualization factor in dense retrieval (Section 2.1). It also details DenseC3's architecture (Section 2.2).

2.1 Motivation

Prior work has shown that injecting linguistic features (Miaschi et al., 2024) and semantic information derived from query/document interactions into textual embeddings enhances the quality of the resulting representations and improves the model's performance on downstream NLP and IR tasks (Khattab and Zaharia, 2020; Pang et al., 2020; Formal et al., 2021; Luan et al., 2021; Zerveas et al., 2022; Yang, 2024). We argue that Cognitive Complexity is a textual characteristic that reflects the level of conceptual difficulty in a text and can be used to contextualize embeddings. Our hypothesis is based on the assumption that queries defined by expert users and expressing complex information should require documents of high Cognitive Complexity to be effectively answered (Sokli et al., 2024b). For example, a complex query in the medical domain is more likely to be better answered by documents such as academic publications, which typically exhibit higher Cognitive Complexity. A more generic query in the same domain could instead be adequately addressed by documents with lower complexity, as it demands less specialized or analytical content. Our model leverages Cognitive Complexity as defined in Bloom's Taxonomy to categorize and represent documents according to their conceptual difficulty. To estimate similarity, the query representation is adapted based on the complexity level of the document it is compared with. Since the model is trained in this way, we hypothesize that relevance estimation is most accurate when the Cognitive Complexity levels of the query and document are aligned.[1] DenseC3 employs a MoE framework, where each expert is specialized in a specific Cognitive Complexity level, under the assumption that these specialized experts produce higher-quality representations than a single, general-purpose encoder. Our empirical evaluation (Section 4) validates our intuition, showing that our model achieves better retrieval performance than approaches using non-contextualized embeddings and not Cognitive Complexity-aware query/document representations.

2.2 Incorporating Cognitive Complexity of Texts in Dense Retrieval through MoE

This section details DenseC3's architecture and training process (2.2.1). We also describe how Cognitive Complexity can enrich query and document embeddings and position their representations into the contextualized dense vector space(s) (2.2.2).

2.2.1 DenseC3

Figure 1 illustrates the model's architecture, which consists of four components: (1) an *underlying DRM* that encodes text inputs and creates embeddings capturing full-text semantics rather than token-level information; (2) *the gating mechanism*, a multi-head BERT multi-label classifier trained to classify texts into the six levels of Cognitive Complexity provided by Bloom's Taxonomy; (3) the six *experts* each trained to specialize in one of the six different levels of Cognitive Complexity; and (4) *the pooling module* used in the final stage to aggregate the experts' outputs and produce a single Cognitive Complexity-aware embedding to be used for similarity estimation between queries and documents. The *gating mechanism* (CLS - Figure 1) takes the document text as input and outputs a six-dimensional vector, where the weights represent the likelihood of the document to be characterized by each of the six Cognitive Complexity levels of Bloom's Taxonomy. For its implementation, we independently train a multi-label text classifier following the approach of Li et al. (2022), by using their publicly available external datasets. Once trained, the classifier is employed as a frozen module, acting as the gating mechanism to estimate the Cognitive Complexity of documents in an offline setting. The gating mechanism operates in conjunction with the pooling module to effectively guide the experts and the representation learning process. This process is described in Section 2.2.2, where we explain how a document's Cognitive Complexity drives the creation of both query and document representations, ensuring their alignment within the same complexity space. *The experts* receive either the query (q') or the document (d') embedding as input. These embeddings are generated by an *underlying DRM* (e.g., BERT-Base), which is employed as a bi-encoder. For this reason, the document embeddings are computed once, offline. Since the gating and the pooling mechanisms ensure that em-

[1] In our experiments, we also evaluate a query representation computed as the average of six separate representations, each corresponding to one of the six Cognitive Complexity levels defined by Bloom's Taxonomy (see Section 2.2.2).

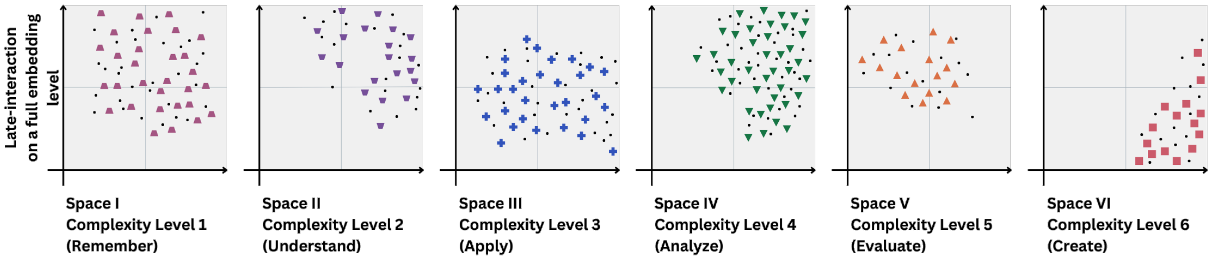

Figure 3: The *six distinct dense vector spaces* of different complexity levels as expected to be formed by the specialization of the experts. Each space contains documents of the same complexity. Each black dot depicts a unique representation *of the same query*, driven by and contextualized with each document's Cognitive Complexity.

beddings of the same Cognitive Complexity level are routed to the corresponding expert, each expert specializes in processing inputs of a specific level of Bloom's Taxonomy. Accordingly, our model includes six experts—one for each Cognitive Complexity level. For instance, $Expert1$ processes embeddings of texts at the *Remember* level, $Expert2$ for *Create*, and so on. Consequently, each query and document will have six representations. These must then be aggregated to form a single, contextualized representation for each query and document, which will be used for the semantic similarity estimation. This aggregation is performed by the *pooling module*, which defines how the experts' outputs are combined. We explore two pooling strategies, resulting in two distinct retrieval settings detailed in Section 2.2.2.

2.2.2 Cognitive Complexity-Aware Retrieval: Aligning Query and Document Representations

This section details how our model's gating and pooling modules interact to generate Cognitive Complexity-aware query and document representations for semantic similarity estimation in dense retrieval. We introduce two distinct pooling strategies, each defining a unique retrieval setting.

The first, called ***DenseC3_top1***, produces six separate dense vector spaces (one for each Cognitive Complexity level) by selecting solely the representation corresponding to the most likely complexity level as determined by the gating mechanism. Every document is embedded only in the space associated with the expert corresponding to its Cognitive Complexity (Figure 3). For example, an academic paper is expected to be represented solely in *Space VI*, as it exhibits the highest level of Cognitive Complexity. Formally, let x be the document's embedding and $f_i(x)$ the output of function f_i learned by the i-th expert. If $g_i(x)$ is the weight assigned to the i-th expert by the gating mechanism given the input document x, then:

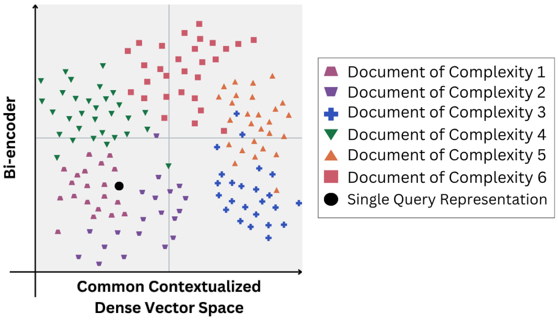

Figure 4: The *common contextualized space*. Each document representation is the *weighted* sum of all experts' outputs, maintaining Cognitive Complexity contextualization. The black dot depicts the query representation derived from the *averaged* experts' outputs.

$$m = \arg\max_{i=1,\ldots,N}(g_i(x)), \tag{1}$$

$$y = f_m(x), \tag{2}$$

where N is the total number of experts (in our case $N = 6$), and m represents the document's assigned Cognitive Complexity level. This leads to the final representation y of the given document, which is the one outputted by the expert that specializes in the selected Cognitive Complexity level. To estimate retrieval scores, our model uses the query representation generated by the same expert function f_m that has been used to represent the document to be ranked. This ensures that the query representation is contextualized to match the document's assigned Cognitive Complexity level, aligning them for similarity estimation. Formally, each query z is represented by a unique contextualized embedding per document, $f_m(z)$, driven by the document's Cognitive Complexity level. The underlying assumptions of this retrieval setting are relatively strong. It enforces the assignment of each document to a single Cognitive Complexity level, assuming that its dominant Cognitive Complexity level is the most suitable for retrieval (I). It assumes that the gating mechanism accurately classifies the documents' Cognitive Complexity (II). Finally, the model presumes that semantic similarity is better estimated within Cognitive Complexity-specific spaces rather than a shared, global space. This setting can lead to significant computational overhead during inference, as it requires a unique query representation for each document being scored (III).

To cope with the above limitations, we introduced ***DenseC3_w***, where we exploit the experts' specialization in Cognitive Complexity *collectively to create a common contextualized dense vector space* (Figure 4). This approach removes the assumption that each document belongs to a single Cognitive Complexity level (I), and it represents a document as the *weighted sum* of the outputs from all experts rather than assigning it to a unique Cognitive Complexity level. Formally, the document representation is the weighted sum of the outputted embeddings from all six experts:

$$y = \sum_{i=1}^{N} f_i(x) \cdot g_i(x) \tag{3}$$

Therefore, *DenseC3_w* eliminates the need for separate spaces of Cognitive Complexity (III), as it positions all documents within a shared space while preserving their Cognitive Complexity distinctions through weighted expert contributions. Nonetheless, the model still assumes that the gating mechanism provides meaningful complexity weights (II) and that documents of the same Cognitive Complexity will cluster together in the shared space. Additionally, this approach assigns to each query a single representation by averaging the outputs of all experts. Since queries can convey complex information needs that may not reflect the user's exact Cognitive Complexity level on the topic, averaging helps mitigate this issue. Formally, the query representation is the average of all six experts' outputs:

$$y = \sum_{i=1}^{N} \frac{f_i(z)}{N} \tag{4}$$

This retrieval strategy enables greater flexibility and generalizability in representation learning, while still incorporating Cognitive Complexity into semantic similarity estimation[2].

Figure 2d illustrates how DenseC3 combines both bi-encoder and late query/document interaction paradigms: the *DenseC3_w* variant works as a bi-encoder by independently encoding queries and documents into a shared dense space; the *DenseC3_top1* variant additionally incorporates late interactions during inference by conditioning query embeddings on the Cognitive Complexity of the document being ranked. Computational aspects of both variants are further discussed in Section 6.

3 Experiments

In this section, we report the empirical evaluation conducted to address the following research questions (RQs): **RQ1.** How Cognitive Complexity-specific spaces (*DenseC3_top1*) do compare to a shared contextualized space (*DenseC3_w*) in terms of retrieval effectiveness? **RQ2.** How does incorporating Cognitive Complexity into both query and document embeddings impact relevance estimation in DRMs compared to standard embeddings? **RQ3.** Are the observed improvements in retrieval effectiveness directly attributable to the use of Cognitive Complexity-aware embeddings?

To address the above research questions, we conduct an empirical evaluation on three IR tasks: passage retrieval, open-domain Q&A (following the formulation as a search task proposed by Chen et al. (2017)), and domain-specific search. The experimental framework uses seven publicly available benchmarks (Table 1) from the TREC Deep Learning Tracks (TREC DL 19 (Craswell et al., 2020) & 20 (Craswell et al., 2021)), the BEIR Collection proposed by Thakur et al. (2021) (MS-MARCO (Nguyen et al., 2016), NQ (Kwiatkowski et al., 2019), and HotpotQA (Yang et al., 2018)), and a multi-domain benchmark for search evaluation proposed by Bassani et al. (2022) (PS and CS). These datasets contain queries and documents of diverse Cognitive Complexity levels, from fact-based (NQ) to reasoning-intensive questions (HotpotQA) and from general-domain passages (MS-MARCO, TREC DL 19 & 20)[3] to specialized academic texts of the Plotical & Computer Science domains (PS and CS). This diversity makes them suitable for evaluating the effectiveness of Cognitive Complexity-aware representations in improving query/document matching in DRMs.

[2] Refer to Appendix D for additional illustrations of the embedding space formulation.

[3] These three benchmarks share the same corpus, but differ in their query sets and relevance judgments (see Table 1).

3.1 Experimental Setup

This section presents the experimental setup used in our study. The code is publicly available for reproducibility[4]. To estimate the Cognitive Complexity of documents, we independently trained the CLS module (Figure 1) on external datasets annotated with Bloom's Taxonomy labels and evaluated its performance on their respective test sets (refer to Appendix A for further details on the CLS performance). The CLS effectively distinguished among all six Cognitive Complexity levels, exhibiting robust classification performance. However, since the employed experimental datasets do not include explicit Cognitive Complexity labels, we cannot directly assess the classifier's accuracy. Nevertheless, the classification distributions presented in Figures 5 & 6 follow an expected pattern based on the characteristics of each dataset. For instance, collections with academic documents are predominantly classified into the highest Cognitive Complexity levels, which aligns with our expectations (more details are provided in Appendix A).

Architecture Details. We used four different models, namely TinyBERT (14.5M, Jiao et al. (2020)), BERT (110M, Devlin et al. (2019)), Contriever (110M, Izacard et al. (2022)), and ColBERT (110M, Khattab and Zaharia (2020)) to serve as underlying DRMs and encode text inputs to create full-text embeddings. The MoE architecture consists of six experts corresponding to the six Cognitive Complexity levels of Bloom's Taxonomy and a skip connection. Following Houlsby et al. (2019), each expert consists of a Feed-Forward Network (FFN) with a down-projection layer that reduces the input dimension by half, followed by an up-projection FFN layer that restores the dimensionality to match the original input embedding.

Training. We configure the training batch size to 64, setting the learning rate at 10^{-6} for the underlying model and 10^{-4} for the experts. TinyBERT is trained for 30 epochs (GPU hours: 5-10 mins/epoch) across all datasets, due to its small size. For the remaining models, training is conducted for 20 epochs (GPU hours: 20-60 mins/epoch) across all datasets, except for Computer Science, where training is limited to 10 epochs (GPU hours: ~2 hrs/epoch) due to the large number of training queries (approximately 3.5% more than the second-largest dataset). We employ 5% of the training data for validation and retain the checkpoint with the lowest validation loss. We set the random seed to 42 and use contrastive loss (Izacard et al., 2022) with a temperature of 1 for all models except Contriever, where the authors report an optimal temperature of 0.05. To train the *DenseC3_top1* variant, we used Top-1 gating for both queries and documents. The same setting (Top-1 gating for both queries and documents) applies during inference, which leads to the creation of the six distinct dense vector spaces (Figure 3) and separates documents based on their Cognitive Complexity level. We trained the *DenseC3_w* variant by selecting the document representation of the Top-1 expert, while for the query we used the averaged representations of the six experts[5]. However, during inference, while we keep the averaged query representations, for the documents, we use the *weighted sum* of the six experts' outputs. We select this approach to allow for the unification of the six spaces (Figure 4) while maintaining the encoding of the documents' Cognitive Complexity into their embeddings.

Table 1: Dataset Statistics. The average number of relevance judgments per query is indicated in parentheses.

Dataset	Corpus Size	Train Queries	Test Queries
MSMARCO v1 passage corpus (MSMARCO)	8.8M passages	532k (1.1 avg rel.)	7k
TREC Deep Learning Track 2019 (TREC DL 19)	8.8M passages	503k (215 avg rel.)	43
TREC Deep Learning Track 2020 (TREC DL 20)	8.8M passages	503k (211 avg rel.)	54
Natural Questions (NQ)	2.6M passages	132k (1.2 avg rel.)	3.5k
HotpotQA	5.2M documents	85k (2 avg rel.)	7.4k
Political Science (PS)	4.8M documents	160k (3.8 avg rel.)	5.7k
Computer Science (CS)	4.8M documents	550k (3.25 avg rel.)	6.5k

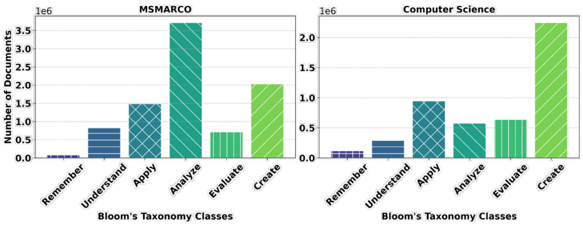

Figure 5: Distributions of documents across Cognitive Complexity levels, based on our independently trained classifier (CLS - Figure 1). Additional graphs are shown in Figure 6.

3.2 Metrics & Baselines

We evaluate the experimental results using NDCG@10, the designated metric for model evaluation in the employed TREC (Craswell et al., 2020, 2021) and BEIR (Kamalloo et al., 2024) collections. We also report Recall@100, a metric

[4] https://github.com/FaySokli/DenseC3

[5] We also trained a variant with weighted sum document and averaged query representations, but found that it did not train effectively. This aligns with related studies showing that training MoE with Top-1 gating can outperform training with averaged representations (Shazeer et al., 2017).

Table 2: Results on all datasets. Metrics refer to Recall@100 and nDCG@10. Symbol * indicates a statistically significant difference over *fine-tuned*, calculated using the ASPIRE toolkit (Peikos et al., 2024). Best results for each dataset are in **bold**.

		TinyBERT					BERT					Contriever					ColBERT				
		fine-tuned	*random*	*unsupervised*	*DenseC3_top1*	*DenseC3_w*	*fine-tuned*	*random*	*unsupervised*	*DenseC3_top1*	*DenseC3_w*	*fine-tuned*	*random*	*unsupervised*	*DenseC3_top1*	*DenseC3_w*	*fine-tuned*	*random*	*unsupervised*	*DenseC3_top1*	*DenseC3_w*
MSMARCO	*recall*	.682	.686	.688*	.666*	**.689***	.791	.788*	.790	.777*	**.793**	**.850**	.838*	.839*	.819*	**.850**	.820	.818	.821	.817	**.822**
	nDCG	.244	.243	.246	.234*	**.249***	.293	.286*	.290*	.277*	**.294**	.321	.310*	.309*	.290*	**.323**	.309	.310	.312*	.307	**.314***
TREC DL 19	*recall*	.421	.418	.422	.400	**.435**	**.531**	.509	.530	.491*	.527	**.605**	.592	.584	.586	.602	.549	.552	.554	.544	**.562**
	nDCG	.449	.444	.440*	**.452**	**.452**	.507	**.508**	.506	.468*	.501	.540	.533	.545	.504*	**.556***	.567	.565	.571	.566	**.573**
TREC DL 20	*recall*	.521	.516	.516	.506	**.522**	.614	.615	.620	.603	**.621**	.678	.665	.683	.659	**.681**	.628	.622	.628	.629	**.632**
	nDCG	.466	.466	.468	.454*	**.479***	.516	.508*	.516	.473*	**.518**	.544	.504*	.542	.508*	**.547**	.579	.574	.574	.560*	**.580**
NQ	*recall*	.722	.726	.730*	.704*	**.743***	.879	.873	.874	.879	**.886***	**.933**	.926	.932	.928	**.933**	.884	.877	.888	.885	**.893***
	nDCG	.261	.262	.263	.256	**.287***	.353	.350	.346*	.362*	**.374***	.426	.403	.416*	.414*	**.428**	.383	.372	.375	.381	**.402***
HotpotQA	*recall*	.463	.468	.472*	.446*	**.473***	.700	.703	.689*	.690*	**.707***	**.862**	.855	.861	.839*	.842*	.684	.680	.683	.678	**.692***
	nDCG	.240	.243	.248*	.223	**.249***	.441	.448	.432*	.433*	**.458***	**.672**	.653	.667*	.624*	.630*	.419	.415	.416	.411	**.437***
PS	*recall*	.275	.274	.283*	.278	**.294***	.384	.382	.387	.389*	**.402***	**.483**	.471*	**.483**	.466*	.476*	.385	.375*	.378*	.388	**.400***
	nDCG	.136	.134	.140*	.142*	**.151***	.192	.189	.194	.200	**.207***	**.251**	.243*	**.251**	.243*	**.251**	.200	.196	.196	.204	**.212***
CS	*recall*	.319	.322	.326*	.305*	**.332***	.383	.375*	.376*	.362*	**.397***	.437	.435	.438	.418*	**.439**	.395	.393	.392	.376*	**.412***
	nDCG	.161	.159	.163	.159	**.175***	.192	.190	.188*	.187*	**.205***	.224	.224	.223	.213	**.225**	.201	.199	.197	.194	**.213***

commonly used in the literature for the evaluation of IR models on the selected datasets (Hashemi et al., 2023; Kasela et al., 2024; Rau and Kamps, 2024). Statistical significance has been evaluated based on two-sided paired Student's t-tests with Bonferroni multiple testing correction, at significance levels of 0.05. To address RQ1, we compare the performance of the two variants between them (*DenseC3_top1* & *DenseC3_w*). For RQ2, we consider as baseline the fine-tuned version of the underlying DRM (*fine-tuned*), trained on the available dataset-specific training data using the same hyperparameters as our models. For RQ3, we reproduce experiments with unsupervised training of the experts as proposed by Sokli et al. (2024a) (*unsupervised*). This approach relies on a gating mechanism operating without any awareness of Cognitive Complexity trained in an unsupervised manner along with the experts. Additionally, we implement a baseline in which the gating mechanism assigns random weights to the experts during training (*random*).

4 Results & Discussion

This section presents the results of the empirical evaluations conducted to address our research questions. Table 2 compares the performance of the *DenseC3_top1* and *DenseC3_w* variants in terms of retrieval effectiveness against various baselines on four different underlying DRMs.

RQ1. The obtained results show that *DenseC3_w* consistently outperforms *DenseC3_top1* across all retrieval settings, achieving significant performance gains up to 9.67% in Recall@100 and 12.10% in nDCG@10. Notably, the most significant improvements are observed on the CS collection, where the *DenseC3_w* variant outperforms *DenseC3_top1* across all four models, with gains ranging from 5.05% to 9.67%. The observed performance gains can be attributed to several factors. Firstly, *DenseC3_w* mitigates the impact of classification errors by distributing representations across multiple Cognitive Complexity levels. *DenseC3_top1* relies on the gating mechanism to assign documents to a single Cognitive Complexity level, making misclassifications more detrimental. Moreover, *DenseC3_w* allows for both query and document embeddings to capture multiple levels of Cognitive Complexity. This leads to a common contextualized dense space during inference, which improves semantic alignment between queries and documents. In contrast, *DenseC3_top1* confines embeddings to separate Cognitive Complexity-specific spaces, potentially limiting cross-complexity generalization. Additional experiments that we have conducted further strengthen the finding that integrating a MoE framework into a DRM leads to higher retrieval performance when all experts are used to compute the final query and document representations (see Appendix B). Our findings suggest that unifying the six Cognitive Complexity spaces into a shared contextualized dense vector space enhances the effectiveness of Cognitive Complexity-aware representations for relevance estimation in DRMs.

RQ2. To answer RQ2, we compare the standard *fine-tuned* versions of the underlying DRMs (non-Cognitive Complexity-aware embeddings) with our model (Cognitive Complexity-aware embeddings). We observe that *DenseC3_w* consistently yields greater retrieval performance across all seven benchmarks, on both evaluation metrics, and for three of the four DRMs employed. The most notable gains are marked on TinyBERT in nDCG@10, ranging from a 2.05% to 11.03% increase for all collections except TREC DL 19 and HotpotQA. When integrated with Contriever, *DenseC3_w* resulted in marginal gains in four datasets and slight performance degradations in others. This can be attributed to Contriever's unsupervised contrastive pre-training on mined hard negative text pairs, which may already capture semantic relationships effectively. In contrast, models like TinyBERT, BERT, and ColBERT leverage labeled data or knowledge distillation, potentially allowing greater benefit from additional contextualization introduced by *DenseC3_w*. Nevertheless, as shown in Table 2 (comparing the *unsupervised* and *DenseC3_w* columns), Cognitive Complexity-aware embeddings still benefit Contriever, as our model matches or surpasses the *unsupervised* baseline across all collections except HotpotQA. This observation suggests that the Cognitive Complexity supervision positively impacts Contriever's performance in terms of retrieval effectiveness. Further details on our model's performance when integrated with Contriever are discussed in Section 6.

RQ3. To further investigate whether the improved retrieval performance of *DenseC3_w* can be genuinely attributed to Cognitive Complexity-aware embeddings, we compare it against two additional baselines: *random* and *unsupervised*. This comparison helps to isolate the impact of Cognitive Complexity-aware representations and the role of the gating mechanism. We observe that the *random* baseline is mostly on par with *fine-tuned*, indicating that simply extending the Transformer architecture with additional MoE layers without any specific training strategy or supervision of the gating mechanism and the experts does not enrich the embeddings nor improve retrieval effectiveness. Comparing the *DenseC3_w* and *unsupervised* columns of Table 2, we observe consistent performance gains. These results suggest that the selected supervision strategy actually contextualizes embeddings with useful information, resulting in higher-quality representations for relevance estimation. Our findings showcase that Cognitive Complexity-aware embeddings as derived from *DenseC3_w* outperform both the *fine-tuned* and *unsupervised* MoE baselines.

5 Related Work

Neural IR models can exhibit strong retrieval performance; however, depending on their architecture, this often entails substantial computational overhead (Mitra et al., 2018). Cross-encoders (Figure 2a) exploit early token-level query/document interactions to create a unified representation of the query and document. While effective, this approach is computationally expensive, hence limited to multi-stage retrieve-then-rerank pipelines (Rosa et al., 2022). In contrast, DRMs based on bi-encoders (Figure 2c) disregard all query/document interactions, which allows for the independent creation of query and document embeddings and the usage of these models directly as first-stage retrievers (Yu et al., 2022; Izacard et al., 2022). However, the lack of interactions prevents embedding contextualization, potentially reducing effectiveness. Late-interaction models (Figure 2b) retain a bi-encoder structure while introducing token-level query/document interactions at retrieval time (Khattab and Zaharia, 2020). Other models achieve embedding contextualization by exploiting query/document interactions on a full embedding level (Pang et al., 2020; Zerveas et al., 2022). These approaches improve the embeddings without sacrificing efficiency but rely solely on the query/document interactions for their contextualization. In our work, we achieve embedding contextualization by introducing a novel late query/document interaction strategy based on a modular MoE framework, which operates at the full embedding level. Our approach enriches both query and document representations with information about Cognitive Complexity before similarity estimation. Prior research in IR has leveraged MoE for tasks such as passage retrieval (Cai et al., 2023; Ma et al., 2023) and Q&A (Dai et al., 2023; Kasela et al., 2024; Shen et al., 2024). These studies exploit MoE in two ways. One approach integrates experts within the feed-forward block of the Transformer model's layers. While beneficial for the underlying model, this substantially increases parameter count and only permits a token-level embedding contextualization. Another approach partially expands the underlying DRM by applying a MoE architecture solely on the outputted query embedding. Our model improves dense retrieval by contextualizing both query and document representations within Cognitive Complexity space(s), enhancing query/document alignment. Unlike prior MoE approaches, it applies global embedding-level adjustments rather than token-wise refinements, introduces a single MoE module for efficiency, and employs a supervised gating mechanism to ensure well-defined expert specialization, yielding a more interpretable and effective retrieval process.

6 Conclusions

In this work, we leverage the Cognitive Complexity of texts as defined in Bloom's Taxonomy to contextualize DRM query and document representations. Our approach leverages MoE for contextualization at the full embedding level, ensuring that queries and documents are aligned within Cognitive Complexity-aware space(s). We evaluated two retrieval settings: *DenseC3_top1* that enforces Cognitive Complexity-specific spaces, and *DenseC3_w* that creates a unified contextualized space by aggregating expert representations. Results show that *DenseC3_w* consistently outperforms standard fine-tuned baselines and other MoE approaches. Our model introduces a novel late-interaction strategy that combines bi-encoder efficiency with query/document interactions for the contextualization of embeddings with Cognitive Complexity. Our findings highlight the benefits of injecting external linguistic information into DRMs, suggesting that Cognitive Complexity-aware embeddings improve retrieval performance across multiple IR tasks.

Limitations

The *DenseC3_w* variant exhibits strong retrieval effectiveness in passage retrieval, open-domain Q&A, and domain-specific search, outperforming fine-tuned baselines and existing MoE-based approaches (see Table 2 in Section 4). However, it eliminates explicit Cognitive Complexity-specific space separation, which may be beneficial in scenarios such as educational search or personalized IR models with a focus on user expertise, where preserving distinct Cognitive Complexity levels can potentially improve retrieval. The *DenseC3_top1* variant, on the other hand, ensures strict query/document alignment based on Cognitive Complexity levels, enhancing retrieval in specialized settings. However, this approach does not fully leverage the benefits of the bi-encoder architecture and increases query latency, due to the introduced query/document interactions during inference, making it computationally expensive for large-scale and real-time retrieval tasks. Nonetheless, document representations can still be computed independently and stored offline, retaining the key efficiency advantages of a bi-encoder. Hence, this variant appears to have potential to be used as a re-ranker, although its effectiveness in this capacity is yet to be investigated. Moreover, enforcing Cognitive Complexity-specific vector spaces introduces strong retrieval assumptions, which may not generalize well across diverse datasets. Such retrieval settings could be open-domain collections (e.g., NQ, HotpotQA - Table 2), where the document distributions showcase stronger imbalances among Cognitive Complexity levels (see Appendix A for further details).

When integrated with Contriever, *DenseC3_w* yielded only marginal improvements on certain datasets and even slight decreases on others (columns *fine-tuned* and *DenseC3_w* - Table 2). This outcome may be linked to Contriever's distinct pre-training strategy and hyperparameter configurations, which set it apart from the other DRMs employed (i.e., TinyBERT, BERT, and ColBERT). For instance, the reported optimal temperature for Contriever (0.05) differs notably from that of the other models (1). These discrepancies suggest that the hyperparameter optimization of our model could be sub-optimal for Contriever. As future work, we intend to further investigate the unique behavior of our model with Contriever and propose a tailored experimental setup for this setting.

Both of the proposed variants rely on a classifier (CLS - Figure 1) pre-trained on external datasets to estimate the Cognitive Complexity of documents. Yet the benchmarks we used for evaluation lack explicit Cognitive Complexity labels, preventing the assessment of the classifier's accuracy on these collections. Potential misclassifications could lead to suboptimal expert assignments, negatively affecting the retrieval quality. Annotating additional collections based on their Cognitive Complexity could overcome this issue. We also observed an underrepresentation of two out of six Bloom's Taxonomy levels in the selected datasets, namely *Remember* and *Understand*. *Remember* is significantly underrepresented, while *Understand* has limited presence, as shown in Figure 5. An overview of document distributions across datasets is also presented in Figure 6 of Appendix A. This imbalance may limit the contributions of experts specialized in these two Cognitive Complexity levels to the model's overall performance. To address this limitation, we aim to identify collections with balanced Cognitive Complexity level representation to evaluate our model.

Additionally, our approach demands significant computational resources, particularly for large datasets like CS, making it less feasible for low-resource environments. Since our model is fine-tuned separately for each dataset, it requires a large number of training queries (minimum 85k), increasing training costs and limiting its generalization capabilities. To overcome this issue, we are working towards the direction of exploiting a single collection for the training of our model (i.e., MS-MARCO) and evaluating it on other collections in a zero-shot setting. Preliminary empirical evidence shows the potential of this approach to improve retrieval effectiveness while reducing computational costs (see Appendix C for further details).

Acknowledgments

This work has received funding from the European Union's Horizon Europe research and innovation programme under the Marie Skłodowska-Curie grant agreement No 101073307. This work was partially supported by the MUR under the grant "Dipartimenti di Eccellenza 2023-2027" of the Department of Informatics, Systems and Communication of the University of Milano-Bicocca, Italy.

References

Lorin W. Anderson and David R. Krathwohl. 2001. A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives. Longman, New York.

Yushi Bai, Jiahao Ying, Yixin Cao, Xin Lv, Yuze He, Xiaozhi Wang, Jifan Yu, Kaisheng Zeng, Yijia Xiao, Haozhe Lyu, Jiayin Zhang, Juanzi Li, and Lei Hou. 2023. Benchmarking foundation models with language-model-as-an-examiner. In Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023.

Elias Bassani, Pranav Kasela, Alessandro Raganato, and Gabriella Pasi. 2022. A multi-domain benchmark for personalized search evaluation. In Proceedings of the 31st ACM International Conference on Information & Knowledge Management, CIKM '22, page 3822–3827, New York, NY, USA. Association for Computing Machinery.

Yinqiong Cai, Yixing Fan, Keping Bi, Jiafeng Guo, Wei Chen, Ruqing Zhang, and Xueqi Cheng. 2023. CAME: competitively learning a mixture-of-experts model for first-stage retrieval. CoRR, abs/2311.02834.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers, pages 1870–1879. Association for Computational Linguistics.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2021. Overview of the TREC 2020 deep learning track. CoRR, abs/2102.07662.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. 2020. Overview of the TREC 2019 deep learning track. CoRR, abs/2003.07820.

Damai Dai, Wenbin Jiang, Jiyuan Zhang, Yajuan Lyu, Zhifang Sui, and Baobao Chang. 2023. Mixture-of-experts for biomedical question answering. In Natural Language Processing and Chinese Computing - 12th National CCF Conference, NLPCC 2023, Foshan, China, October 12-15, 2023, Proceedings, Part I, volume 14302 of Lecture Notes in Computer Science, pages 604–615. Springer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), pages 4171–4186. Association for Computational Linguistics.

Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021. SPLADE: sparse lexical and expansion model for first stage ranking. In SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021, pages 2288–2292. ACM.

Helia Hashemi, Yong Zhuang, Sachith Sri Ram Kothur, Srivas Prasad, Edgar Meij, and W. Bruce Croft. 2023. Dense retrieval adaptation using target domain description. In Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval, ICTIR 2023, Taipei, Taiwan, 23 July 2023, pages 95–104. ACM.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA, volume 97 of Proceedings of Machine Learning Research, pages 2790–2799. PMLR.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. Trans. Mach. Learn. Res., 2022.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. 1991. Adaptive mixtures of local experts. Neural Comput., 3(1):79–87.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. TinyBERT: Distilling BERT for natural language understanding. In Findings of the Association for Computational Linguistics: EMNLP 2020, pages 4163–4174, Online. Association for Computational Linguistics.

Ehsan Kamalloo, Nandan Thakur, Carlos Lassance, Xueguang Ma, Jheng-Hong Yang, and Jimmy Lin. 2024. Resources for brewing BEIR: reproducible reference models and statistical analyses. In Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Washington DC, USA, July 14-18, 2024, pages 1431–1440. ACM.

Pranav Kasela, Gabriella Pasi, Raffaele Perego, and Nicola Tonellotto. 2024. DESIRE-ME: domain-enhanced supervised information retrieval using mixture-of-experts. In Advances in Information Retrieval - 46th European Conference on Information Retrieval, ECIR 2024, Glasgow, UK, March 24-28, 2024, Proceedings, Part II, volume 14609 of Lecture Notes in Computer Science, pages 111–125. Springer.

Omar Khattab and Matei Zaharia. 2020. 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, SIGIR 2020, Virtual Event, China, July 25-30, 2020, pages 39–48. ACM.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural Questions: A Benchmark for Question Answering Research. Transactions of the Association for Computational Linguistics, 7:453–466.

Yuheng Li, Mladen Rakovic, Boon Xin Poh, Dragan Gasevic, and Guanliang Chen. 2022. Automatic classification of learning objectives based on bloom's taxonomy. In Proceedings of the 15th International Conference on Educational Data Mining, EDM 2022, Durham, UK, July 24-27, 2022. International Educational Data Mining Society.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. CoRR, abs/1907.11692.

Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2021. Sparse, dense, and attentional representations for text retrieval. Trans. Assoc. Comput. Linguistics, 9:329–345.

Guangyuan Ma, Xing Wu, Peng Wang, and Songlin Hu. 2023. Cot-mote: Exploring contextual masked autoencoder pre-training with mixture-of-textual-experts for passage retrieval. CoRR, abs/2304.10195.

Alessio Miaschi, Felice Dell'Orletta, and Giulia Venturi. 2024. Linguistic knowledge can enhance encoder-decoder models (if you let it). In Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy, pages 10539–10554. ELRA and ICCL.

Bhaskar Mitra, Nick Craswell, et al. 2018. An introduction to neural information retrieval. Foundations and Trends® in Information Retrieval, 13(1):1–126.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016, volume 1773 of CEUR Workshop Proceedings. CEUR-WS.org.

Liang Pang, Jun Xu, Qingyao Ai, Yanyan Lan, Xueqi Cheng, and Jirong Wen. 2020. Setrank: Learning a permutation-invariant ranking model for information retrieval. In Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020, pages 499–508. ACM.

Georgios Peikos, Wojciech Kusa, and Symeon Symeonidis. 2024. ASPIRE: assistive system for performance evaluation in IR. CoRR, abs/2412.15759.

David Rau and Jaap Kamps. 2024. Query generation using large language models - A reproducibility study of unsupervised passage reranking. In Advances in Information Retrieval - 46th European Conference on Information Retrieval, ECIR 2024, Glasgow, UK, March 24-28, 2024, Proceedings, Part IV, volume 14611 of Lecture Notes in Computer Science, pages 226–239. Springer.

Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1994. Okapi at TREC-3. In Proceedings of The Third Text REtrieval Conference, TREC 1994, Gaithersburg, Maryland, USA, November 2-4, 1994, volume 500-225 of NIST Special Publication, pages 109–126. National Institute of Standards and Technology (NIST).

Guilherme Moraes Rosa, Luiz Henrique Bonifacio, Vitor Jeronymo, Hugo Queiroz Abonizio, Marzieh Fadaee, Roberto A. Lotufo, and Rodrigo Frassetto Nogueira. 2022. No parameter left behind: How distillation and model size affect zero-shot retrieval. CoRR, abs/2206.02873.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. OpenReview.net.

Sheng Shen, Le Hou, Yanqi Zhou, Nan Du, Shayne Longpre, Jason Wei, Hyung Won Chung, Barret Zoph, William Fedus, Xinyun Chen, Tu Vu, Yuexin Wu, Wuyang Chen, Albert Webson, Yunxuan Li, Vincent Y. Zhao, Hongkun Yu, Kurt Keutzer, Trevor Darrell, and Denny Zhou. 2024. Mixture-of-experts meets instruction tuning: A winning combination for large language models. In The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024. OpenReview.net.

Effrosyni Sokli, Pranav Kasela, Georgios Peikos, and Gabriella Pasi. 2024a. Investigating mixture of experts in dense retrieval. CoRR, abs/2412.11864.

Effrosyni Sokli, Alessandro Raganato, and Gabriella Pasi. 2024b. Incorporating cognitive complexity of text in dense retrieval for personalized search. In Proceedings of the 14th Italian Information Retrieval Workshop, Udine, Italy, September 5-6, 2024, volume 3802 of CEUR Workshop Proceedings, pages 82–85. CEUR-WS.org.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogenous benchmark for zero-shot evaluation of information retrieval models. CoRR, abs/2104.08663.

Eugene Yang. 2024. Contextualization with SPLADE for high recall retrieval. In Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Washington DC, USA, July 14-18, 2024, pages 2337–2341. ACM.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Yue Yu, Chenyan Xiong, Si Sun, Chao Zhang, and Arnold Overwijk. 2022. COCO-DR: combating the distribution shift in zero-shot dense retrieval with contrastive and distributionally robust learning. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022, pages 1462–1479. Association for Computational Linguistics.

George Zerveas, Navid Rekabsaz, Daniel Cohen, and Carsten Eickhoff. 2022. CODER: an efficient framework for improving retrieval through contextual document embedding reranking. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022, pages 10626–10644. Association for Computational Linguistics.

A The Gating Mechanism

As outlined in Section 2.2.1, the *gating mechanism* is implemented as a multi-head BERT-based multi-label text classifier (CLS), which is trained once offline using the external datasets provided by Li et al. (2022) and remains frozen throughout all retrieval settings. Since the datasets used in our evaluation do not include explicit Cognitive Complexity labels (see Section 6), a direct assessment of the CLS on these collections is not feasible. To validate its effectiveness, we evaluated the trained CLS on the test sets of the external datasets and found its performance to be consistent with the results reported in the original work, across all levels and evaluation metrics. As shown in Table 3, the CLS is trained effectively and achieves F1-scores above 0.88 for all Cognitive Complexity levels, with an average F1-score of 0.913. These results suggest that the CLS is likely to perform adequately on the unlabeled collections as well.

Table 3: The Classifier's performance (CLS - acting as the gating mechanism in Figure 1) on the test sets of the external datasets of learning objectives provided by (Li et al., 2022).

	Remember	**Understand**	**Apply**	**Analyze**	**Evaluate**	**Create**
Precision	.840	.941	.910	.945	.929	.916
Recall	.932	.925	.934	.902	.930	.861
F1-score	.884	.933	.922	.923	.929	.888

Figure 6 shows the distributions of Cognitive Complexity levels, as predicted by the CLS, for the documents in our evaluation datasets. All three datasets from the BEIR collection (MSMARCO, NQ, and HotpotQA) are based on the same open-domain Wikipedia corpus (Thakur et al., 2021), which explains the similar distribution patterns noted. The observed distributions on these three collections are anticipated, given that open-domain collections contain a wide range of articles that can vary in conceptual difficulty. In contrast, the PS and CS collections consist of academic texts from

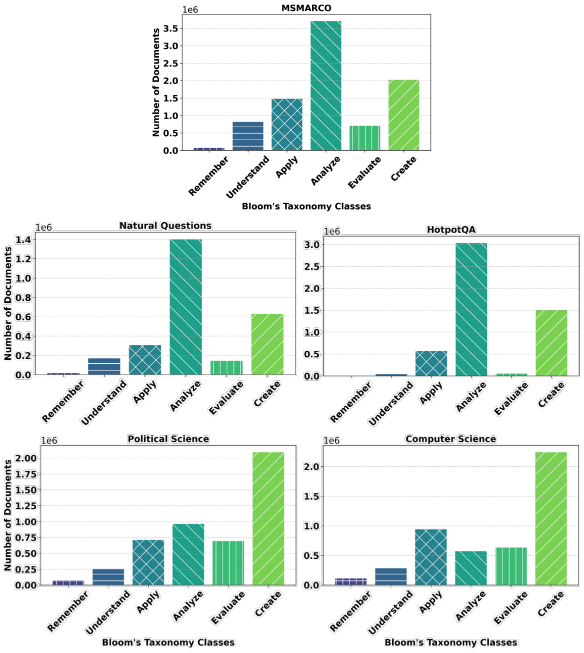

Figure 6: Documents' Cognitive Complexity distributions based on the independently trained CLS. Given that MSMARCO and TREC DL 19 & 20 share a common corpus, they are represented jointly in a single subfigure.

the domains of Political Science and Computer Science, respectively. Given the high conceptual difficulty of academic content, these datasets exhibit a predominance of documents classified to the highest level of Cognitive Complexity, in line with our expectations. Based on these observations, we can conclude that the CLS assigns Cognitive Complexity levels in a manner that aligns with the nature of the underlying document collections.

B Expert Aggregation Strategies

To further investigate RQ1 discussed in Section 4, we compare our model with an approach proposed in the literature (Sokli et al., 2024a) that trains the experts, gating mechanism, and base model together in an unsupervised manner. The authors report results for two different variants: SB-MoE$_{\text{TOP-1}}$ and SB-MoE$_{\text{ALL}}$, both trained using a Top-1 gating strategy and a temperature of 0.05. During inference, SB-MoE$_{\text{TOP-1}}$ retains the Top-1 strategy while SB-MoE$_{\text{ALL}}$ uses the weighted sum of all experts' outputs. In this work, we reproduced both variants, denoted as *unsupervised*TOP-1 and *unsupervised*ALL, and evaluated them on our datasets. Results presented in Table 4 show that the variant that leverages all experts (*unsupervised*$_{\text{ALL}}$) outperforms *unsupervised*$_{\text{TOP-1}}$, suggesting that aggregating all experts' outputs is more effective for retrieval than relying solely on the top expert. This performance gap suggests that aggregating all experts' outputs enables the model to better handle the diversity of user queries and document content. This outcome aligns with our findings (see Table 2 and Section 4), where *DenseC3_w* outperforms *DenseC3_top1*.

Table 4: Results for SB-MoE on all benchmarks. Metrics refer to Recall@100 and nDCG@10. For these experiments, we set the temperature to 1 for all models except Contriever, where the reported optimal temperature is 0.05. Best results for each dataset are in **bold**.

		unsupervised							
		TinyBERT		BERT		Contriever		ColBERT	
		TOP-1	ALL	TOP-1	ALL	TOP-1	ALL	TOP-1	ALL
MSMARCO	*recall*	**.688**	**.688**	**.790**	**.790**	**.839**	**.839**	.820	**.821**
	nDCG	.245	**.246**	.289	**.290**	**.309**	**.309**	**.313**	.312
TREC DL 19	*recall*	.421	**.422**	.528	**.530**	.582	**.584**	.553	**.554**
	nDCG	.438	**.440**	.502	**.506**	.539	**.545**	.562	**.571**
TREC DL 20	*recall*	**.520**	.516	.617	**.620**	**.683**	**.683**	.627	**.628**
	nDCG	.452	**.468**	.513	**.516**	.535	**.542**	.568	**.574**
NQ	*recall*	.727	**.730**	.871	**.874**	.930	**.932**	**.888**	**.888**
	nDCG	.260	**.263**	.343	**.346**	**.416**	**.416**	**.375**	**.375**
HotpotQA	*recall*	.460	**.472**	.686	**.689**	.853	**.861**	.680	**.683**
	nDCG	.232	**.248**	.425	**.432**	.653	**.667**	.411	**.416**
PS	*recall*	.276	**.283**	.385	**.387**	.479	**.483**	**.378**	**.378**
	nDCG	.137	**.140**	.193	**.194**	.250	**.251**	**.196**	**.196**
CS	*recall*	.322	**.326**	.374	**.376**	.435	**.438**	**.392**	**.392**
	nDCG	.161	**.163**	.185	**.188**	.222	**.223**	.195	**.197**

C Zero-shot Evaluation

In the main experiments reported in this paper, we fine-tuned our model separately on the training set of each dataset. While this approach is effective, it increases training costs and may be less practical in low-resource environments (a limitation already discussed in Section 6). For example, training on the largest collection (CS) with BERT-based retrieval models like ColBERT or Contriever requires approximately 2 hours per epoch on a single NVIDIA RTX 6000 Ada Generation GPU. In this appendix, we explore the performance of our model in a zero-shot retrieval setting as a step towards addressing this limitation.

Table 5: Preliminary results on the zero-shot evaluation of DenseC3 with two different DRMs on two BEIR collections, namely Natural Questions and HotpotQA. Best results for each model are indicated with **bold** text.

		NQ		HotpotQA	
Retriever	Variant	Recall@100	NDCG@10	Recall@100	NDCG@10
Contriever	*fine-tuned*	.896	.376	**.799**	**.650**
	DenseC3_w	**.901 (0,56%↑)**	**.379 (0,80%↑)**	.796 (0,38%↓)	.647 (0,46%↓)
ColBERT	*fine-tuned*	**.814**	**.341**	.584	.430
	DenseC3_w	.810 (0,49%↓)	.338 (0,88%↓)	**.593 (1,54%↑)**	**.443 (3,02%↑)**

While this remains an area of ongoing research, our preliminary results in Table 5 suggest that

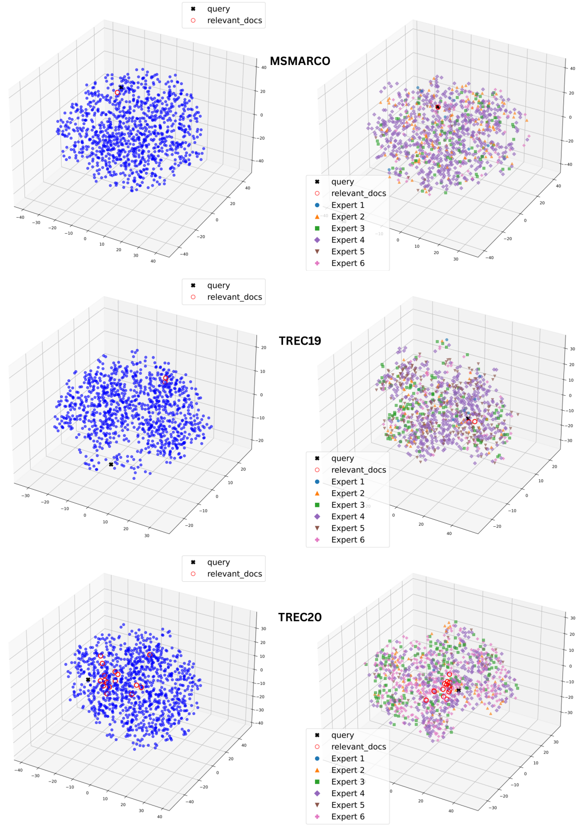

Figure 7: Dense vector space 3D t-SNE visualizations of query and document embeddings from the original DRM (left) and DenseC3 (right) on the MSMARCO, TREC19 & 20 benchmarks. Queries are shown as black crosses, their relevant documents as red circles, and the remaining points represent the top 1000 retrieved documents. Color figure online.

DenseC3_w may already be effective in such settings. For this evaluation, we annotated all MSMARCO training set documents using our frozen CLS for Cognitive Complexity. MSMARCO is widely adopted for zero-shot retrieval evaluation (Kamalloo et al., 2024). We then fine-tuned our model on the annotated MSMARCO training set and evaluated it in a zero-shot setting on the BEIR test sets of NQ and HotpotQA. For this preliminary analysis, we used Contriever and ColBERT as the underlying DRMs, as they are two of the best-performing retrievers in the IR literature.

D Query and Document Alignment

Figures 7 & 8 display 3D t-SNE visualizations of queries and their top 1000 retrieved documents as embedded in the Dense Vector Space (DVS) by TinyBERT[6]. The illustrations compare embeddings derived from the original DRM (Figures 7 & 8 – left) with those derived from DenseC3 (Figures 7 & 8 – right) across all seven benchmarks. Our analysis reveals that queries and their relevant documents undergo substantial positional shifts within the DVS when embedded with DenseC3 (Cognitive Complexity-aware embeddings) compared to their initial position given by the original DRM. DenseC3 enriches the query and document representations and better aligns them in the DVS for similarity estimation, leading to the retrieval effectiveness improvements noted in Table 2.

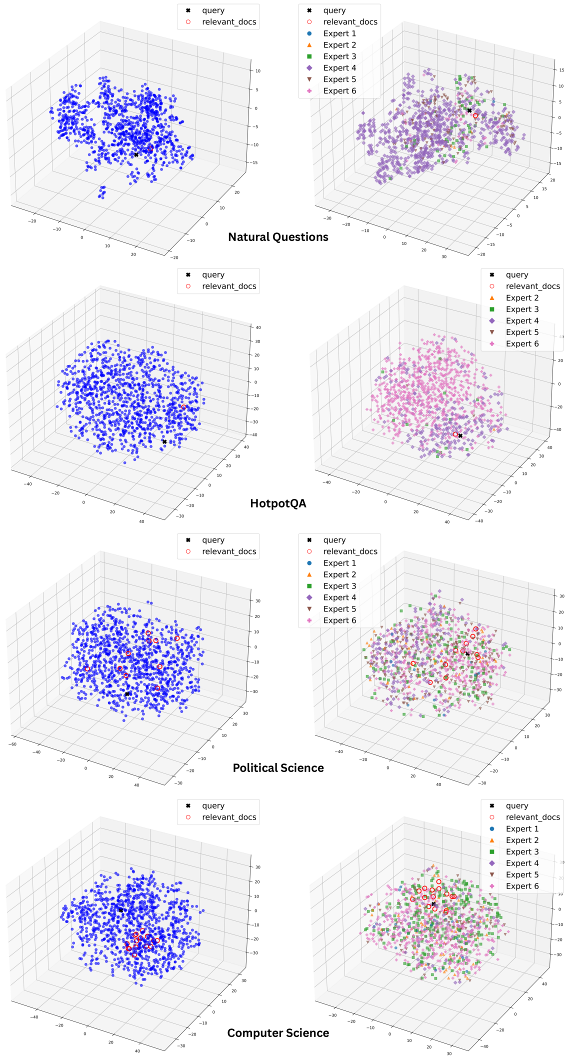

Figure 8: Dense vector space 3D t-SNE visualizations of query and document embeddings from the original DRM (left) and DenseC3 (right) on the NQ, HotpotQA, PS, and CS benchmarks. Queries are shown as black crosses, their relevant documents as red circles, and the remaining points represent the top 1000 retrieved documents. Color figure online.

[6] Additional visualizations derived from all four employed DRMs are presented in the provided code repository.