RARe: Retrieval-Augmented Retrieval With In-Context Examples

Atula Tejaswi[♠], **Yoonsang Lee**[♡], **Sujay Sanghavi**[♠*], **Eunsol Choi**[◇*]
[♠]The University of Texas at Austin [♡]Princeton University [◇]New York University
atutej@utexas.edu

Abstract

While in-context learning is well-studied with decoder-only language models (LLMs), its utility for encoder-only models remains underexplored. We study in-context learning for encoder-only models for text retrieval tasks. Can incorporating in-context examples (query-document pairs) to the target query enhance retriever performance? Our approach, RARe, finetunes a pre-trained model with in-context examples whose query is semantically similar to the target query. This approach achieves performance gains of up to +2.72% nDCG across open-domain retrieval datasets (BeIR, RAR-b) compared to using the target query only as an input. In particular, we find RARe exhibits stronger out-of-domain generalization compared to models using queries without in-context examples, similar to what is seen for in-context learning in LLMs. We further provide analysis on the design choices of in-context example augmentation for retrievers and lay the foundation for future work.

1 Introduction

In-context learning (ICL) (Brown et al., 2020) has emerged as a powerful paradigm enabling diverse applications without parameter updates in large language models (LLMs). While in-context learning has been extensively studied for the decoder-only models (Xu et al., 2023; Min et al., 2022a; Dong et al., 2024), its application to encoder-only, embedding models have been limited. Recently, Li et al. (2024) proposed to incorporate randomly sampled few-shot examples into the query side to enhance the query embedding for embedding models. They reported strong performances, achieving state-of-the-art performances in a popular retrieval benchmark (Muennighoff et al., 2023). In this paper, we further study how in-context examples enhance the performance in retriever models, exploring its design choices and generalization capability with controlled studies.

Compared to decoder models which generates new tokens from the input as a continuation, how in-context examples would impact the output vector from an embedding model is less obvious. Unlike in decoder-only LLMs where in-context examples expand model capacity at generation time, in-context examples in encoder-only models may primarily provide task-relevant information rather than increasing model capacity. We study injecting *semantically similar* in-context examples to build a dense retriever model (Karpukhin et al., 2020) which embeds queries and documents into a shared representational space for efficient search over a large corpus. State-of-the-art retriever models started to leverage decoder-only models as a backbone (Wang et al., 2024b; BehnamGhader et al., 2024; Muennighoff et al., 2024; Meng et al., 2024; Lee et al., 2024a), further motivating our study of applying in-context examples.

We begin by naively prepending in-context examples to the target query and provide it to existing retriever models (BehnamGhader et al., 2024; Wang et al., 2024b; Meng et al., 2024). Unlike in decoder-only models, zero-shot modification leads to significant performance drop. We propose to construct retrieval models that can effectively leverage in-context examples, which we name as RARe: Retrieval Augmented Retrieval with In-Context

[*] Equal advising.

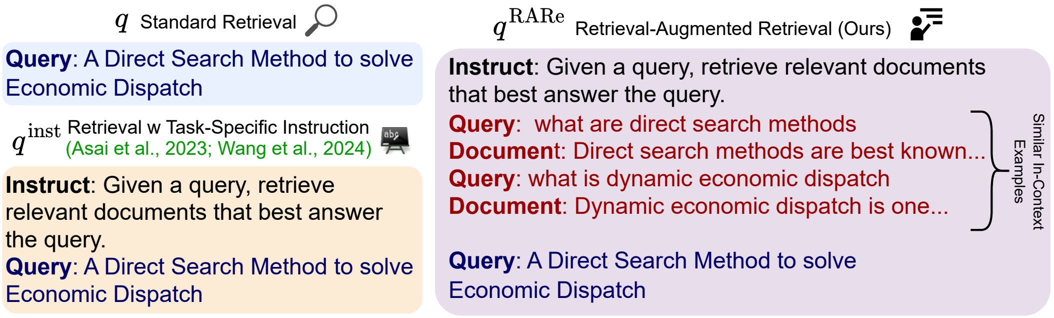
q Standard Retrieval
Query: A Direct Search Method to solve Economic Dispatch
q^{inst} Retrieval w Task-Specific Instruction (Asai et al., 2023; Wang et al., 2024)
Instruct: Given a query, retrieve relevant documents that best answer the query.
Query: A Direct Search Method to solve Economic Dispatch
q^{RARe} Retrieval-Augmented Retrieval (Ours)
Instruct: Given a query, retrieve relevant documents that best answer the query.
Query: what are direct search methods
Document: Direct search methods are best known...
Query: what is dynamic economic dispatch
Document: Dynamic economic dispatch is one...
Similar In-Context Examples
Query: A Direct Search Method to solve Economic Dispatch

Figure 1: Overview. Prior work augments a task-specific instruction to a given query as input to the Retriever. In RARe, we further leverage a set of in-context exemplars that contain pairs of queries and relevant documents. These in-context examples are augmented with the original query as input to the retriever along with the instruction.

Examples. Our approach modifies the query format of retrieval systems by providing in-context examples whose query is semantically similar to the target query. Then, we apply standard continued fine-tuning with contrastive loss. We conduct a comprehensive evaluation of new query format across various experimental settings, initializing from both decoder-only checkpoints and pre-trained retriever model checkpoints. We demonstrate that RARe outperforms baseline models across multiple tasks, achieving improvements of up to +1.41% nDCG@10 compared to methods that do not use in-context examples on standard retrieval benchmarks (Thakur et al., 2021), and showing even larger gains (+2.72% nDCG@10) on reasoning-oriented retrieval tasks (Xiao et al., 2024).

Our contributions can be summarized as follows:

- We introduce RARe, an approach that adapts pre-trained checkpoints to use semantically similar in-context examples for retrieval.
- We demonstrate that this recipe can improve the performance of various base architectures, including decoder-only models and existing retriever models.
- We provide detailed analyses on how the quality, quantity, and selection of in-context examples affect performance, contextualizing the sources of performance gains in embedding-based ICL.

All our code and model checkpoints are publicly released.[1]

2 Method

Standard Retrieval Setting We consider a dense retriever (Karpukhin et al., 2020), where input queries q and documents d are encoded with an embedder $E(\cdot)$ into a fixed-dimensional embedding. The embedder $E(\cdot)$ is trained on a training set $\mathcal{D}$ which consists of multiple retrieval tasks $\{\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_T\}$, where each task contains training examples of the form (q, d^+, d^-) (Wang et al., 2024b; BehnamGhader et al., 2024). Here, q is the input query, d^+ is a positive (relevant) document, and d^- is a hard-negative (irrelevant) document, which allows for a contrastive-loss based training.

The evaluation task $\mathcal{D}_{\text{test}}$ consists of a corpus of documents C, as well as test pairs (q, R^+), where $R^+ = \{d_1^+, d_2^+, ..., d_m^+\} \subset C$ is a set of relevant document(s) for the query (Thakur et al., 2021). The aim is to retrieve these relevant documents R^+ from the corpus C using the embedder $E(\cdot)$. Specifically, an index C_e of the corpus with document embeddings $E(d), \forall d \in C$ is created. Then, the embedding $E(q)$ of a test query q is used to retrieve the documents d whose embedding $E(d)$ is closest to $E(q)$, typically with the cosine (cos) similarity function.

[1] https://github.com/atutej/RARe

Algorithm 1: RARe - Training

```
Input: Training set $\mathcal{D}$, embedder $E(\cdot)$, BM25, the number of in-context examples $k$,
       mini-batch size $B$.
for each training iteration do
    Sample mini-batch $\mathcal{B}$ of size $B$ from $\mathcal{D}$
    for $(t_i, q, d^+, d^-) \in \mathcal{B}$ do
        In-Context Example Retrieval:
        $\{q_1^{\text{RARe}}, q_2^{\text{RARe}}, \ldots, q_k^{\text{RARe}}\} \leftarrow$ Retrieve nearest queries of $q$ from $\mathcal{D}$ using BM25
        $\{d_1^{\text{RARe+}}, d_2^{\text{RARe+}}, \ldots, d_k^{\text{RARe+}}\} \leftarrow \{d^+ : (q', d^+) \in \mathcal{D}, q' \in \{q_1^{\text{RARe}}, \ldots, q_k^{\text{RARe}}\}\}$
        $\mathcal{D}_i^{\text{RARe}} \leftarrow \{(q_1^{\text{RARe}}, d_1^{\text{RARe+}}), \ldots, (q_k^{\text{RARe}}, d_k^{\text{RARe+}})\}$
        Query Augmentation:
        $q^{\text{RARe}}$ = Instruct: $\{t_i\}$; Query: $\{q_1^{\text{RARe}}\}$; Document: $\{d_1^{\text{RARe+}}\} \cdots$; Query: $\{q\}$
    Training with Contrastive Loss:
    Compute the mini-batch contrastive loss $\mathcal{L}_{\text{in-context}}$ as described in Equation 5.
    Update $E(\cdot)$ by minimizing $\mathcal{L}_{\text{in-context}}$.
Output: Trained embedder $E(\cdot)$
```

Retrieval with Instruction Setting Current architectures (Asai et al., 2023; BehnamGhader et al., 2024) prepend task-specific instruction t_i, $i \in [1, 2, \cdots, T]$ to the query to contextualize the task:

$$q^{\text{inst}} = \text{Instruct: } \{t_i\}\text{; Query: } \{q\}, \quad q \in \mathcal{D}_i \tag{1}$$

Then, the embedder $E(\cdot)$ is trained with a standard contrastive loss (Izacard et al., 2022; Karpukhin et al., 2020), incorporating q^{inst}, and $d^+, d^- \in \mathcal{D}_i$, along in-batch negatives $n \in \mathbf{N}$, where $\mathbf{N}$ represents the set of in-batch negatives,

$$\boldsymbol{e}_{q^{\text{inst}}} = E(q^{\text{inst}}); \quad \boldsymbol{e}_{d^+} = E(d^+); \quad \boldsymbol{e}_{d^-} = E(d^-); \quad \boldsymbol{e}_n = E(n) \tag{2}$$

$$\mathcal{L} = -\log \frac{f(e_{q^{\text{inst}}}, e_{d^+})}{f(e_{q^{\text{inst}}}, e_{d^+}) + f(e_{q^{\text{inst}}}, e_{d^-}) + \sum_{n \in \mathbf{N}} f(e_{q^{\text{inst}}}, e_n)} \tag{3}$$

Where $f(x, y) = \exp(\cos(x, y))$. During evaluation on $\mathcal{D}_{\text{test}}$, each test query is augmented with task-specific instruction t_{test}.

BGE-ICL Setting Recently, Li et al. (2024) proposed BGE-ICL, a model that augments in-context examples to enhance the query representation of retrievers. Specifically, given a query q, an in-context example set $\mathcal{D}_i^{\text{ic}} = \{(q_1^{\text{ic}}, d_1^{\text{ic+}}), (q_2^{\text{ic}}, d_2^{\text{ic+}}), \cdots, (q_k^{\text{ic}}, d_k^{\text{ic+}})\}$ is constructed by *randomly sampling* from $\mathcal{D}_i \in \mathcal{D}$, which are then augmented into the original query q^{inst} to obtain $q^{\text{inst+ic}}$

$$q^{\text{inst+ic}} = \text{“Instruct: } \{t_i\}\text{; Query: } \{q_1^{\text{ic}}\}\text{; Document: } \{d_1^{\text{ic+}}\} \cdots\text{; Query: } \{q\}\text{”} \tag{4}$$

Then, $E(\cdot)$ is trained with the same loss as Equation 3, but with $q^{\text{inst+ic}}$ instead of q^{inst},

$$\mathcal{L}_{\text{in-context}} = -\log \frac{f(e_{q^{\text{inst+ic}}}, e_{d^+})}{f(e_{q^{\text{inst+ic}}}, e_{d^+}) + f(e_{q^{\text{inst+ic}}}, e_{d^-}) + \sum_{n \in \mathbf{N}} f(e_{q^{\text{inst+ic}}}, e_n)} \tag{5}$$

Our RARe Setting We propose to enhance the query representation by incorporating *semantically-similar* in-context examples. This provides additional, highly relevant query-specific guidance to the model.

Given a query q, we use BM25 (Robertson & Zaragoza, 2009), a sparse retriever that ranks documents based on keyword matching, and find k closest queries q_j from $\mathcal{D}_i \in \mathcal{D}$ to obtain in-context examples $\mathcal{D}_i^{\text{RARe}} = \{(q_1^{\text{RARe}}, d_1^{\text{RARe+}}), (q_2^{\text{RARe}}, d_2^{\text{RARe+}}), \cdots, (q_k^{\text{RARe}}, d_k^{\text{RARe+}})\}$. As shown in Figure 1, we augment these examples to the original query q^{inst} to obtain q^{RARe},

$$q^{\text{RARe}} = \text{“Instruct: }\{t_i\}\text{; Query: }\{q_1^{\text{RARe}}\}\text{; Document: }\{d_1^{\text{RARe+}}\}\cdots\text{; Query: }\{q\}\text{”} \tag{6}$$

We then train $E(\cdot)$ using Equation (5) to leverage these semantically similar examples. Algorithm 1 presents our training procedure in detail. At inference time, we similarly perform a search to find nearest in-context examples to form an augmented query. Algorithm 2 in the Appendix provides an overview of the inference procedure.

3 Experimental Setup

3.1 Fine-Tuning

Base Models We explore two training setups: fine-tuning decoder-only models for retrieval, and fine-tuning existing retriever models. For the first setup, we train the *Llama-3* family of models, following the training methodology outlined by Ma et al. (2023); Weller et al. (2024b). For the second setup, we use two high-performing publicly available embedding models that were trained with task-specific instructions: *LLM2Vec-Llama-3-8b-Supervised* (BehnamGhader et al., 2024) and *E5-Mistral-Instruct* (Wang et al., 2024b). We chose these two models because, unlike some other strong performers (Meng et al., 2024; de Souza P. Moreira et al., 2024), they were not trained on most of the datasets used in our downstream benchmarks. The *LLM2Vec-Llama-3-8b-Supervised* model is initially trained using an unsupervised text reconstruction objective and then fine-tuned with supervised contrastive learning on a public subset of the E5 dataset, which incorporates various supervised training datasets (Gao et al., 2021; Nguyen et al., 2016; Kwiatkowski et al., 2019). In contrast, *E5-Mistral-Instruct* undergoes further training on synthetic data that is not publicly available. These models are chosen to assess the impact of additional supervised training on an existing retriever model versus training a model for retrieval from scratch.

Training Data For fine-tuning existing retriever models, we follow prior work (BehnamGhader et al., 2024) and train on a publicly available portion of E5 dataset (Springer et al., 2024; Wang et al., 2024b), which contains MS-MARCO (Nguyen et al., 2016) NLI (Gao et al., 2021), ELI5 (Fan et al., 2019), FEVER (Thorne et al., 2018), HotpotQA (Yang et al., 2018), NQ (Kwiatkowski et al., 2019), SQuAD (Rajpurkar et al., 2016), Quora Duplication Questions (DataCanary et al., 2017), TriviaQA (Joshi et al., 2017). For fine-tuning models from LLM checkpoint, we use the MS-MARCO (Nguyen et al., 2016) passage ranking dataset and train without an instruction prefix, following Ma et al. (2023).

Constructing In-Context Examples During training, we provide each training example with five in-context examples from the dataset that it belongs to (k=5). Specifically, the set of examples $\mathcal{D}_i^{\text{RARe}}$ for each task is drawn from the training set $\mathcal{D}_i$, $q \notin \mathcal{D}_i^{\text{RARe}}$.

3.2 Evaluation

Datasets We evaluate on the widely used BeIR retrieval benchmark (Thakur et al., 2021). For ablative experiments, we follow prior work and focus on low-resource datasets (Wang et al., 2023) that potentially benefit more from few-shot examples. Since the BeIR benchmark contains a few datasets whose training sets are in the E5 dataset mixutre, we categorize them as in-domain and out-of-domain (i.e. datasets not seen during training). See Table 6 in the Appendix for a list of in-domain and out-of-domain datasets from BeIR. We also evaluate on a subset of the RAR-b (Xiao et al., 2024) benchmark, which requires complex reasoning for retrievers. Specifically, we evaluate on HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2020), ARC-C (Clark et al., 2018), TempReason-L1 (Tan et al., 2023), WinoGrande (Sakaguchi et al., 2021), α-NLI (Bhagavatula et al., 2020), SiQA (Sap et al., 2019), and Quail (Rogers

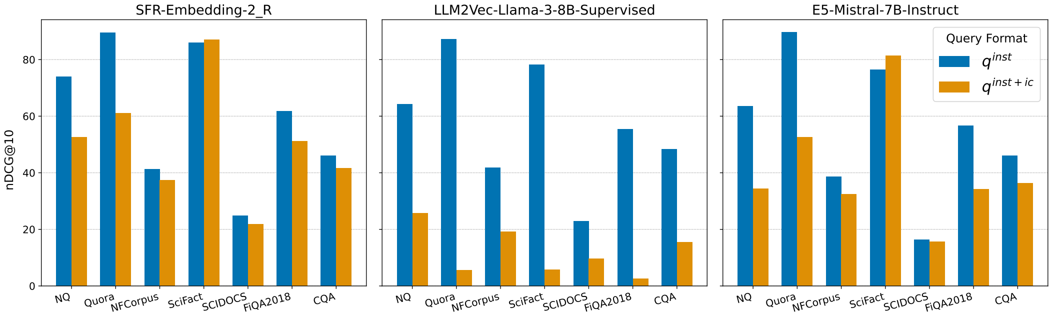

Figure 2: **Inference-only modification does not work.** We report performance before and after adding in-context examples to the query without updating model parameters. Embedding models are not able to leverage in-context examples out of the box, as opposed to decoder-only models.

et al., 2020). Unlike BeIR, some RAR-b queries are composed of sentences with (multiple) indicators (e.g., Start:, End:). Each dataset is associated with a task-specific instruction, following prior work (Muennighoff et al., 2023; Wang et al., 2024b; BehnamGhader et al., 2024). We provide additional preprocessing details in Appendix A.

Constructing In-Context Examples We construct $\mathcal{D}_{\text{test}}^{\text{RARe}}$ from the training/development set of each datasets. For datasets on BeIR that do not have either of these, we use a synthetically generated collection of document-query pairs (GenQ) from Thakur et al. (2021). For all experiments, we use k=5 in-context examples.

Metrics We use standard metrics for retrieval benchmarks. Following Thakur et al. (2021), we report nDCG@10, which measures the ranking quality of the top 10 retrieved documents, taking into account both the relevance and position of each retrieved document.

4 Results

We evaluate in-context example augmented queries in three settings. First, we evaluate the performance after inference-only modification, where we take existing pre-trained retrievers and simply provide in-context examples at inference time (Section 4). Second, we evaluate training retriever with in-context examples from an LLM (decoder-only) backbone (Section 4.1). Third, we compare training retriever models with in-context examples from a pre-trained retriever (Section 4.2).

Inference-only Modification Figure 2 illustrates the impact of incorporating in-context examples at inference time. Here, we simply modify the query format with retrieved in-context examples (i.e. q^{RARe}, Eq. 6) at inference time and compare its performance with the query format that does not have retrieved in-context examples (i.e. q^{inst}, Eq. 1). We evaluate the performance on three retriever models: *SFR-Embedding-2-R* (Meng et al., 2024), *LLM2Vec-Llama-3-8B-Supervised* (BehnamGhader et al., 2024), and *E5-Mistral-7B-Instruct* (Wang et al., 2024b). Unlike in autoregressive LLMs, these embedding models generally exhibit decreased performance when in-context examples are added, with *LLM2Vec-Llama-3-8B-Supervised* showing the largest drops in performance, except on one dataset (SciFact), where 2 out of 3 models show marginal gains over providing only instructions. Our experiments, which include adding more in-context examples and using nearest-neighbor examples, extend the findings of Muennighoff et al. (2024), where in-context examples led to decrease in performance on the GritLM models.

Table 1: **Training from decoder-only (LLM) checkpoint.** We initialize all models with *Llama-3.1-8b-Instruct*. Performance is measured by nDCG@10. RARe shows +2.72% and +0.87% nDCG@10 over Promptriever and ICL, respectively, on the reasoning oriented RAR-b benchmark. We provide a breakdown of In-Domain (ID), Out-of-Domain (OOD), and overall (Average) performance.

Method	**Training Data**	**ID**	**OOD**		**Average**
		BeIR	BeIR	RAR-b	
RepLLaMA	MS-MARCO	**43.67**	54.34	19.20	39.07
Promptriever	MS-MARCO + Synthetic	42.70	**56.10**	20.95	39.94
ICL	MS-MARCO	42.57	53.91	22.80	39.76
RARe	MS-MARCO	42.93	56.05	**23.67**	**40.88**

Table 2: **Training from retriever checkpoints.** Performance (nDCG@10) on BeIR (Thakur et al., 2021) and RAR-b (Xiao et al., 2024) benchmarks when fine-tuning retriever model on E5 dataset. We report a breakdown of performance on In-Domain (ID), Out-of-Domain (OOD, and overall (Average) performance.

Method	*LLM2Vec-Llama-3-8b-Supervised*				*E5-Mistral-Instruct*			
	ID	**OOD**		**Average**	**ID**	**OOD**		**Average**
	BeIR	BeIR	RAR-b		BeIR	BeIR	RAR-b	
Base	71.31	49.28	21.55	47.38	71.95	49.33	22.17	47.81
Instruct	70.46	47.79	**23.44**	47.23	72.91	48.98	24.12	48.67
ICL	**71.81**	47.69	21.94	47.14	72.03	49.46	24.69	48.72
RARe	71.67	**49.30**	23.10	**48.02**	**72.98**	**50.93**	**25.79**	**49.90**

4.1 Training from LLM Checkpoints

Next, we present the results of applying our approach when training from LLM checkpoint. This might preserve in-context learning capacity of the LLM, which can be lost during standard IR training, which compresses query and passage into a fixed dimensional vector. We initialize our models with *Llama-3.1-8b-Instruct* (Dubey et al., 2024) to enable comparison with prior work Ma et al. (2023); Weller et al. (2024b).

Comparison Systems We compare training with our in-context example augmented query with two baselines. The first baseline is vanilla query (Eq. 1), which was explored in RepLLaMA (Ma et al., 2023). The second baseline is Promptriever (Weller et al., 2024b) which augments query-specific instructions using a synthetically generated training set from MS-MARCO. The third baseline is ICL, which prepends random in-context examples to the query during training and inference, equivalent to that of Li et al. (2024). In all these systems, the task-specific instruction is a null string (Ma et al., 2023) as we train on a single task (MS-MARCO).

Results Table 1 presents the performance on downstream benchmarks when training from LLM checkpoints. Comparing within the same base LLM checkpoint, our apporach outperforms both baselines (RepLLaMA and Promptriever). Our performance is competitive to that of Promptriever (Weller et al., 2024b), without incorporating synthetic data during training. Specifically, RARe achieves +0.94% nDCG@10 over Promptriever, and +1.12% nDCG@10 over random ICL on average.

4.2 Training from Retriever Checkpoints

Lastly, we continue training retriever models – *LLM2Vec-Llama-3-8B-Supervised* (BehnamGhader et al., 2024), *E5-Mistral-Instruct* (Wang et al., 2024b) on a training

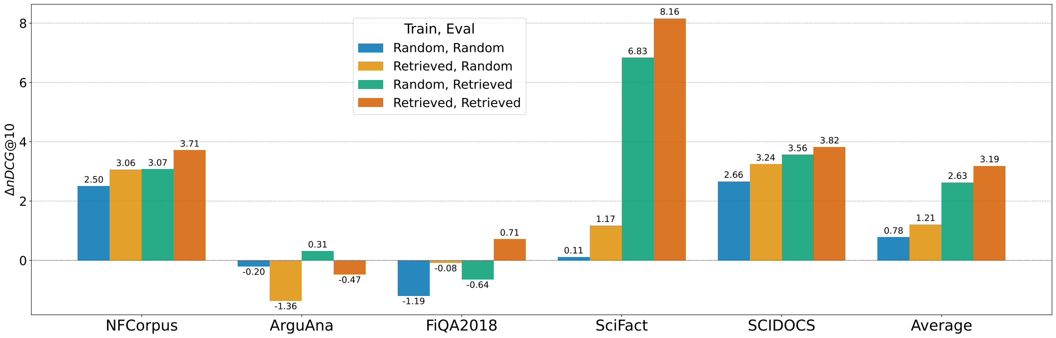

Figure 3: **Retrieved vs. Random In-context Examples.** Change in performance (ΔnDCG@10) on *E5-Mistral-Instruct* with RARe (q^{RARe}) from the baseline setting (q^{inst} both during training and evaluation time). Using retrieved examples during training and inference enhance model performance in most benchmark datasets.

set where queries are augmented with in-context examples. As these initial checkpoints have already been trained on the training dataset, the extent that retrievers adapt to new query format can be limited.

Comparison Systems We first report the initial retriever performance (**Base**) without any modification. Then, we compare continued fine-tuning with the task-specific instruction query format (Eq. 1) which only prepends the task specific instruction (**Instruct**, q^{inst}). Finally, we compare against **ICL** (Li et al., 2024), which augments random in-context examples to the query ($q^{\text{inst+ic}}$, Eq. 4) during training and evaluation.

Results Table 2 reports experimental results in this setting. Overall, both fine-tuning approaches provides gains over the base checkpoints. Comparing two settings, Instruct (q^{inst}) vs. RARe (q^{RARe}), our method achieves notable improvement with *E5-Mistral-Instruct* base model (up to +1.47% over random ICL on out-of-domain tasks, and +1.18% overall). Both RARe and ICL perform similar to Instruct (q^{inst}) setting when trained with the LLM2Vec base model. It is hard to attribute why experimental results varies based on the base retriever checkpoint, but we note the following differences between the two models. *LLM2Vec-Llama-3-8b-Supervised* is the only model in our experiments where further fine-tuning with only instructions led to a decrease in in-domain performance. *E5-Mistral-Instruct* employs causal attention with last token pooling, and trains on a proprietary synthetic dataset, *LLM2Vec-Llama-3-8b-Supervised* uses bidirectional attention with mean pooling, training only on the public portion.

The effectiveness of learning with in-context examples may depend on factors such as the underlying model architecture and data setting. Future work can further investigate the impact of base model characteristics, including architectural choices and other constraints.

5 Discussions and Analysis

5.1 Choice of In-context Examples

Retrieved (Similar) vs. Random In-Context Examples In Figure 3, we study the impact of retrieving the nearest neighbor query-document pairs as examples against randomly chosen examples during training and evaluation. We observe that using retrieved examples during both training and evaluation (Retrieved, Retrieved) consistently outperforms other configurations across most datasets. (Random, Retrieved) and (Retrieved, Random) generally outperform (Random, Random), suggesting retrieved examples are advantageous even when trained with randomly paired in-context examples. On ArguAna, we observe that (Retrieved, Random) performs the worst. There is a mismatch in the lengths of the queries

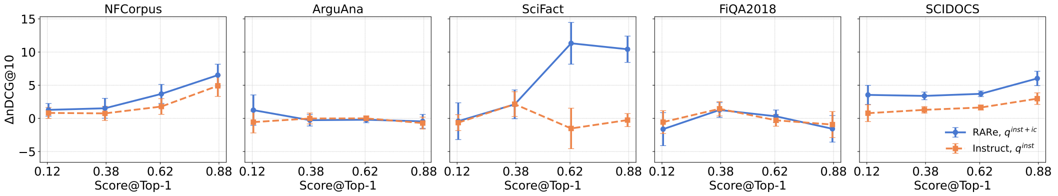

Figure 4: Change in performance (ΔnDCG@10) from the base model (*E5-Mistral-Instruct*) for varying similarity between the closest in-context example query and target query (Score@Top-1). For RARe, we use retrieved in-context examples q^{RARe} on the augmented in-context query format $q^{\text{inst+ic}}$.

Table 3: **Impact of the number of in-context examples (k) during training and evaluation.** All results are on *E5-Mistral-Instruct*. In general, performance increases when increasing the number of examples, and the optimal number of examples depends on the task.

k	**Arguana**	**CQADupStack**	**FiQA2018**	**NFCorpus**	**SciFact**	**Touche2020**	**Average**
Instruct (0)	61.19	44.82	**57.39**	40.99	77.28	29.35	51.84
1	60.47	46.76	56.07	40.67	81.47	**29.78**	52.54
3	**62.98**	47.12	57.08	40.77	83.71	27.12	53.13
5	60.87	48.46	57.31	**42.28**	84.79	28.70	53.74
10	58.85	**48.92**	57.03	42.24	**87.61**	28.29	**53.82**

used as in-context examples[2] (which are significantly shorter) versus the actual test queries in this dataset. This mismatch may introduce variability in performance, which has also been observed in decoder-only LLMs (Mishra et al., 2022). Overall, our findings align with prior work in in-context learning – that the incorporation of semantically similar examples is beneficial (Agrawal et al., 2022; Rubin et al., 2022). We observe similar overall trends on the other OOD datasets on the BeIR benchmark, reported in Figure 5 in the Appendix.

Does Having Semantically Relevant In-Context Example Help? For some test examples, augmented in-context examples are very relevant, and for others, much less so. In this section, we group the evaluation examples by the maximum similarity of in-context query and the test query measured by an off-the-shelf sentence embedding model (Score@Top-1).[3] and plot the performances for each group. Figure 4 presents the performance of our system (RARe) and baseline (Instruct). On NFCorpus and SciFact datasets, we observe that when the closest in-context example has a high similarity with the target query, RARe demonstrates over 10% gains compared to the base model. On the other hand, fine-tuning with with q^{inst} exhibits relatively lower performance gains with increasing similarity thresholds. In some datasets, such as ArguAna and FiQA2018, gains with increasing Score@Top-1 are less pronounced, but generally matches the performance of the base model. We observe similar overall trends on other BeIR OOD datasets (Figure 6 in the Appendix).

How Many In-Context Examples Are Sufficient? We analyze the performance of RARe when varying the number of in-context examples provided during training and inference. Table 3 shows that increasing the number of in-context examples generally enhances performance. On ArguAna, we observe that 0 examples are optimal, which is likely due to the mismatch in the lengths of the queries used as in-context examples versus the actual test queries. The impact of adding more in-context example is not uniformly positive across all datasets, suggesting that the optimal number of in-context examples may be dataset-dependent. We observe similar trends when we fix the number of in-context examples to

[2] https://huggingface.co/datasets/BeIR/arguana-generated-queries

[3] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

Table 4: **In-Context Format** Comparing variants of in-context example format on *E5-Mistral-Instruct*. Instruct refers to the baseline which does not use any in-context examples.

Method	ArguAna	CQA	FiQA2018	NFCorpus	SciFact	Touche2020	Average
Instruct	**61.19**	44.82	**57.39**	40.99	77.28	**29.35**	51.83
Queries-Only	58.88	46.66	54.44	41.42	78.84	28.09	51.39
Doc-Only	57.54	48.28	56.02	41.62	79.80	29.01	52.05
Shuffle-NC	60.17	45.78	54.25	41.17	80.70	29.18	51.88
Shuffle-C	58.97	47.97	55.98	41.78	80.51	28.97	52.36
Regular	60.87	**48.46**	57.31	**42.28**	**84.79**	28.70	**53.74**

Table 5: **Impact of adding negative documents in the in-context prompt.** All results are on *E5-Mistral-Instruct*. Negative documents (d^-) in the prompt do not enhance performance.

Training / Eval Setting	ArguAna	CQA	FiQA2018	NFCorpus	SciFact	Touche2020	Average
RARe-q^{RARe}	60.87	48.46	57.31	42.28	84.79	28.70	53.74
RARe-$q^{\text{RARe+neg}}$	61.19	48.09	56.89	41.58	82.37	30.51	53.44

five during training and vary the number of examples provided during inference, which are provided in Table 11 in the Appendix.

Ablating Content and Format of In-context Examples One can view in-context examples as a form of query expansion (Lv & Zhai, 2009; Wang et al., 2023), providing useful keywords to improve the performance. In Table 4, we analyze the impact of various formats of in-context examples. Each row represents a different model, fine-tuned with the format that they are evaluated on. Query-Only and Doc-Only contain only queries and documents of in-context examples, respectively. For Shuffle-Constrain (C), we randomly shuffle the mapping between q and d. On the other hand, for Shuffle-No Constrain (NC), we do not assume any structure, meaning that a query can be followed by a query as well as a document.

First, we observe that Query-Only shows a larger performance drop over Doc-Only, suggesting in-context documents might contain more useful contents than in-context queries. Second, we observe that shuffling the pairings (Shuffle-C) marginally hurts in-context learning in RARe, as opposed to Shuffle-NC. Our findings align with prior study in decoder-only models (Min et al., 2022b) which showed strict correspondence between q and d is not required for performance gains from in-context examples. We observe similar trends when keeping the training format fixed (Regular) and vary only the evaluation format (Table 14 and Table 13 in the Appendix).

Negative Documents in the Query So far, we have used (q, d^+) i.e (Query, Positive Document) pairs as the in-context prompt. Therefore, we study the impact of including negative documents. Specifically, the augmented query $q^{\text{RARe+neg}}$ includes examples of the form (q, d^+, d^-), where the documents are prefixed with the term "Positive Document: " and "Negative Document: " respectively. Table 5 presents the downstream performance comparison between RARe variants trained solely on positive examples and those trained with augmented negative documents. The results indicate no performance gains from including negative documents. In fact, training with negative examples led to a slight decrease in performance.

6 Related Work

In-context learning ICL (Brown et al., 2020) allows models to adapt to new tasks in a few-shot manner by conditioning on the input data and the context provided at inference time. ICL has been effectively applied to a wide range of tasks such as classification (Milios et al., 2023), translation (Zhu et al., 2024), mathematical reasoning (Wei et al., 2022; Zhou et al., 2022), and code generation (Li et al., 2023a). Recent advancements have enhanced the ICL capabilities of language models through additional training procedures (Huang et al., 2022; Gu et al., 2023; Shi et al., 2024). Min et al. (2022a) and Chen et al. (2022) perform meta-learning with in-context examples on a wide collection of tasks, with the goal of adapting to a new task at inference time through few-shot in-context examples. Other works have explored improving performance through more principled approaches to select in-context examples during inference (Zhang et al., 2022; Sorensen et al., 2022; Wang et al., 2024c; Qin et al., 2024; Lee et al., 2024b). A simple and popular approach is to retrieve examples that are most similar to the input (Liu et al., 2022; Rubin et al., 2022; Li et al., 2023c). Providing in-context examples to re-ranking models has been studied in prior work (Drozdov et al., 2023), but the potential of augmenting retrievers themselves by leveraging in-context examples remains unexplored. Muennighoff et al. (2024) explored providing an in-context example out-of-the-box, but showed an overall decrease in performance compared to zero-shot inference. The work most closely related to ours is by Li et al. (2024), who use a fixed, randomly selected set of in-context examples rather than retrieving query-specific ones. In contrast, our approach retrieves relevant examples tailored to each query and includes a comprehensive analysis of key design choices such as the quality, quantity, and format of in-context examples, as well as the choice of base model.

Retrieval Large language models pre-trained with autoregressive setups (Jiang et al., 2023; Dubey et al., 2024) have shown remarkable performance when adapted to retrieval tasks (Wang et al., 2024b; BehnamGhader et al., 2024), outperforming encoder-style retrievers (Izacard et al., 2022; Wang et al., 2024a). Despite these advancements, a challenge that remains is the ability to tailor retrieval systems to specific tasks or queries. To address this, a recent line of work explores incorporating instructions into retrieval by training models to use task-specific instructions along with the query (Su et al., 2023; Asai et al., 2023). Oh et al. (2024) and Weller et al. (2024a) further propose using instructions that are specific to each query. Another well-established technique in retrieval is query expansion (Jagerman et al., 2023; Li et al., 2023b; Chen et al., 2024), where the query is augmented with additional terms to enrich the embedding as a form of relevance feedback (Lv & Zhai, 2009). Recent efforts have focused on applying LLMs to expand the original query before retrieval (Wang et al., 2023; Shen et al., 2024; Dai et al., 2022). However, our focus is on the dense retrieval paradigm where a separate strong generative LLM is not available or used.

7 Conclusion

In this paper, we investigated the impact of in-context example augmentation for retrieval models, building on recent efforts to extend in-context learning for retrievers. We introduced RARe, a simple strategy that equips retrievers with the ability to leverage in-context examples by training with semantically similar in-context examples. Through detailed experiments and analyses, we demonstrated that RARe consistently improves performance across various architectures and downstream retrieval tasks, demonstrating the effectiveness of in-context learning for retriever models.

Acknowledgments

The work is partially supported by NSF grant IIS-2312948. This research has been supported by computing support on the Vista GPU Cluster through the Center for Generative AI (CGAI) and the Texas Advanced Computing Center (TACC) at UT Austin. We also thank Fangyuan Xu, Michael Zhang, Anuj Diwan, and other members of the UT NLP community for insightful feedback.

References

Sweta Agrawal, Chunting Zhou, Mike Lewis, Luke Zettlemoyer, and Marjan Ghazvininejad. In-context examples selection for machine translation, 2022. URL https://arxiv.org/

abs/2212.02437.

Akari Asai, Timo Schick, Patrick Lewis, Xilun Chen, Gautier Izacard, Sebastian Riedel, Hannaneh Hajishirzi, and Wen-tau Yih. Task-aware retrieval with instructions. In *Findings of the Association for Computational Linguistics: ACL 2023*, Toronto, Canada, July 2023. Association for Computational Linguistics. URL https://aclanthology.org/2023.findings-acl.225.

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. Llm2vec: Large language models are secretly powerful text encoders, 2024. URL https://arxiv.org/abs/2404.05961.

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Wen tau Yih, and Yejin Choi. Abductive commonsense reasoning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Byg1v1HKDB.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7432–7439, 2020.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.

Xinran Chen, Xuanang Chen, Ben He, Tengfei Wen, and Le Sun. Analyze, generate and refine: Query expansion with LLMs for zero-shot open-domain QA. In *Findings of the Association for Computational Linguistics ACL 2024*, Bangkok, Thailand and virtual meeting, August 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.findings-acl.708.

Yanda Chen, Ruiqi Zhong, Sheng Zha, George Karypis, and He He. Meta-learning via language model in-context tuning, 2022. URL https://arxiv.org/abs/2110.07814.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Zhuyun Dai, Vincent Y. Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith B. Hall, and Ming-Wei Chang. Promptagator: Few-shot dense retrieval from 8 examples, 2022. URL https://arxiv.org/abs/2209.11755.

DataCanary, hilfialkaff, Lili Jiang, Meg Risdal, Nikhil Dandekar, and tomtung. Quora question pairs, 2017. URL https://kaggle.com/competitions/quora-question-pairs.

Gabriel de Souza P. Moreira, Radek Osmulski, Mengyao Xu, Ronay Ak, Benedikt Schifferer, and Even Oldridge. Nv-retriever: Improving text embedding models with effective hard-negative mining, 2024. URL https://arxiv.org/abs/2407.15831.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. A survey on in-context learning, 2024. URL https://arxiv.org/abs/2301.00234.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The faiss library, 2024. URL https://arxiv.org/abs/2401.08281.

Andrew Drozdov, Honglei Zhuang, Zhuyun Dai, Zhen Qin, Razieh Rahimi, Xuanhui Wang, Dana Alon, Mohit Iyyer, Andrew McCallum, Donald Metzler, and Kai Hui. Parade: Passage ranking using demonstrations with large language models, 2023. URL https://arxiv.org/abs/2310.14408.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, July 2019. URL https://aclanthology.org/P19-1346.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2021. URL https://aclanthology.org/2021.emnlp-main.552.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Pre-training to learn in context. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Toronto, Canada, July 2023. Association for Computational Linguistics. URL https://aclanthology.org/2023.acl-long.267.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

Yukun Huang, Yanda Chen, Zhou Yu, and Kathleen McKeown. In-context learning distillation: Transferring few-shot learning ability of pre-trained language models, 2022. URL https://arxiv.org/abs/2212.10670.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL https://openreview.net/forum?id=jKN1pXi7b0.

Rolf Jagerman, Honglei Zhuang, Zhen Qin, Xuanhui Wang, and Michael Bendersky. Query expansion by prompting large language models, 2023. URL https://arxiv.org/abs/2305.03653.

Nour Jedidi, Yung-Sung Chuang, Leslie Shing, and James Glass. Zero-shot dense retrieval with embeddings from relevance feedback, 2024. URL https://arxiv.org/abs/2410.21242.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, July 2017. URL https://aclanthology.org/P17-1147.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2020. URL https://aclanthology.org/2020.emnlp-main.550.

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. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7, 2019. URL https://aclanthology.org/Q19-1026.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. Nv-embed: Improved techniques for training llms as generalist embedding models, 2024a. URL https://arxiv.org/abs/2405.17428.

Yoonsang Lee, Pranav Atreya, Xi Ye, and Eunsol Choi. Crafting in-context examples according to lms' parametric knowledge, 2024b. URL https://arxiv.org/abs/2311.09579.

Yibin Lei, Yu Cao, Tianyi Zhou, Tao Shen, and Andrew Yates. Corpus-steered query expansion with large language models, 2024. URL https://arxiv.org/abs/2402.18031.

Chaofan Li, MingHao Qin, Shitao Xiao, Jianlyu Chen, Kun Luo, Yingxia Shao, Defu Lian, and Zheng Liu. Making text embedders few-shot learners, 2024. URL https://arxiv.org/abs/2409.15700.

Jia Li, Ge Li, Chongyang Tao, Jia Li, Huangzhao Zhang, Fang Liu, and Zhi Jin. Large language model-aware in-context learning for code generation, 2023a. URL https://arxiv.org/abs/2310.09748.

Minghan Li, Honglei Zhuang, Kai Hui, Zhen Qin, Jimmy Lin, Rolf Jagerman, Xuanhui Wang, and Michael Bendersky. Generate, filter, and fuse: Query expansion via multi-step keyword generation for zero-shot neural rankers. *arXiv preprint arXiv:2311.09175*, 2023b.

Xiaonan Li, Kai Lv, Hang Yan, Tianyang Lin, Wei Zhu, Yuan Ni, Guotong Xie, Xiaoling Wang, and Xipeng Qiu. Unified demonstration retriever for in-context learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Toronto, Canada, July 2023c. Association for Computational Linguistics. URL https://aclanthology.org/2023.acl-long.256.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, Dublin, Ireland and Online, May 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.deelio-1.10.

Yuanhua Lv and ChengXiang Zhai. A comparative study of methods for estimating query language models with pseudo feedback. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pp. 1895–1898, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605585123. doi: 10.1145/1645953.1646259. URL https://doi.org/10.1145/1645953.1646259.

Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. Fine-tuning llama for multi-stage text retrieval, 2023. URL https://arxiv.org/abs/2310.08319.

Rui Meng, Ye Liu, Shafiq Rayhan Joty, Caiming Xiong, Yingbo Zhou, and Semih Yavuz. Sfr-embedding-mistral:enhance text retrieval with transfer learning. Salesforce AI Research Blog, 2024. URL https://blog.salesforceairesearch.com/sfr-embedded-mistral/.

Aristides Milios, Siva Reddy, and Dzmitry Bahdanau. In-context learning for text classification with many labels. In *Proceedings of the 1st GenBench Workshop on (Benchmarking) Generalisation in NLP*, Singapore, December 2023. Association for Computational Linguistics. URL https://aclanthology.org/2023.genbench-1.14.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. MetaICL: Learning to learn in context. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Seattle, United States, July 2022a. Association for Computational Linguistics. URL https://aclanthology.org/2022.naacl-main.201.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work?, 2022b. URL https://arxiv.org/abs/2202.12837.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, Yejin Choi, and Hannaneh Hajishirzi. Reframing instructional prompts to GPTk's language. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 589–612, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.50. URL https://aclanthology.org/2022.findings-acl.50.

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. MTEB: Massive text embedding benchmark. In Andreas Vlachos and Isabelle Augenstein (eds.), *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 2014–2037, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.eacl-main.148. URL https://aclanthology.org/2023.eacl-main.148.

Niklas Muennighoff, Hongjin Su, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. Generative representational instruction tuning, 2024. URL https://arxiv.org/abs/2402.09906.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. MS MARCO: A human generated machine reading comprehension dataset. *CoRR*, abs/1611.09268, 2016. URL http://arxiv.org/abs/1611.09268.

Hanseok Oh, Hyunji Lee, Seonghyeon Ye, Haebin Shin, Hansol Jang, Changwook Jun, and Minjoon Seo. Instructir: A benchmark for instruction following of information retrieval models, 2024. URL https://arxiv.org/abs/2402.14334.

Chengwei Qin, Aston Zhang, Chen Chen, Anirudh Dagar, and Wenming Ye. In-context learning with iterative demonstration selection, 2024. URL https://arxiv.org/abs/2310.09881.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, November 2016. URL https://aclanthology.org/D16-1264.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389, apr 2009. ISSN 1554-0669. URL https://doi.org/10.1561/1500000019.

Anna Rogers, Olga Kovaleva, Matthew Downey, and Anna Rumshisky. Getting closer to ai complete question answering: A set of prerequisite real tasks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 8722–8731, 2020.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Seattle, United States, July 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.naacl-main.191.

Jon Saad-Falcon, Daniel Y. Fu, Simran Arora, Neel Guha, and Christopher Ré. Benchmarking and building long-context retrieval models with loco and m2-bert, 2024. URL https://arxiv.org/abs/2402.07440.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social iqa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4463–4473, 2019.

Tao Shen, Guodong Long, Xiubo Geng, Chongyang Tao, Yibin Lei, Tianyi Zhou, Michael Blumenstein, and Daxin Jiang. Retrieval-augmented retrieval: Large language models are strong zero-shot retriever. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics ACL 2024*, pp. 15933–15946, Bangkok, Thailand and virtual meeting, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.943. URL https://aclanthology.org/2024.findings-acl.943.

Weijia Shi, Sewon Min, Maria Lomeli, Chunting Zhou, Margaret Li, Gergely Szilvasy, Rich James, Xi Victoria Lin, Noah A. Smith, Luke Zettlemoyer, Scott Yih, and Mike Lewis. In-context pretraining: Language modeling beyond document boundaries, 2024. URL https://arxiv.org/abs/2310.10638.

Taylor Sorensen, Joshua Robinson, Christopher Rytting, Alexander Shaw, Kyle Rogers, Alexia Delorey, Mahmoud Khalil, Nancy Fulda, and David Wingate. An information-theoretic approach to prompt engineering without ground truth labels. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Dublin, Ireland, May 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.acl-long.60.

Jacob Mitchell Springer, Suhas Kotha, Daniel Fried, Graham Neubig, and Aditi Raghunathan. Repetition improves language model embeddings, 2024. URL https://arxiv.org/abs/2402.15449.

Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. One embedder, any task: Instruction-finetuned text embeddings. In *Findings of the Association for Computational Linguistics: ACL 2023*, Toronto, Canada, July 2023. Association for Computational Linguistics. URL https://aclanthology.org/2023.findings-acl.71.

Qingyu Tan, Hwee Tou Ng, and Lidong Bing. Towards benchmarking and improving the temporal reasoning capability of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14820–14835, 2023.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL https://openreview.net/forum?id=wCu6T5xFjeJ.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, June 2018. URL https://aclanthology.org/N18-1074.

Liang Wang, Nan Yang, and Furu Wei. Query2doc: Query expansion with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Singapore, December 2023. Association for Computational Linguistics. URL https://aclanthology.org/2023.emnlp-main.585.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training, 2024a. URL https://arxiv.org/abs/2212.03533.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Improving text embeddings with large language models, 2024b. URL https://arxiv.org/abs/2401.00368.

Xinyi Wang, Wanrong Zhu, Michael Saxon, Mark Steyvers, and William Yang Wang. Large language models are latent variable models: Explaining and finding good demonstrations for in-context learning, 2024c. URL https://arxiv.org/abs/2301.11916.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Orion Weller, Benjamin Chang, Sean MacAvaney, Kyle Lo, Arman Cohan, Benjamin Van Durme, Dawn Lawrie, and Luca Soldaini. Followir: Evaluating and teaching information retrieval models to follow instructions, 2024a. URL https://arxiv.org/abs/2403.15246.

Orion Weller, Benjamin Van Durme, Dawn Lawrie, Ashwin Paranjape, Yuhao Zhang, and Jack Hessel. Promptriever: Instruction-trained retrievers can be prompted like language models, 2024b. URL https://arxiv.org/abs/2409.11136.

Chenghao Xiao, G Thomas Hudson, and Noura Al Moubayed. Rar-b: Reasoning as retrieval benchmark, 2024. URL https://arxiv.org/abs/2404.06347.

Benfeng Xu, Quan Wang, Zhendong Mao, Yajuan Lyu, Qiaoqiao She, and Yongdong Zhang. *k*nn prompting: Beyond-context learning with calibration-free nearest neighbor inference, 2023. URL https://arxiv.org/abs/2303.13824.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, 2018. Association for Computational Linguistics. URL https://aclanthology.org/D18-1259.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472.

Hanqi Zhang, Chong Chen, Lang Mei, Qi Liu, and Jiaxin Mao. Mamba retriever: Utilizing mamba for effective and efficient dense retrieval, 2024. URL https://arxiv.org/abs/2408.08066.

Yiming Zhang, Shi Feng, and Chenhao Tan. Active example selection for in-context learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.emnlp-main.622.

Hattie Zhou, Azade Nova, Hugo Larochelle, Aaron Courville, Behnam Neyshabur, and Hanie Sedghi. Teaching algorithmic reasoning via in-context learning, 2022. URL https://arxiv.org/abs/2211.09066.

Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. Multilingual machine translation with large language models: Empirical results and analysis. In *Findings of the Association for Computational Linguistics: NAACL 2024*, Mexico City, Mexico, June 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.findings-naacl.176.

Appendix

The appendix is organized as follows: In Appendix A, we present details on additional data preprocessing and other training details. In Appendix B, we present additional results and experiments.

A Experimental Details

A.1 Training Details

Hyperparameters For fine-tuning *Llama-3-8B*, we follow the setting outlined in Ma et al. (2023). We train on 4 H100 GPUs with per-device batch size 8 and gradient accumulation steps 4. We apply LoRA (Hu et al., 2021) with r=32, temperature of 0.01, learning rate 1e-4 with 100 warmup steps. We use a sequence length of 512 for documents and 1024 for queries as in-context augmented queries are longer. For RARe we use a mixture of 70% examples with in-context examples and 30% without Table 15.

When fine-tuning existing retriever models (*E5-Mistral-Instruct*, *LLM2Vec-Llama-3-8B-Supervised*), we follow a setting similar to BehnamGhader et al. (2024). We train on 8 H100 GPUs with a largest possible per-device batch size of 32 along with 2 gradient accumulation steps. We consider a random subset of 100K examples from the public E5 dataset mixture (Springer et al., 2024; Wang et al., 2024b). We use a learning rate of 2e-4, maximum sequence length 1024, warmup ratio 0.1 for 1 epoch. For *E5-Mistral-Instruct*, we apply LoRA (Hu et al., 2021) r=16, and r=4 for *LLM2Vec-Llama-3-8B-Supervised* since a higher rank was leading to severe overfitting on the instruction baseline.

A.2 Data Processing

RAR-b Since RAR-b benchmark provides only test split, we parse the original training data for each dataset to use as in-context examples. We exclude datasets without any training splits and 2 datasets that were a mixture of multiple tasks or datasets, thereby being difficult to parse. This results in 8 datasets to evaluate on. We preprocess the training split to match the format of RAR-b test split, without excluding any instances. An exception is made for α-NLI, where there were multiple identical instances. Therefore, we removed such duplicates, resulting in 72,046 in-context candidates. Furthermore, some RAR-b queries are composed of sentences with (multiple) indicators (e.g., Start:, End:). To address this, we make a minor modification in formatting, enclosing the queries in brackets. The final query representation is:

$$q^{\text{RARe}} = \text{“Instruct: }\{t\}\text{; Query: }[\{q_1^{\text{RARe}}\}]\text{; Document: }\{d_1^{\text{RARe+}}\}\cdots\text{;}\quad\text{Query: }[\{q\}]\text{”} \tag{7}$$

Inference Algorithm Algorithm 2 provides a detailed outline of inference with RARe.

Promptriever Promptriever (Weller et al., 2024b) employs 10 different prompts and reports the highest score for each dataset. We apply the prompt that works the best (outperforms 5/15 datasets), which is as follows: `A document that meets these criteria is considered relevant, while a document that does not meet these criteria is considered non-relevant.`

Synthetic In-Context Examples for BeIR For QuoraRetrieval, MSMARCO, DBPedia, FiQA-2018, NFCorpus, and SciFact, we utilize the original training splits without incorporating synthetically generated examples. For the remaining datasets—ArguAna, ClimateFEVER, FEVER, HotpotQA, Touche2020, TREC-COVID, CQADupStack, SCIDOCS, and NQ—we employ synthetic examples provided by the BEIR benchmark (e.g., ArguAna: `https://huggingface.co/datasets/BeIR/arguana-generated-queries`). For further details, please refer to our codebase at `https://github.com/atutej/RARe`.

Algorithm 2: RARe - Inference

```
Input: A list of test queries $D^{test}$, Corpus $C$, embedder $E(\cdot)$, the number of in-context
       examples $k$, Training dataset $\mathcal{D}^{\mathcal{T}}$, task instruction $t$.
$C_e \leftarrow$ Construct document index as $E(d), \forall d \in C$.
for $i \in [0, len(D^{test})]$ do
    $q = D^{\text{test}}[i]$
    In-Context Example Retrieval:
    $\{q_1^{\text{RARe}}, q_2^{\text{RARe}}, \ldots, q_k^{\text{RARe}}\} \leftarrow$ Retrieve nearest neighbor queries of $q$ from $\mathcal{D}^{\mathcal{T}}$ using BM25
    $\{d_1^{\text{RARe+}}, d_2^{\text{RARe+}}, \ldots, d_k^{\text{RARe+}}\} \leftarrow \{d^+ : (q', d^+) \in \mathcal{D}, q' \in \{q_1^{\text{RARe}}, \ldots, q_k^{\text{RARe}}\}\}$
    $\mathcal{D}^{\text{RARe}} \leftarrow \{(q_1^{\text{RARe}}, d_1^{\text{RARe+}}), \ldots, (q_k^{\text{RARe}}, d_k^{\text{RARe+}})\}$
    Query Augmentation / Encoding:
    $q^{\text{RARe}}$ = Instruct: $\{t\}$; Query: $\{q_1^{\text{RARe}}\}$; Document: $\{d_1^{\text{RARe+}}\} \cdots$; Query: $\{q\}$
    $\boldsymbol{e}_q \leftarrow E(q_{\text{test}}^{\text{RARe}})$
    Prediction:
    $d = \text{argmax}_{d \in C} \exp(\cos(\boldsymbol{e}_q, \boldsymbol{e}_d))$
    $D_{\text{pred}}$.append($d$)
Output: Predictions $D_{\text{pred}}$.
```

B Additional Experiments

Table 6: Performance (nDCG@10) on BeIR (Thakur et al., 2021) when fine-tuning retriever model on E5 dataset. We report a breakdown of performance on In-Domain (ID) and Out-of-Domain (OOD) tasks on BeIR. * indicates statistical significance over Instruct ($p < 0.05$) using the paired bootstrap test. For the Average score, we compute the overall p-value using Fisher's method.

		LLM2Vec-Llama-3-8b-Supervised				*E5-Mistral-Instruct*			
Category	**Dataset**	Base	Instruct	RARe		Base	Instruct	RARe	
	Eval Format	q^{inst}	q^{inst}	q^{inst}	q^{RARe}	q^{inst}	q^{inst}	q^{inst}	q^{RARe}
ID	FEVER	**90.20**	88.12	88.43	86.62	87.84	**91.50**	90.18	90.48
	HotpotQA	71.76	72.50	73.83	**79.09***	75.72	73.91	72.18	**75.95***
	NQ	64.21	63.63	65.00	**66.13***	63.53	67.44	**68.15**	67.66
	QuoraRetrieval	87.16	87.85	**87.88**	87.63	89.61	**89.82**	89.59	88.95
	MSMARCO	**43.24**	40.19	40.77	38.88	43.06	41.89	41.88	41.88
OOD	ArguAna	**62.78**	60.51	59.54	57.05	61.65	61.19	**62.90**	60.87
	ClimateFEVER	34.27	34.49	34.67	**34.73***	38.35	**39.03**	38.99	37.50
	CQADupStack	48.25	49.76	49.10	**49.93**	42.97	44.82	45.57	**48.46***
	DBPedia	48.34	48.61	48.41	**49.09***	48.89	48.92	49.24	**49.65***
	FiQA2018	**55.33**	52.99	54.26	52.82	56.81	**57.39**	56.33	57.31
	NFCorpus	41.83	**41.92**	41.61	41.84	38.58	40.99	41.19	**42.28***
	SCIDOCS	22.96	**23.97**	22.92	23.35	16.32	17.94	18.71	**20.19***
	SciFact	78.22	76.89	77.70	**81.77***	76.42	77.28	77.11	**84.79***
	Touche2020	20.50	22.11	**22.71**	19.54	26.27	**29.35**	27.56	28.7
	TRECCOVID	80.34	68.37	78.55	**82.78***	**87.03**	72.89	77.03	79.58
	Average	56.63	55.35	56.36	**56.76***	56.87	56.96	57.11	**58.28***

Table 7: Performance on reasoning-focused IR benchmark RAR-b (Xiao et al., 2024) when fine-tuning existing retriever models. * indicates statistical significance over Instruct ($p < 0.05$) using the paired bootstrap test. For the Average score, we compute the overall p-value using Fisher's method.

	LLM2Vec-Llama-3-8b-Supervised				*E5-Mistral-Instruct*			
Dataset	Base	Instruct	RARe		Base	Instruct	RARe	
Eval Format	q^{inst}	q^{inst}	q^{inst}	q^{RARe}	q^{inst}	q^{inst}	q^{inst}	q^{RARe}
ARC-C	**18.81**	18.77	18.28	17.02	19.00	20.37	22.72	**26.44***
α-NLI	26.59	**27.29**	25.25	23.66	**26.04**	25.70	24.19	23.23
HellaSwag	**34.32**	34.19	34.19	33.29	35.38	35.99	35.07	**36.29***
PIQA	33.57	37.07	38.12	**39.72***	39.80	39.35	37.22	**41.35***
Quail	**6.83**	6.06	5.57	4.25	8.40	10.94	**15.34**	14.69
SiQA	**6.99**	5.34	4.39	4.55	**5.66**	5.45	5.75	**6.15**
TempReason-L1	5.24	5.89	5.55	**7.87***	3.60	**4.71**	4.55	4.67
WinoGrande	40.02	52.88	48.47	**54.44***	39.48	50.41	44.26	**53.50***
Average	21.55	**23.44**	22.48	23.10	22.17	24.12	23.64	**25.79***

Table 8: Performance (nDCG@10) on BeIR when training decoder-only models. * indicates statistical significance over RepLLaMA ($p < 0.05$) using the paired bootstrap test. For the Average score, we compute the overall p-value using Fisher's method.

Dataset	*Llama2*	*Llama3*		*Llama-3.1-Instruct*			
	RepLLaMA	RepLLaMA	RARe	RepLLaMA	Promptreiver	RARe	
Eval Format	q^{inst}	q^{inst}	q^{RARe}	q^{inst}	q^{inst}	q^{inst}	q^{RARe}
ArguAna	48.60	52.83	49.48	51.38	58.90	**54.77**	52.83
ClimateFEVER	29.30	32.52	32.12	33.13	29.80	**35.91**	34.24*
CQADupStack	37.91	42.59	42.96	41.58	42.18	42.55	**43.31***
DBPedia	44.80	45.62	45.79	44.73	**46.00**	45.87	45.95*
FEVER	82.90	81.79	83.66	79.22	**85.50**	80.05	81.84*
FiQA2018	45.00	44.31	47.13	44.50	**47.20**	44.36	46.20*
HotpotQA	68.80	72.24	72.72	70.90	71.70	70.55	**74.01***
MSMARCO	42.00	43.56	44.77	**43.67**	42.70	41.65	42.93*
NFCorpus	36.00	37.73	39.34	38.77	38.50	38.16	**39.74***
NQ	63.00	62.70	65.96	61.09	63.80	60.92	**65.20***
Quora	86.00	88.34	87.65	86.84	87.30	**87.95**	87.65*
SCIDOCS	16.10	19.66	19.45	19.26	**20.80**	20.02	19.52
SciFact	75.30	75.02	77.20	75.38	**77.50**	74.59	76.54
TRECCOVID	83.90	83.15	85.76	83.15	84.50	77.52	**85.30***
Touche2020	34.10	27.84	32.89	30.77	31.70	25.47	**32.38**
Average	52.91	53.99	55.13	53.62	**55.21**	53.36	55.18*

B.1 Performance on BeIR and RAR-b

Table 6 and Table 7 provide detailed numbers on each dataset from BeIR and RAR-b respectively when training from retriever checkpoints. Table 8 and Table 9 provide detailed numbers on each dataset from BeIR and RAR-b respectively when training from decoder-only LLMs.

B.2 Efficiency Evaluation

In Table 10, we present a breakdown of the latency of each stage of the retrieval pipeline for both baseline (q^{inst}) and in-context ($q^{\text{inst+ic}}$) settings. We measure the total time required to obtain nearest-neighbour in-context examples (**NN**) from BM25, compute query embeddings (**Query**), and perform search with FAISS (Douze et al., 2024) with encoded query embeddings on the pre-computed document index (**Search**). We observe that the largest

Table 9: Performance (nDCG@10) on datasets from RAR-b when training decoder-only models. * indicates statistical significance over Promptriever ($p < 0.05$) using the paired bootstrap test. For the Average score, we compute the overall p-value using Fisher's method.

	Llama2	*Llama3*		*Llama-3.1-Instruct*			
Dataset	RepLLaMA	RepLLaMA	RARe	RepLLaMA	Promptreiver	RARe	
Eval Format	q^{inst}	q^{inst}	q^{RARe}	q^{inst}	q^{inst}	q^{inst}	q^{RARe}
ARC-C	11.79	11.65	13.48	11.68	14.63	13.24	**15.02**
α-NLI	25.40	24.35	30.38	24.96	24.70	27.34	**31.58***
HellaSwag	30.83	31.47	30.27	31.03	**32.57**	31.42	28.81
PIQA	31.56	32.84	34.12	33.42	34.80	34.23	**35.59***
Quail	6.40	6.21	5.98	5.71	**7.80**	6.92	6.91
SiQA	2.82	2.61	**3.87**	2.75	3.53	2.18	3.14
TempReason-L1	1.49	1.75	3.61	2.05	4.32	4.84	**6.59***
WinoGrande	51.58	37.11	57.01	42.01	45.25	44.72	**61.69***
Average	20.23	18.50	22.34	19.20	20.95	20.61	**23.67***

Table 10: Latency breakdown (in milliseconds per query) of each stage in the retrieval pipeline for q^{inst} and q^{RARe} evaluation settings. **# Corpus** denote the number of documents and **Avg Q len.** denote the average number of query tokens split by whitespace. Table 10 in the Appendix provides numbers on additional datasets.

Dataset	**# Corpus**	**Eval Setting**	**Avg Q len.**	**NN**	**Query**	**Search**	**Total**	**Inc.**
NFCorpus	3633	q^{inst}	3.3	0	10.22	1.67	11.89	-
		q^{RARe}	866.0	0.62	473.65	1.76	476.04	40.04×
FiQA2018	57638	q^{inst}	10.9	0	10.68	12.22	22.90	-
		q^{RARe}	1016.6	0.69	429.97	13.63	444.29	19.40×
TRECCOVID	171332	q^{inst}	10.6	0	36.60	81.60	118.20	-
		q^{RARe}	722.54	6.20	435.60	86.00	527.80	4.47×
Touche2020	382545	q^{inst}	6.6	0	28.98	189.59	218.57	-
		q^{RARe}	1287.8	4.08	822.86	214.29	1041.22	4.76×
Quora	522931	q^{inst}	9.5	0	11.39	98.64	110.04	-
		q^{RARe}	129.5	0.32	53.03	98.26	151.61	1.38×
DBPedia	4635922	q^{inst}	5.5	0	92.33	1470.95	1563.28	-
		q^{RARe}	158.2	0.48	115.53	1773.18	1889.18	1.21×
SciFact	5183	q^{inst}	12.5	0	15.07	2.03	17.10	-
		q^{RARe}	1250.7	0.83	707.83	2.03	710.70	41.56×
SCIDOCS	25657	q^{inst}	9.4	0	11.29	5.74	17.03	-
		q^{RARe}	901.1	0.67	354.82	5.79	361.28	21.21×
CQADupStack	38100	q^{inst}	8.6	0	9.13	7.75	16.88	-
		q^{RARe}	678.2	1.15	466.23	7.79	475.17	28.15×
ClimateFEVER	5416593	q^{inst}	20.2	0	16.98	1124.36	1141.34	-
		q^{RARe}	831.3	2.31	424.60	1123.02	1549.93	1.36×

contributing factors to latency are the average length of input queries (**Avg Q len.**), and the size of the index (**# Corpus**). The increased time for query encoding is not exclusive to our approach, since query expansion methods Wang et al. (2023); Drozdov et al. (2023) also require encoding longer queries. **Moreover, the increased latency can be viewed**

as a form of test-time scaling, where the user trades-off throughput for performance depending on the preference. When no in-context examples are provided, we match the performance of the Instruct baseline. For large query length and small corpus sizes, the in-context setting demonstrates a significant increase in total latency (19.40-40.04× for FiQA2018 and NFCorpus, respectively). However, for smaller average query lengths, this latency diminishes, as seen for Quora (1.38×) and DBPedia (1.21×). Moreover, the added latency due to the in-context setting also diminishes when the corpus size grows (which is the case in several real world settings), as the time required for search outweighs the time to encode the query. For example, on Touche2020 with a larger corpus of 380K documents, the increase in latency is 4.76× compared to FiQA2018 (19.40×) for similar query lengths. Note that we did not perform any optimization that can further reduce latency. For instance, when the in-context examples are fixed, their KV representations can be pre-computed and cached, providing latency that is comparable to when only the original query is available. Furthermore, future research could explore several promising avenues, such as using efficient long-context retrievers (Saad-Falcon et al., 2024; Zhang et al., 2024) as a backbone, or developing extractive and/or abstractive compression techniques of in-context examples.

Table 11: **Impact of the number of in-context examples (k) at inference time.** $k = 5$ during training. All results are on *E5-Mistral-Instruct*. In general, performance increases when increasing the number of examples, and the optimal number of examples can vary by task.

		# Examples				
Dataset	**Instruct (0)**	**0**	**1**	**3**	**5**	**10**
ArguAna	61.19	**62.90**	61.24	60.99	61.18	60.37
ClimateFEVER	**39.03**	38.99	38.27	37.97	37.50	37.67
CQADupStack	44.82	45.57	47.49	48.33	48.46	**48.48**
DBPedia	48.92	49.24	49.79	48.34	49.65	**49.82**
FiQA2018	57.39	56.33	**57.61**	57.42	57.31	57.38
NFCorpus	40.99	41.19	41.48	42.10	42.28	**42.29**
SCIDOCS	17.94	18.71	19.83	20.17	20.19	**20.20**
SciFact	77.28	77.11	83.56	84.45	84.79	**85.12**
Touche2020	29.35	27.56	27.53	27.70	28.70	**30.77**
TRECCOVID	72.89	77.03	76.96	78.99	**79.58**	78.77
Average	48.98	49.46	50.38	50.65	50.96	**51.09**

Table 12: **Impact of the number of in-context examples (k) during training and inference.** All results are on *E5-Mistral-Instruct*. In general, performance increases when increasing the number of examples, and the optimal number of in-context examples can vary by task.

		# Examples				
Dataset	**Instruct (0)**	**0**	**1**	**3**	**5**	**10**
Arguana	61.19	**62.90**	60.47	62.98	60.87	58.85
ClimateFEVER	**39.03**	38.99	37.94	36.45	37.50	36.54
CQADupStack	44.82	45.57	46.76	47.12	48.46	**48.92**
DBPedia	48.92	49.24	47.70	49.05	**49.65**	47.95
FiQA2018	**57.39**	56.33	56.07	57.08	57.31	57.03
NFCorpus	40.99	41.19	40.67	40.77	**42.28**	42.24
SCIDOCS	17.94	18.71	20.01	19.28	20.19	**21.54**
SciFact	77.28	77.11	81.47	83.71	84.79	**87.61**
Touche2020	**29.35**	27.56	29.78	27.12	28.70	28.29
TRECCOVID	72.89	77.03	78.95	73.25	79.58	**86.11**
Average	48.98	49.46	50.18	48.83	51.11	**53.16**

B.3 Choice of In-Context Examples

Table 12 provides detailed numbers for varying in-context examples on all OOD BeIR tasks. Table 14 provides detailed numbers for various prompt formats on all OOD BeIR tasks.

Table 13: **In-Context Format** Comparing variants of in-context example format on *E5-Mistral-Instruct* during inference only. Training is done with the Regular format. Instruct refers to the baseline which does not use any in-context examples.

	Instruct	RARe				
Dataset	-	Query-Only	Doc-only	Shuffle-NC	Shuffle-C	Regular
ArguAna	**61.19**	57.36	60.35	55.64	60.49	60.87
ClimateFEVER	39.03	38.35	38.32	37.44	**37.84**	37.50
CQADupStack	44.82	39.56	48.43	47.70	48.27	**48.46**
DBPedia	48.92	49.14	49.69	49.72	**50.04**	49.65
FiQA2018	57.39	55.67	56.85	56.64	**57.41**	57.31
NFCorpus	40.99	41.00	42.09	42.02	41.92	**42.28**
SCIDOCS	17.94	19.06	20.06	19.98	**20.25**	20.19
SciFact	77.28	77.46	81.88	81.51	82.20	**84.79**
Touche2020	**29.35**	27.04	29.02	28.60	29.31	28.70
TRECCOVID	72.89	75.11	79.97	79.07	**80.03**	79.58
Average	48.98	47.98	50.67	49.83	50.78	**50.93**

Table 14: **In-Context Format** Comparing variants of in-context example format on *E5-Mistral-Instruct*. Instruct refers to the baseline which does not use any in-context examples.

	Instruct	RARe				
Dataset	-	Query-Only	Doc-Only	Shuffle-NC	Shuffle-C	Regular
ArguAna	**61.19**	58.88	57.54	60.17	58.97	60.87
ClimateFEVER	**39.03**	36.21	35.59	30.83	35.71	37.50
CQADupStack	44.82	46.66	48.28	45.78	47.97	**48.46**
DBPedia	48.92	49.98	49.08	**50.93**	50.24	49.65
FiQA2018	**57.39**	54.44	56.02	54.25	55.98	57.31
NFCorpus	40.99	41.42	41.62	41.17	41.78	**42.28**
SCIDOCS	17.94	20.04	20.12	20.35	20.11	**20.19**
SciFact	77.28	78.84	79.80	80.70	80.51	**84.79**
Touche2020	**29.35**	28.09	29.01	29.18	28.97	28.70
TRECCOVID	72.89	79.54	**83.29**	82.14	82.97	79.58
Average	48.98	49.41	50.04	49.55	50.32	**50.93**

B.4 Mixture of Training Data

In Table 15, we analyze the impact of training with only in-context examples when starting from decoder-only LLMs. As opposed to starting from existing retriever models, which have been trained without in-context examples, we observe that performance drops in the instruction-only setting. This can be largely mitigated by considering a mixture of in-context and instruction-only queries.

B.5 Applicability to Other Embedding Tasks

In Table 16, we report the performance on a subset of classification tasks from MTEB Muennighoff et al. (2023). We observe similar trends as retrieval, where providing in-context examples further enhances performance over providing task-specific instructions. The results underscore the applicability of RARe on embedding tasks beyond retrieval.

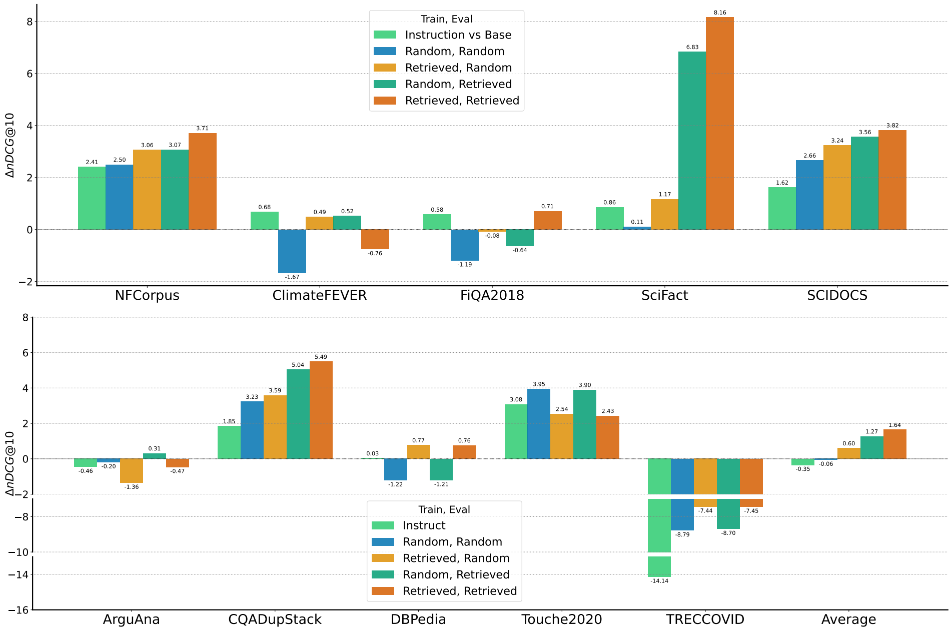

Figure 5: **Retrieved vs. Random In-context Examples.** Change in performance (ΔnDCG@10) on *E5-Mistral-Instruct* with RARe ($q^{\text{inst+ic}}$) from the baseline setting (q both during training and evaluation time). Using retrieved examples during training and/or inference enhance model performance in 7/10 datasets.

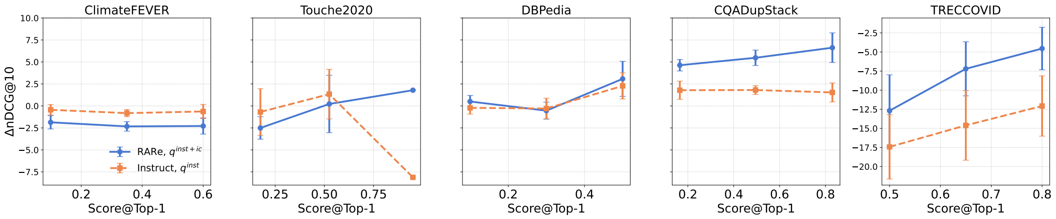

Figure 6: Change in performance (ΔnDCG@10) from the base model (*E5-Mistral-Instruct*) for varying similarity between the closest in-context example query and target query (Score@Top-1). Incorporation of semantically similar examples is beneficial on 3/5 datasets.

Table 15: Performance (nDCG@10) on datasets from the BeIR benchmark Thakur et al., 2021 when training decoder-only model (*Llama3*). Applying RARe with only in-context examples can lead to degradation of performance in the zero-shot setting (q^{inst}), but this is easily mitigated my including a mixture of q^{inst} and q^{RARe} data (30% and 70%) respectively.

Training	**Eval**	**NQ**	**Quora**	**NFCorpus**	**SciFact**	**SCIDOCS**	**FiQA2018**	**CQA**	**Average**
RepLLaMA-q^{inst}	q^{inst}	62.70	88.34	37.73	75.02	19.66	44.31	42.59	52.91
RARe-q^{RARe}	q^{inst}	39.64	**88.39**	35.42	74.52	**21.04**	30.44	37.74	46.74
	q^{RARe}	65.19	86.79	38.87	**78.41**	19.70	46.58	**43.75**	54.18
RARe-$q^{\text{inst}} + q^{\text{RARe}}$	q^{inst}	63.68	87.84	38.06	76.07	20.11	46.02	42.99	53.54
	q^{RARe}	**65.96**	87.65	**39.34**	77.20	19.45	**47.13**	42.96	**54.24**

Table 16: Performance (accuracy) with *E5-Mistral-Instruct* on classification tasks, demonstrating the applicability of RARe on other embedding domains and tasks.

Method	**Banking77**	**Emotion**	**Intent**	**ToxicChat**	**Avg**
Base	81.41	58.41	77.07	81.67	74.64
Instruct	83.27	58.00	77.57	82.31	75.29
RARe	**86.98**	57.18	**78.11**	**83.99**	**76.57**

Method	**DBPedia**	**NFCorpus**	**SciFact**	**Covid**	**Touche2020**
E5 (Wang et al., 2024a)	40.7	35.0	70.4	74.1	30.9
E5-Query2Doc (Wang et al., 2023)	42.4	35.2	67.5	75.1	31.7
E5-Mistral-Instruct (Wang et al., 2024b)	48.92	40.99	77.28	72.89	29.35
RARe	49.65	42.28	84.79	79.58	28.7

Table 17: Comparisons against PRF method which use generative LLM to augment queries.

Table 18: **Comparison against PRF methods.** The RARe pipeline i.e. BM25 based in-context selector and LLM-based retriever significantly outperforms methods where a generative LLM is used as a rewriter and BM25 as a retriever (Lei et al., 2024; Jedidi et al., 2024).

Method	**Retriever**	**PRF /ICL Model**	**#Used Params**	**Arguana**	**Covid**	**FiQA**	**SciFact**	**DBPedia**	**NFCorpus**
ReDE-RF	BM25	Mistral-7B	7B	-	65.60	29.30	66.90	37.60	35.50
CSQE	BM25	ChatGPT-3.5	Unknown	40.30	74.20	25.00	69.60	40.30	-
RARe	E5-Mistral-7B	BM25	7B	**60.87**	**79.58**	**57.31**	**84.79**	**49.65**	**42.28**

Table 19: Performance of RARe against Instruct variant trained with 5× steps. RARe performs better, especially on OOD settings, while Instruct begins to overfit on the in-doomain data.

Method	**BeIR**		**RAR-b**
	ID	OOD	
Instruct	72.91	48.98	24.12
Instruct (5x Steps)	72.97	48.04	24.02
RARe (5 examples)	72.98	50.93	25.79