OPEN-RAG: OPTIMIZING RAG END-TO-END VIA IN-CONTEXT RETRIEVAL LEARNING

Jiawei Zhou   Lei Chen
The Hong Kong University of Science and Technology
{jzhoubu,leichen}@ust.hk

ABSTRACT

In this paper, we analyze and empirically demonstrate that the relevance learned for traditional information retrieval scenarios may not consistently apply to retrieval-augmented generation (RAG) in wild environments. To bridge this gap, we introduce **OPEN-RAG**, a RAG framework that is **OP**timized **EN**d-to-end by tuning the retriever to capture in-context, **open**-ended relevance, enabling adaptation to the diverse and evolving needs. Extensive experiments across a wide range of tasks demonstrate that OPEN-RAG, by tuning a retriever end-to-end, leads to a consistent improvement of 4.0% over the original retriever, consistently outperforming existing state-of-the-art retrievers by 2.1%. Additionally, our results show that for certain tasks, a 0.2B retriever tuned end-to-end can achieve improvements surpassing those of RAG-oriented or instruction-tuned 8B LLMs, underscoring the cost-effectiveness of our approach for improving RAG systems.

1 INTRODUCTION

As the large language models (LLMs) (Zhao et al., 2023; Minaee et al., 2024) scale, they face a data bottleneck where the high-quality internet data unable to meet growing training demands. Meanwhile, the volume of downstream data is expanding rapidly but often remains unusable for pre-training due to their real-time availability (Wang et al., 2024b; Liu et al., 2023), privacy concerns (Arora et al., 2023), licensing restrictions (Min et al., 2024), and ethical concern (Serouis & Sèdes, 2024; Ayyamperumal & Ge, 2024). Retrieval-augmented generation (RAG) (Lewis et al., 2020; Guu et al., 2020; Gao et al., 2023) emerges as a promising solution to this challenge. Rather than relying solely on well-curated internet data, RAG leverages information retrieval (Manning, 2009) to fetch relevant data from external wild data sources and incorporates it as context to enhance generation quality. This is particularly valuable as RAG enables the use of rapidly expanding downstream data in wild, which are more scalable and up-to-date compared to the heavily processed and regulated internet data used in pre-training.

Despite their success, existing RAG frameworks typically rely on off-the-shelf retrievers trained on QA datasets, which can lead to inconsistencies between the learned retrieval relevance and the needs of downstream tasks. This discrepancy highlights key relevance gaps between IR and RAG scenarios. We explore these gaps in detail below, drawing on insights from prior research. First, there is the **broadening of tasks**: traditional IR datasets (Kwiatkowski et al., 2019; Bajaj et al., 2016) are designed mainly for open-domain question-answering (OpenQA), while RAG framework are applied to a wider range of tasks, such as recommendation (Manzoor & Jannach, 2022), dialog systems (Liu et al., 2024), and role-playing (Wang et al., 2023), where task requirements can be flexibly written as instructions. We refer to relevance in these two cases as *QA relevance* and *in-context relevance*, respectively, as shown in Figure 1. Second, the **role of retrieved documents** has shifted: in IR, retrieved documents are the final output provided to users, whereas in RAG, they are fed into the LLM to generate a response. Recent studies (Cuconasu et al., 2024a;b; Wu et al., 2024) have shown that including more answer-containing documents, which align with QA relevance in IR scenarios, can harm RAG performance, while documents without direct answers may actually help. These findings challenge traditional IR assumptions in the RAG setting. Finally, the **complexity of queries** has increased: unlike traditional IR, where queries are typically simple questions, RAG queries tend to be more diverse and noisy, reflecting varying levels of task complexity. Several studies highlight the challenges of complex queries and suggest that refining queries (Chan et al.,

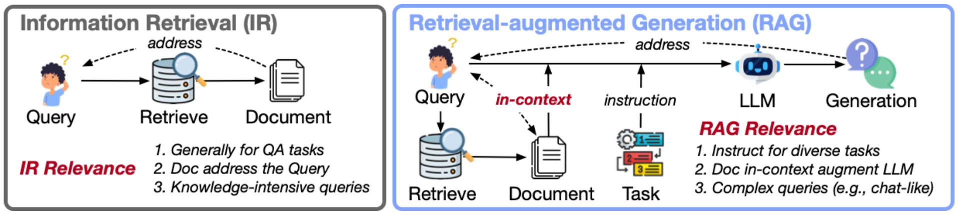

Figure 1: Comparison of query-document relevance in IR scenario and RAG scenario.

2024) or generating task-specific queries (Wu & Cao, 2024; Koo et al., 2024) based on documents can significantly enhance RAG performance.

To address this gap, we introduce **OPEN-RAG**, a RAG framework that is **OP**timized **EN**d-to-end by tuning the retriever to capture **open**-ended relevance in wild. During training, OPEN-RAG on-the-fly retrieves documents and identifies them as positives or negatives for contrastive learning. To reduce the training costs, we approximate the autoregressive generation process using likelihood threshold and employ semi-parametric retrieval (Zhou et al., 2024) to avoid re-indexing. Our training process requires only four GPU cards and can be completed within a day. Extensive experiments demonstrate that our method, initialized with an average retriever, adapts well to RAG scenarios and leads to significant performance improvements of up to 4.0%, consistently outperforming state-of-the-art retrievers by 2.2%. For certain tasks, our improvements even surpass those achieved by tuning an 8B LLM, showcasing that end-to-end retrieval learning is a cost-effective approach for enhancing RAG systems.

Our contribution can be summarized as follows:

• We analyze the relevance gap between IR and RAG scenarios and empirically show when and how it negatively impacts the performance of RAG in wild. Following our experiments, we identify potential biases in previous research that could hinder progress in this field. These findings offer important insights for future research.

• We propose OPEN-RAG, an end-to-end optimized RAG framework that learns in-context retrieval for downstream tasks. Experiments show that our framework trains a standard retriever to better adapt to downstream RAG scenarios, consistently outperforming existing SOTA retrievers.

• OPEN-RAG enables the learning of any natural-language-defined relevance without human annotations, overcoming major dataset construction challenges in IR research and facilitating RAG deployment in wild.

2 PRELIMINARY

2.1 TRANSFERRING FROM IR TO RAG SCENARIOS

In Table 1, we present the performance of off-the-shelf retrievers across different datasets, from IR to RAG scenarios. Details about the datasets and retrievers can be found in Appendix A and B, while the evaluation metric is described in Section 4.1. Key findings are summarized below.

Superiority of retrievers in IR scenarios can transfer to RAG scenarios in QA tasks but not in wild task. For QA tasks, a retriever with higher accuracy in IR scenarios tends to perform better in RAG scenarios on the NQ and TQA datasets. However, this trend does not hold for non-QA tasks, such as the PubHealth and ARC datasets, where a relatively weaker retriever, DPR_{MS}, outperforms others on the PubHealth dataset, and the unsupervised retriever CONTRIEVER surpasses all advanced retrievers on the ARC dataset.

Training in-domain is effective for both IR and RAG. As shown, with comparable training complexity, SIDR_{NQ} excels on the NQ dataset relative to other SIDR and DPR models. Additionally, SIDR_{TQA} outperforms the state-of-the-art retriever E5 in RAG scenarios on the TriviaQA dataset.

Significant untapped potential in datastores is limited by current retrieval capability. We use the *upper bound* to demonstrate the proportion of queries that can be addressed by using the top-1 retrieved document from any of the retrievers mentioned above. For each dataset, the *upper bound* exceeds the best performance by approximately 20%. This suggests that for most queries, the data-

Table 1: Accuracy in IR and RAG scenarios using Llama3-8b with top-1 retrieved document in-context; **Bold**: best performance; Δ: improvement or decline compared to SIDR_{MS}; §: has been trained in-domain.

Dataset (→)	NQ				TriviaQA				PubHealth		ARC-C	
Retriever (↓)	***IR***	Δ	***RAG***	Δ	***IR***	Δ	***RAG***	Δ	***RAG***	Δ	***RAG***	Δ
Unsupervised Pre-training												
Contriever	23.6	-15.5	30.9	-3.5	37.2	-18.9	56.6	-5.4	61.8	-1.7	**58.6**	**+1.7**
E5-unsup	30.8	-8.3	33.4	-1.0	39.5	-16.6	54.3	-7.7	62.9	-0.6	58.3	+1.4
Supervised on MSMARCO												
DPR_{MS}	38.9	-0.2	34.9	+0.5	43.7	-12.4	55.2	-6.8	64.5	+1.0	56.3	-0.6
SiDR_{MS}	39.1	–	34.4	–	56.1	–	62.0	–	63.5	–	56.9	–
Supervised on NQ												
DPR_{NQ}	‡43.5	+4.4	‡38.5	+4.1	39.4	-16.7	55.9	-6.1	62.9	-0.6	56.6	-0.3
SiDR_{NQ}	‡49.5	+10.4	‡42.7	+8.3	47.4	-8.7	59.8	-2.2	63.5	–	57.1	+0.2
Supervised on TQA												
DPR_{TQA}	32.1	-7.0	32.9	-1.5	‡55.4	-0.7	‡61.1	-0.9	63.1	-0.4	56.7	-0.2
SiDR_{TQA}	30.6	-8.5	32.9	-1.5	‡56.9	+0.8	‡**63.6**	**+1.6**	61.1	-2.4	**58.6**	**+1.7**
Pre-training + Supervised on Multiple Datasets												
$\text{Contriever}_{\text{MS}}$	41.5	+2.4	36.5	+2.1	53.5	-2.6	60.7	-1.3	63.1	-0.4	58.1	+1.2
E5	‡**58.0**	**+18.9**	‡**43.2**	**+8.8**	**58.7**	**+2.6**	63.2	+1.2	**64.7**	**+1.2**	58.0	+1.1
Upper-bound	65.9		77.6		78.5		80.3		92.1		71.5	

store contains at least one relevant document that could be retrieved by one of the retrievers and would be useful for RAG. However, no single retriever can capture all relevant documents, and the gap remains substantial.

Motivated by these observations, our work aims to develop a RAG framework that learns retrieval relevance in an end-to-end manner, going beyond the traditional retrieval relevance learned from existing QA datasets.

2.2 Problem Setup

A RAG framework typically consists of:

- A retriever $\mathcal{R}_\theta$ parameterized by θ
- A large language model $\mathcal{G}_\phi$ parameterized by ϕ
- A task $\mathcal{T}$ presented as an instruction prompt
- A datastore $\mathcal{D}$ with a vast number of documents d
- A user query q
- The answers a to the query
- An evaluation metric $\textsc{Eval}$ determining whether the output generation addresses the query

The downstream RAG pipeline generally follows:

1. Retrieve the top-k relevant documents from the $\mathcal{D}$ based on q, with a relevance function f_θ:

$$\{\hat{d}\}_k = \mathcal{R}_\theta(q, \mathcal{D}, k) \triangleq \underset{d \in \mathcal{D}}{\text{argmax}_k} f_\theta(q, d)$$

2. Formulate the task-specific prompt x using the query q and the retrieved documents $\{\hat{d}\}_k$.
3. Generate response $\hat{y}$ from input x via LLM, denoted as $\hat{y} = \mathcal{G}_\phi(x)$.
4. Evaluate if the generation $\hat{y}$ reflects the answer a:

$$\textsc{Eval}(\hat{y}) = \begin{cases} 1 & \text{if } \hat{y} \text{ reflects } a, \\ 0 & \text{otherwise.} \end{cases}$$

The goal of Open-Rag is to train the retriever component to maximize the likelihood of generating a response $\hat{y}$ that meets the downstream evaluation metric, which can be formulated as:

$$\hat{\theta} = \underset{\theta}{\text{argmax}} \sum_{\forall q} \textsc{Eval}(\hat{y} \mid \mathcal{R}_\theta, \mathcal{G}_\phi, \mathcal{T}, \mathcal{D}, q)$$

3 METHODOLOGY

In this section, we introduce **OPEN-RAG**, an **OP**timized **EN**d-to-end **RAG** framework designed to fine-tune a retriever to capture in-context, **open**-ended relevance, optimizing it for the downstream RAG pipeline.

To summarize, OPEN-RAG training comprises two stages: offline RAG and online RAG. The primary goal is to on-the-fly identify positive and negative documents for the contrastive learning of the retriever. An illustration of our framework is depicted in Figure 2.

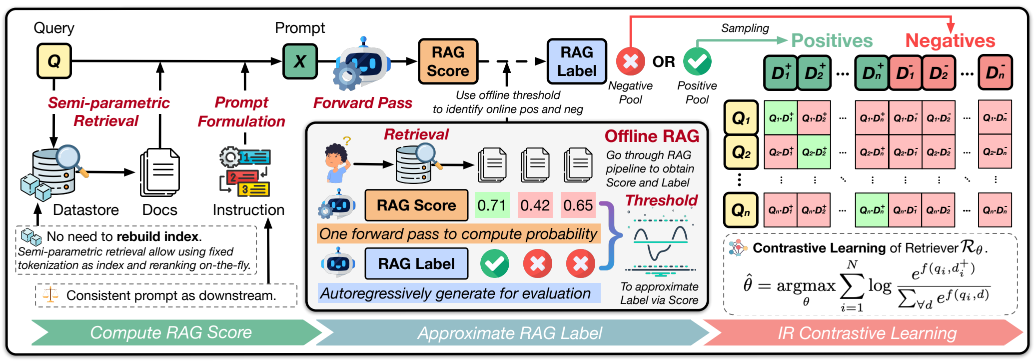

Figure 2: Illustration of the OPEN-RAG training process.

3.1 PRELIMINARY CONCEPTS

Continuation y and Generation $\hat{y}$. In knowledge-intensive generative tasks, information is aggregated and prompted as input x to a LLM for generation. The expected output could be an answer string a in question-answering tasks or might be a choice label c in reasoning and fact-checking tasks. Here, we refer to the expected output as the ground truth continuation, denoted as y, and the actual output generated by the LLM as $\hat{y}$. In a well-performing RAG framework, it is generally expected that $\hat{y} = y$ or that $\hat{y}$ contain or reflect y.

RAG Label. Given a query q, the RAG label $\mathcal{L}_d^q$ for a document d is a binary value that indicates whether the RAG outcome, when d is used in the context, meets the evaluation metric. The computation involves the following steps:

$$x = \mathit{Prompt}_{\mathcal{T}}(q, d); \quad \hat{y} = \mathcal{G}_\phi(x)$$
$$\mathcal{L}_d^q \triangleq \textsc{Eval}(\hat{y})$$

This assessment is typically based on whether the generated response contains the answers. The computation of RAG labels aligns with downstream inference, which involves autoregressive generation. For a clearer understanding, we provide examples in Appendix G.

RAG Score. Given a query q, the RAG score $\mathcal{S}_d^q$ of a d is the joint probability that LLM generates continuation y with d in context:

$$x = \mathit{Prompt}_{\mathcal{T}}(q, d)$$
$$\mathcal{S}_d^q \triangleq P_\phi(y \mid x) = \prod_{\forall t_i \in y} P_\phi(t_i \mid t_{<i}, x)$$

Here, $y = (t_1, \ldots, t_n)$ is a sequence of n tokens and P_ϕ is the function measures the probability of generating the next token or spans. Unlike the RAG label, the computation of the RAG score requires only a single forward pass of the LLM.

3.2 OFFLINE RAG

For offline RAG, we follow the traditional RAG pipeline as mentioned in Section 2.2. Given a query q, we retrieve top-k documents and denote this retrieved subset as $\mathcal{D}_q \subset \mathcal{D}$ where $|\mathcal{D}_q| = k$. We then compute the RAG label and score for each retrieved document d_i, resulting in the set

$\{(q, d_i, \mathcal{L}^q_{d_i}, \mathcal{S}^q_{d_i})\}^k_{i=1}$. Based on their RAG labels, $\mathcal{D}_q$ is further divided into a positive pool $\mathcal{D}^+_q$ and a negative pool $\mathcal{D}^-_q$. In our experiments, we set k to 100 and discard any sample where either pool is empty.

These RAG offline preparation serve two purposes. First, they establish initial positive and negative query-document pairs to warm up the retriever, enabling it to capture relevance in tasks. Second, they provide insights into the relationship between the RAG score and the RAG label. Specifically, we want to determine when the RAG score is above a certain threshold, the RAG label is 1, and when the RAG score is below a threshold, the label is 0. This relationship will be used to approximate the label via the score during online RAG training.

3.3 ONLINE RAG

In-training Retrieval. During retriever training, as its parameters update, the index needs to be rebuilt accordingly, which incurs significant costs. To address this challenge, we employ the semi-parametric retriever SIDR (Zhou et al., 2024). Specifically, SIDR incorporates both a parametric and a non-parametric encoder. The parametric encoder embeds text input x into a sparse representation with $|V|$ dimensions, where each dimension signifies the importance of a token within the language model's vocabulary V, denoted as $V_\theta(x)$. Conversely, the non-parametric encoder converts x into bag-of-tokens representation, referred to as $V_{\text{BoT}}(x)$, which is constructed via a tokenizer and is independent of θ. SIDR is strategically trained to allow the embedded query $V_\theta(q)$ to search on both an embedding-based index $V_\theta(\mathcal{D})$ and a bag-of-tokens index $V_{\text{BoT}}(\mathcal{D})$.

We adopt the late parametric mechanism of SIDR, which firstly retrieve the top-m documents using the bag-of-tokens index $V_{\text{BoT}}(\mathcal{D})$, denoted as:

$$\{\hat{d}\}_m = \mathcal{R}_\theta(V_\theta(q), V_{\text{BoT}}(\mathcal{D}), m)$$

These retrieved documents are then embedded and re-ranked on-the-fly to yield the top-k well-ranked documents, where $k < m$:

$$\{\hat{d}\}_k = \mathcal{R}_\theta(V_\theta(q), V_\theta(\{\hat{d}\}_m), k)$$

For late parametric mechanism, we set $m = k = 20$ to reduce training cost. More details of SIDR can be found in Appendix D.

Identifying Positives and Negatives On-the-fly. During training, we denote the pool of top-k retrieved documents as $\hat{\mathcal{D}}_q$. Our goal is to divide $\hat{\mathcal{D}}_q$ into a positive pool $\hat{\mathcal{D}}^+_q$ and a negative pool $\hat{\mathcal{D}}^-_q$ without the need for autoregressive generation. We present how to achieve this identification in two generation scenarios.

For *free-form generation*, such as in question answering tasks, the continuation y typically consists of a multi-token answer string. We identify a retrieved document $\hat{d}$ as positive if its RAG score surpasses the highest RAG score in the offline negative pool $\mathcal{D}^-_q$, and as negative if it is below the lowest RAG score in the offline positive pool $\mathcal{D}^+_q$:

$$\hat{\mathcal{L}}^q_{\hat{d}} = \begin{cases} 1, & \text{if } \mathcal{S}^q_{\hat{d}} > \max\{\mathcal{S}^q_d \mid \forall d \in \mathcal{D}^-_q\} \\ 0, & \text{if } \mathcal{S}^q_{\hat{d}} < \min\{\mathcal{S}^q_d \mid \forall d \in \mathcal{D}^+_q\} \end{cases}$$

Here, we use $\hat{\mathcal{L}}$ to denote the online RAG label, as it involves certain approximation. The approximation is based on the *assumption* that a higher RAG score correlates with an increased probability that the generated output $\hat{y}$ will match or reflect the target y. If computational resources are not a concern, ideally, one could perform autoregressive generation and evaluation on-the-fly or employ a larger LLM for identification purposes. We provide further discussion and verification of this assumption in Appendix C.

For *closed-set generation*, such as in multiple-choice reasoning or fact-checking tasks, the continuation y is typically a single-token choice label or can be prompted as such. In this case, we can relax the assumptions:

$$\hat{\mathcal{L}}^q_{\hat{d}} = \begin{cases} 1, & \text{if } P_\phi(c_i \mid x) > \max\{P_\phi(c_j \mid x) \mid \forall j \neq i\} \\ 0, & \text{otherwise.} \end{cases}$$

Here, x is the input prompt and c_i is the correct single-token choice while c_j are the incorrect choices. This setup checks whether LLM is more likely to generate c_i as the next token following x instead of c_j, when $\hat{d}$ is used in context.

For both scenarios, if a query has multiple correct continuation y (answers or choices), each y is treated as an individual entry. If $\hat{d}$ succeeds on at least one of these entries, we label it as positive; if it fails all of them, we label it as negative.

Sampling and Cache. During the online phase, we retrieve the top-k documents and compute their RAG scores to approximate RAG labels, processing them in descending order of retrieval relevance. We stop this process at the first document classified as negative. We then use this highest relevant negative, denoted as $\hat{d}^-$, and randomly select one positive $\hat{d}^+$ from the pool $\hat{\mathcal{D}}_q^+$. If either is unavailable, we fallback to random sampling from offline positive pool $\mathcal{D}_q^+$ or negative $\mathcal{D}_q^-$. To avoid redundant calculations, we cache all the online scores and labels $\{(q, \hat{d}_i, \hat{\mathcal{L}}_{\hat{d}_i}^q, \hat{\mathcal{S}}_{\hat{d}_i}^q)\}$ for reuse.

3.4 Contrastive Learning

Throughout our offline and online efforts, our objective is to acquire high-quality positive and negative query-document pairs for the contrastive learning (Jaiswal et al., 2020) of the retriever $\mathcal{R}_\theta$. Our training objective remains the same as SiDR to maintain its ability for late parametric. Given a batch B that consist of N samples, each sample consists of a query q_i, a positive document d_i^+, and a negative document d_i^-. Our training objective aims to maximize the similarity of positive query-document pairs $f(q_i, d_i^+)$ for all instances i, while minimize the similarity of all negative pairs, denoted as $f(q_i, d)$ for all $d \neq d_i^+$. The contrastive loss can be defined as follows:

$$L(q,d) = -\sum_{i=1}^{N}(\log \underbrace{\frac{e^{f(q_i,d_i^+)}}{\sum_{\forall d \in B} e^{f(q_i,d)}}}_{\text{q-to-d}} + \log \underbrace{\frac{e^{f(d_i^+,q_i)}}{\sum_{\forall q \in B} e^{f(d_i^+,q_i)}}}_{\text{d-to-q}})$$

The final loss integrates contrastive loss of both parametric and semi-parametric components:

$$\begin{aligned} L_{\text{para}}(q,d) &= L(V_\theta(q), V_\theta(d)) \\ L_{\text{semi-para}}(q,d) &= L(V_\theta(q), V_{\text{BoT}}(d))/2 + L(V_{\text{BoT}}(q), V_\theta(d))/2 \\ L_{\text{final}}(q,d) &= L_{\text{para}}(q,d) + L_{\text{semi-para}}(q,d) \end{aligned}$$

4 Experiments

4.1 Experimental Setup

Tasks and Datasets. We evaluate Open-Rag on four public RAG benchmarks. For free-form generation, we utilize Natural Questions (NQ; Kwiatkowski et al., 2019) and TriviaQA (TQA; Joshi et al., 2017), two well-established open-domain QA datasets. For closed-set generation, we employ the PubHealth (Kotonya & Toni, 2020) dataset for fact-checking tasks, and the ARC-Challenge (Clark et al., 2018) dataset for multiple-choice reasoning. More information about the datasets can be found in Appendix A.

Evaluation Metrics. Following previous works (Asai et al., 2023; Mallen et al., 2023), we use accuracy as the evaluation metric and report results on the test set. In IR scenarios, accuracy is measured by whether the retrieved documents contain the expected answers, while in RAG scenarios, it is assessed based on the generated output. Since our training uses 1 document in context while existing research generally uses 10 for RAG, we report accuracy with both 1 and 10 documents in context for comparison.

Implementation Details. Our RAG system employs the LLM Llama3-8b (Dubey et al., 2024) with the retriever SiDR$_{\text{MS}}$ (Zhou et al., 2024) that trained on MS MARCO dataset (Bajaj et al., 2016). We use the same English Wikipedia datastore and prompt as those open-sourced by Self-RAG, detailed in Appendix H. During training, we train the retriever for each dataset for 80 epochs, aligning with the training duration used for SiDR$_{\text{MS}}$. We use a batch size of 128 and an AdamW optimizer (Loshchilov & Hutter, 2018) with a learning rate of 2×10^{-5}. The training process is divided into two phases: the first half involves a warm-up phase using offline positives and negatives, while the second half transitions to in-training retrieval, primarily using the positives and negatives identified on-the-fly. During inference, we set the maximum number of generated token to be 100 for free-form generation while 20 for closed-set generation. Our experiments are conducted with 4 NVIDIA A100 GPUs. Both offline RAG preparation and online RAG training take less than one day, depending on the number of queries in the datasets. We leverage vLLM (Kwon et al., 2023) to accelerate offline generation.

Baselines. We consider the baselines detailed below, with additional model information provided in Appendix B. (1) *Standard RAG with advanced IR*: RAG frameworks using Llama3-8b and state-of-the-art retrievers E5 (Wang et al., 2022) and CONTRIEVER$_{\text{MS}}$ (Izacard et al., 2021). We refer to OPEN-RAG (SIDR$_{\text{NQ}}$) as our framework utilizing SIDR$_{\text{NQ}}$ as the initial retriever. For a fair comparison, we compare E5 with OPEN-RAG (SIDR$_{\text{NQ}}$), both of which have been trained on the NQ dataset. (2) *RAG with IR tuning*: RAG frameworks that incorporate a tunable IR component. We compare against REPLUG (Shi et al., 2023), which uses part of a sequence as query to retrieve documents which maximize the generation likelihood of the remaining part. Since the model weights are not publicly available, we reference a reproduction by (Yue et al., 2024) that uses the top-3 retrieved documents in context. (3) *RAG with LLM tuning*: RAG frameworks that incorporate RAG-oriented or instruction-tuned LLMs, which typically require more resources for tuning an 8B LLM. We compare with SELF-RAG (Asai et al., 2023) using Llama2-7B, along with some reproductions (Zhang et al., 2024; Wang et al., 2024d) employing more recent LLMs. Our primary comparison with SELF-RAG and its variants is designed to ensure a controlled and fair evaluation, as we adhere to the same prompts and downstream evaluation pipeline. (4) *Transferring to other LLMs*: We compare the RAG framework using different LLMs, such as Llama3-Instruct$_{\text{8B}}$ (Dubey et al., 2024), Phi-3-mini-4k-instruct$_{\text{3.8B}}$ (Abdin et al., 2024), Mistral-Instruct$_{\text{7B}}$ (Jiang et al., 2023), along with SIDR$_{\text{MS}}$ before and after tuning. This setup is designed to evaluate whether the learned in-context relevance transfers across different LLMs.

4.2 MAIN EXPERIMENTS

Table 2: Main results of OPEN-RAG and other RAG baselines on 4 datasets, using top-1 and top-10 retrieved documents in context. **Bold**: best RAG method that does not involve LLM tuning. Δ: improvement or decline; ▲: baseline that below methods compare with; †: reproduction from other works; ‡: our reproduction; §: has been trained in-domain.

Task Type (→)	**Free-form**								**Closed-set**							
Dataset (→)	**NQ**				**TriviaQA**				**PubHealth**				**ARC-C**			
Method (↓) Metrics (→)	**1-doc**	**Δ**	**10-doc**	**Δ**	**1-doc**	**Δ**	**10-doc**	**Δ**	**1-doc**	**Δ**	**10-doc**	**Δ**	**1-doc**	**Δ**	**10-doc**	**Δ**
Standard RAG																
Baseline IR																
Llama3$_{\text{8B}}$ + SIDR$_{\text{MS}}$	34.4	▲	37.6	▲	62.0	▲	62.5	▲	63.5	▲	64.9	▲	56.9	▲	57.5	▲
Llama3$_{\text{8B}}$ + SIDR$_{\text{NQ}}$	§42.7	+8.3	§41.6	+4.0	–	–	–	–	–	–	–	–	–	–	–	–
Advanced IR																
Llama3$_{\text{8B}}$ + CONTRIEVER$_{\text{MS}}$	36.5	+2.1	38.3	+0.7	60.7	-1.3	60.6	-1.9	63.1	-0.4	62.9	-2.0	**58.1**	**+1.2**	**58.9**	**+1.4**
Llama3$_{\text{8B}}$ + E5	§**43.2**	**+8.8**	§**41.8**	**+4.2**	63.2	+1.2	61.4	-1.1	64.7	+1.2	63.7	-1.2	58.0	+1.1	58.1	+0.6
RAG with IR tuning																
†REPLUG$_{\text{Llama2-7B}}$ (3-doc, Yue et al. (2024))	–	–	–	–	–	–	–	–	–	–	41.7	–	–	–	47.2	–
Ours																
OPEN-RAG	39.8	+5.4	40.9	+3.3	**65.8**	**+3.8**	**66.2**	**+3.7**	**69.5**	**+6.0**	**69.3**	**+4.4**	**58.1**	**+1.2**	58.3	+0.8
OPEN-RAG (SIDR$_{\text{NQ}}$)	§**44.1**	**+9.7**	§**44.7**	**+7.1**	–	–	–	–	–	–	–	–	–	–	–	–
RAG with LLM tuning																
Llama3-Instruct$_{\text{8B}}$ + SIDR$_{\text{MS}}$	41.2	+6.8	52.1	+14.5	65.2	+3.2	73.3	+10.8	67.2	+3.7	71.8	+6.9	72.1	+15.2	75.5	+18.0
SELF-RAG$_{\text{Llama2-7B}}$ (Asai et al., 2023)	–	–	–	–	–	–	66.4	+3.9	–	–	72.4	+7.5	–	–	67.3	+9.8
†SELF-RAG$_{\text{Mistral-7B}}$ (Wang et al., 2024d)	–	–	–	–	–	–	64.8	+2.3	–	–	72.4	+7.5	–	–	74.9	+17.4
†SELF-RAG$_{\text{Llama3-8B}}$ (Zhang et al., 2024)	–	–	–	–	–	–	56.4	-6.1	–	–	67.8	+2.9	–	–	58.0	+0.5
‡SELF-RAG$_{\text{Llama3-8B}}$ + SIDR$_{\text{MS}}$	30.8	-3.6	37.0	-0.6	51.0	-11.0	57.7	-4.8	64.2	+0.7	64.0	-0.9	58.9	+2.0	59.1	+1.6
Transferring OPEN-RAG to other LLM																
Llama3-Instruct$_{\text{8B}}$ + SIDR$_{\text{MS}}$	41.2	▲	52.1	▲	65.2	▲	73.3	▲	67.2	▲	71.8	▲	72.1	▲	75.5	▲
Llama3-Instruct$_{\text{8B}}$ + OPEN-RAG (SIDR$_{\text{MS}}$)	43.6	+2.4	54.7	+2.6	65.6	+0.4	73.8	+0.5	65.2	-2.0	66.1	-5.7	71.9	-0.2	75.0	-0.5
Phi-3-mini-4k-instruct$_{\text{3.8B}}$ + SIDR$_{\text{MS}}$	40.6	▲	49.2	▲	64.6	▲	69.2	▲	48.2	▲	57.6	▲	84.9	▲	84.3	▲
Phi-3-mini-4k-instruct$_{\text{3.8B}}$ + OPEN-RAG (SIDR$_{\text{MS}}$)	43.4	+2.8	50.3	+1.1	65.6	+1.0	70.4	+1.2	45.3	-2.9	54.4	-3.2	85.1	+0.2	84.6	+0.3
Mistral-Instruct$_{\text{7B}}$ + SIDR$_{\text{MS}}$	37.5	▲	48.0	▲	58.2	▲	57.1	▲	50.1	▲	57.4	▲	69.7	▲	71.5	▲
Mistral-Instruct$_{\text{7B}}$ + OPEN-RAG (SIDR$_{\text{MS}}$)	40.5	+3.0	49.4	+1.4	59.8	+1.6	57.6	+0.5	46.7	-3.4	54.6	-2.8	69.2	-0.5	70.6	-0.9

Table 2 presents results of OPEN-RAG and baselines, with key findings summarized below:

End-to-end tuning effectively improves the retriever in RAG scenarios, surpassing existing SOTA retrievers. Unlike E5 and CONTRIEVER$_{\text{MS}}$, which require extensive pre-training, OPEN-

RAG framework improves SIDR$_{\text{MS}}$ using four GPUs within a day. This efficient approach leads to a notable 4.0% enhancement in performance beyond the original SIDR$_{\text{MS}}$ and consistently achieves a 2.1% better outcome than the SOTA retrievers. For PubHealth, the improvement reaches up to 6%, a significant value that even using instruction-tuned LLMs cannot achieve. For ARC, the modest improvement can be attributed to its limited number of training samples, only a few hundred, compared to other datasets containing tens of thousands. These results demonstrate that, despite approximation, the learned in-context relevance is more effective than the inconsistent relevance derived from existing datasets. In Appendix F, we show that improve the retriever for RAG scenarios may degrade its performance in traditional IR scenarios, further reinforcing this inconsistency.

Relevance learning constitutes a valuable yet overlooked dimension for improving the RAG system. Reproductions of SELF-RAG using Llama3-8B by other works (Zhang et al., 2024; Wang et al., 2024d) and ourselves have not yielded consistent improvements. This suggests that despite the substantial training expenses, enhancing RAG through tuning LLM requires extensive customization and does not reliably generalize. In contrast, tuning a smaller-sized retriever can lead to comparable, or in some cases, superior improvements over those achieved by RAG-oriented or instruction-tuned 8B LLMs on specific datasets. Importantly, learning an in-context retriever does not conflict with LLM enhancements, offering a complementary avenue for improving the RAG system.

The learned in-context retriever can be transferred to other LLMs for free-form generation tasks. Our results show that OPEN-RAG, initially co-trained with Llama3-8b, enhances other LLMs such as Llama3-Instruct-8B, Phi-3-mini-4k-instruct, and Mistral-Instruct in free-form generation tasks. However, for closed-set generation tasks, this transferability does not consistently hold. Despite the limitations, OPEN-RAG significantly enhances performance of PubHealth by a large margin. We hypothesize that closed-set tasks, where the continuation is a single token, are easier to optimize due to less approximation involved. Consequently, the retriever learns a very specific relevance tailored to the particular LLM prediction of the next token, complicating its transferability. Therefore, we recommend end-to-end tuning on a LLM-by-LLM basis to potentially improve outcomes for these tasks.

4.3 ABLATION STUDY

Compared to prior works, our main differences include (i) employing contrastive learning instead of KL divergence to induce supervision signals from the LLM to the IR, and (ii) using late parametric to avoid periodic re-indexing. We systematically analyze these factors in this section.

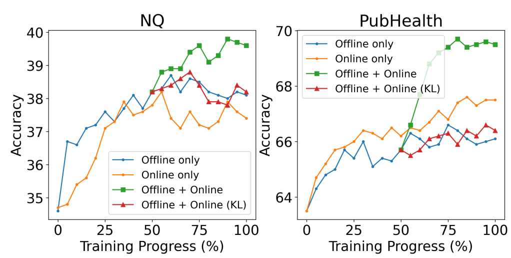

Figure 3: Ablation studies on NQ and Pubhealth datasets.

As shown in Figure 3, we conducted an ablation study on NQ and PubHealth with several setup: our method is labeled as *[offline+online]*, where *[offline-only]* represents using only the offline positives and negatives for contrastive learning, and *[online-only]* indicates that we do not use any warmup. We also explore using KL divergence *[offline+online(KL)]* instead of contrastive learning.

Offline versus Online. During the warmup stage, documents are retrieved using the initial parameters θ. During the in-training retrieval stage, they are retrieved using the up-to-date parameters θ'. We assess the improvements provided by the in-training retrieval stage. As shown in Figure 3, relying solely on either *[offline-only]* or *[online-only]* can lead to suboptimal improvements, proving to be less effective than a combination of a warmup phase followed by online in-training retrieval *[offline+online]*. This observation echoes the conclusions of prior research (Zhou et al., 2024), which indicates that warming up the retriever to initially capture the in-task relevance, followed by in-training retrieval to continuously explore potential positives and challenging negatives in the datastore, can significantly enhance performance.

Contrastive Learning versus KL-Divergence. Prior works (Shi et al., 2023; Guu et al., 2020) have employed KL divergence to align query-document relevance with the distribution of generation likelihood. Our experiments indicate that while KL divergence leads to improvements, these benefits quickly stabilize and the overall enhancement falls short of our method. Unlike our approach, which employs contrastive learning requiring efforts to identify positives and negatives, KL divergence alignment offers a straightforward but potentially overly restrictive solution. On one hand, in RAG scenarios, documents are delivered to LLMs, differing from IR scenarios where documents must be well-ranked before being presented to users. For a proficient LLM, including even a single useful document in the context window should suffice (Cuconasu et al., 2024a). On the other hand, similar works in knowledge distillation (Gou et al., 2021), which uses cross-encoder scores to guide bi-encoder training, demonstrate that improvements for bi-encoders are limited and cannot match the performance of cross-encoder rerankers. Consequently, the prevalent industry practice of retrieve-then-rerank (Gupta et al., 2018) underscores the current limitations of retrievers in capturing complex relationships. We believe that the distribution of generation likelihood from LLMs is too complex for these small-sized retriever to accurately capture, thereby resulting in less improvement.

Late Parametric versus Periodic Re-indexing. Due to page limitations, we detail our comparison of different in-training retrieval methods in Appendix E. This comparison particularly focuses on the late parametric method versus prior solutions that utilize an embedding index and require periodic re-indexing. Our results indicate that the late parametric method not only leads to better improvements but also reduces training costs and simplifies the implementation. We believe that the high costs and complex implementation associated with periodic re-indexing have prevented previous research from effectively training retrievers on a task-by-task basis, using consistent instructions, LLMs, and datastores tailored to downstream tasks, ultimately leading to less effective results.

5 RELATED WORKS

Retrieval-augmented Generation (RAG). The RAG system combines LLMs, retrievers, and datastores, each contributing to performance improvement. Significant research has focused on improving RAG by tuning LLMs to address challenges such as enhancing on-demand retrieval (Asai et al., 2023; Jeong et al., 2024), optimizing response efficiency (Wang et al., 2024d), and enabling self-reasoning capabilities (Li et al., 2024). Additional efforts have explored building domain-specific (Wang et al., 2024e) or extremely large datastores (Shao et al., 2024) to enhance RAG performance. While some studies have focused on the retrieval aspect, emphasizing adaptive retrieval strategies where the retriever does not learn (Wang et al., 2024a;c), and leveraging LLMs to develop more potent retrievers (Guu et al., 2020; Shi et al., 2023), research on end-to-end learning of relevance for RAG scenarios remains limited. Our work seeks to address this gap, pioneering new avenues in the application of RAG systems.

Relevance Learning. Relevance learning is an important and longstanding area of research. Traditionally, text relevance has been defined by heuristic rules based on term overlap, as seen in the widely-used BM25 algorithm (Robertson et al., 2009). With advances in deep learning, neural retrievers have emerged (Karpukhin et al., 2020), learning relevance from human-annotated datasets (Kwiatkowski et al., 2019). Further research has explored pre-training retrievers using large-scale, weakly supervised text pairs, such as cropped text spans within documents (Izacard et al., 2021) and relational text pairs extracted from web data (Zhou et al., 2022; Wang et al., 2022), to enable retrievers to learn general relevance. This general relevance can then be refined to task-specific and domain-specific relevance through downstream fine-tuning, resulting in improved performance. Our method falls within these advancements, where the LLM acts as a container of general relevance, providing on-the-fly supervision of specific in-context relevance for relevance learning.

6 CONCLUSION

In this work, we demonstrate that traditional retrieval relevance derived from QA datasets is inconsistent in RAG scenarios. To bridge this gap, we introduces OPEN-RAG, a RAG framework that learns in-context retrieval for downstream tasks. Our framework shows that through end-to-end tuning, a standard retriever can surpass SOTA off-the-shelf retrievers in RAG scenarios. Furthermore, for certain datasets, our RAG framework, which tunes a 0.2B retriever, performs better than other RAG frameworks that tune an 8B LLM. This underscores the substantial potential of in-context retrieval learning to enhance RAG performance, particularly for wild tasks that are unseen by the retriever and LLM during pre-training.

ACKNOWLEDGMENTS

Lei Chen's work is partially supported by National Key Research and Development Program of China Grant No. 2023YFF0725100, National Science Foundation of China (NSFC) under Grant No. U22B2060, Guangdong-Hong Kong Technology Innovation Joint Funding Scheme Project No. 2024A0505040012, the Hong Kong RGC GRF Project 16213620, RIF Project R6020-19, AOE Project AoE/E-603/18, Theme-based project TRS T41-603/20R, CRF Project C2004-21G, Guangdong Province Science and Technology Plan Project 2023A0505030011, Guangzhou municipality big data intelligence key lab, 2023A03J0012, Hong Kong ITC ITF grants MHX/078/21 and PRP/004/22FX, Zhujiang scholar program 2021JC02X170, Microsoft Research Asia Collaborative Research Grant, HKUST-Webank joint research lab and 2023 HKUST Shenzhen-Hong Kong Collaborative Innovation Institute Green Sustainability Special Fund, from Shui On Xintiandi and the InnoSpace GBA.

REFERENCES

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024.

Simran Arora, Patrick Lewis, Angela Fan, Jacob Kahn, and Christopher Ré. Reasoning over public and private data in retrieval-based systems. *Transactions of the Association for Computational Linguistics*, 11:902–921, 2023.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511*, 2023.

Suriya Ganesh Ayyamperumal and Limin Ge. Current state of llm risks and ai guardrails. *arXiv preprint arXiv:2406.12934*, 2024.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*, 2016.

Chi-Min Chan, Chunpu Xu, Ruibin Yuan, Hongyin Luo, Wei Xue, Yike Guo, and Jie Fu. Rq-rag: Learning to refine queries for retrieval augmented generation. *arXiv preprint arXiv:2404.00610*, 2024.

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.

Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. The power of noise: Redefining retrieval for rag systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 719–729, 2024a.

Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, Fabrizio Silvestri, et al. Rethinking relevance: How noise and distractors impact retrieval-augmented generation. In *CEUR WORKSHOP PROCEEDINGS*, volume 3802, pp. 95–98. CEUR-WS, 2024b.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, Volume 1 (Long and Short Papers)*, pp. 4171–4186, 2019.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2023.

Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129:1789–1819, 2021.

Vishal Gupta, Manoj Chinnakotla, and Manish Shrivastava. Retrieve and re-rank: A simple and effective ir approach to simple question answering over knowledge graphs. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pp. 22–27, 2018.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval augmented language model pre-training. In *International conference on machine learning*, pp. 3929–3938. PMLR, 2020.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Towards unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*, 2021.

Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2020.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C Park. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. *arXiv preprint arXiv:2403.14403*, 2024.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

Mandar Joshi, Eunsol Choi, Daniel S 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)*, pp. 1601–1611, 2017.

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)*, pp. 6769–6781, 2020.

Zixuan Ke, Weize Kong, Cheng Li, Mingyang Zhang, Qiaozhu Mei, and Michael Bendersky. Bridging the preference gap between retrievers and LLMs. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10438–10451, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.562. URL `https://aclanthology.org/2024.acl-long.562/`.

Hamin Koo, Minseon Kim, and Sung Ju Hwang. Optimizing query generation for enhanced document retrieval in rag. *arXiv preprint arXiv:2407.12325*, 2024.

Neema Kotonya and Francesca Toni. Explainable automated fact-checking for public health claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7740–7754, 2020.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33: 9459–9474, 2020.

Huayang Li, Pat Verga, Priyanka Sen, Bowen Yang, Vijay Viswanathan, Patrick Lewis, Taro Watanabe, and Yixuan Su. Alr: A retrieve-then-reason framework for long-context question answering. *arXiv preprint arXiv:2410.03227*, 2024.

Xiao-Yang Liu, Guoxuan Wang, and Daochen Zha. Fingpt: Democratizing internet-scale data for financial large language models. *arXiv preprint arXiv:2307.10485*, 2023.

Zihan Liu, Wei Ping, Rajarshi Roy, Peng Xu, Chankyu Lee, Mohammad Shoeybi, and Bryan Catanzaro. Chatqa: Surpassing gpt-4 on conversational qa and rag. *arXiv preprint arXiv:2401.10225*, 2024.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.

Alex Troy Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *The 61st Annual Meeting Of The Association For Computational Linguistics*, 2023.

Christopher D Manning. *An introduction to information retrieval*. Cambridge university press, 2009.

Ahtsham Manzoor and Dietmar Jannach. Towards retrieval-based conversational recommendation. *Information Systems*, 109:102083, 2022.

Sewon Min, Suchin Gururangan, Eric Wallace, Weijia Shi, Hannaneh Hajishirzi, Noah A. Smith, and Luke Zettlemoyer. SILO language models: Isolating legal risk in a nonparametric datastore. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=ruk0nyQPec`.

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. Large language models: A survey. *arXiv preprint arXiv:2402.06196*, 2024.

Jinming Nian, Zhiyuan Peng, Qifan Wang, and Yi Fang. W-rag: Weakly supervised dense retrieval in rag for open-domain question answering. *arXiv preprint arXiv:2408.08444*, 2024.

Stephen Robertson, Hugo Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009.

Ibrahim Mohamed Serouis and Florence Sèdes. Exploring large language models for bias mitigation and fairness. In *1st International Workshop on AI Governance (AIGOV) in conjunction with the Thirty-Third International Joint Conference on Artificial Intelligence*, 2024.

Rulin Shao, Jacqueline He, Akari Asai, Weijia Shi, Tim Dettmers, Sewon Min, Luke Zettlemoyer, and Pang Wei Koh. Scaling retrieval-based language models with a trillion-token datastore. *arXiv preprint arXiv:2407.12854*, 2024.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. Replug: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*, 2023.

Fei Wang, Xingchen Wan, Ruoxi Sun, Jiefeng Chen, and Sercan Ö Arık. Astute rag: Overcoming imperfect retrieval augmentation and knowledge conflicts for large language models. *arXiv preprint arXiv:2410.07176*, 2024a.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):1–26, 2024b.

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. *arXiv preprint arXiv:2212.03533*, 2022.

Xiaohua Wang, Zhenghua Wang, Xuan Gao, Feiran Zhang, Yixin Wu, Zhibo Xu, Tianyuan Shi, Zhengyuan Wang, Shizheng Li, Qi Qian, et al. Searching for best practices in retrieval-augmented generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 17716–17736, 2024c.

Xintao Wang, Yaying Fei, Ziang Leng, and Cheng Li. Does role-playing chatbots capture the character personalities? assessing personality traits for role-playing chatbots. *arXiv preprint arXiv:2310.17976*, 2023.

Zilong Wang, Zifeng Wang, Long Le, Huaixiu Steven Zheng, Swaroop Mishra, Vincent Perot, Yuwei Zhang, Anush Mattapalli, Ankur Taly, Jingbo Shang, et al. Speculative rag: Enhancing retrieval augmented generation through drafting. *arXiv preprint arXiv:2407.08223*, 2024d.

Zora Zhiruo Wang, Akari Asai, Xinyan Velocity Yu, Frank F Xu, Yiqing Xie, Graham Neubig, and Daniel Fried. Coderag-bench: Can retrieval augment code generation? *arXiv preprint arXiv:2406.14497*, 2024e.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. Ccnet: Extracting high quality monolingual datasets from web crawl data. *arXiv preprint arXiv:1911.00359*, 2019.

Mingrui Wu and Sheng Cao. Llm-augmented retrieval: Enhancing retrieval models through language models and doc-level embedding. *arXiv preprint arXiv:2404.05825*, 2024.

Siye Wu, Jian Xie, Jiangjie Chen, Tinghui Zhu, Kai Zhang, and Yanghua Xiao. How easily do irrelevant inputs skew the responses of large language models? *arXiv preprint arXiv:2404.03302*, 2024.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808*, 2020.

Peng Xu, Wei Ping, Xianchao Wu, Lawrence McAfee, Chen Zhu, Zihan Liu, Sandeep Subramanian, Evelina Bakhturina, Mohammad Shoeybi, and Bryan Catanzaro. Retrieval meets long context large language models. *arXiv preprint arXiv:2310.03025*, 2023.

Yue Yu, Wei Ping, Zihan Liu, Boxin Wang, Jiaxuan You, Chao Zhang, Mohammad Shoeybi, and Bryan Catanzaro. Rankrag: Unifying context ranking with retrieval-augmented generation in llms. *arXiv preprint arXiv:2407.02485*, 2024.

Shengbin Yue, Siyuan Wang, Wei Chen, Xuanjing Huang, and Zhongyu Wei. Synergistic multi-agent framework with trajectory learning for knowledge-intensive tasks. *arXiv preprint arXiv:2407.09893*, 2024.

Xuanwang Zhang, Yun-Ze Song, Yidong Wang, Shuyun Tang, Xinfeng Li, Zhengran Zeng, Zhen Wu, Wei Ye, Wenyuan Xu, Yue Zhang, et al. Raglab: A modular and research-oriented unified framework for retrieval-augmented generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 408–418, 2024.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

Jiawei Zhou, Xiaoguang Li, Lifeng Shang, Lan Luo, Ke Zhan, Enrui Hu, Xinyu Zhang, Hao Jiang, Zhao Cao, Fan Yu, et al. Hyperlink-induced pre-training for passage retrieval in open-domain question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7135–7146, 2022.

Jiawei Zhou, Li Dong, Furu Wei, and Lei Chen. Semi-parametric retrieval via binary token index. *arXiv preprint arXiv:2405.01924*, 2024.

A Details of Datasets

We present details of datasets as follows.

- Natural Questions (NQ; Kwiatkowski et al., 2019) is a widely used open-domain QA dataset constructed from Wikipedia. The questions originate from Google search queries, and the answers are text spans within Wikipedia passages. This dataset consists of queries with one or more answer strings, requiring RAG systems to generate responses based on factual knowledge.
- TriviaQA (TQA; Joshi et al., 2017) is a challenging QA dataset that comprises question-answer pairs curated by trivia enthusiasts along with independently gathered evidence documents.
- PubHealth (Kotonya & Toni, 2020) is a fact-checking task that focuses on verifying health claims across a variety of biomedical topics.
- ARC-Challenge (Clark et al., 2018) is a multiple-choice reasoning dataset consisting of science exam questions for grades 3 to 9.

B Details of Baseline Models

The information for baseline models are listed as follows.

B.1 Retrieval Model (IR)

- E5 (Wang et al., 2022) is a state-of-the-art dense retriever that pre-trained on millions of weakly related text pairs from the Web. The unsupervised version of this model is denoted as E5-unsup. This model undergoes further fine-tuning on natural language inference (NLI) datasets, as well as the Natural Questions and MS MARCO datasets, to enhance its capabilities in downstream applications. The fine-tuned version is denoted as E5.
- Contriever (Izacard et al., 2021) is a widely-used dense retriever pre-trained unsupervised on Wikipedia data and CCNet (Wenzek et al., 2019). The unsupervised version of this model is denoted as Contriever. It is further fine-tuned on the MS MARCO dataset to enhance its retrieval performance, with the fine-tuned version denoted as $\text{Contriever}_{\text{MS}}$.
- DPR (Karpukhin et al., 2020) is a widely used dense passage retriever initialized with a BERT-based uncased encoder (Devlin et al., 2019), and fine-tuned on downstream dataset. Specifically, DPR_{MS} is fine-tuned on the MS MARCO dataset, DPR_{NQ} on the NQ dataset, and DPR_{TQA} on the TriviaQA dataset.
- SiDR (Zhou et al., 2024) is a semi-parametric sparse retriever that supports using both embeddings and tokenization as index. This nature allows for in-training retrieval, where the model's parameters dynamically update while the retrieval index remains fixed. The model is initialized with a BERT-based uncased encoder (Devlin et al., 2019) and fine-tuned exclusively on single dataset depending on the variant: SiDR_{MS} is fine-tuned on the MS MARCO dataset, SiDR_{NQ} on the NQ dataset, and SiDR_{TQA} on the TriviaQA dataset.

All the above retrieval methods are initialized with a BERT-based encoder, which contains approximately 200 million (0.2B) parameters.

B.2 Large Language Model (LLM)

- $\text{Llama3}_{\text{8B}}$ (Dubey et al., 2024) is a variant of the latest Llama3 model series with 8 billion parameters.
- $\text{Llama3-Instruct}_{\text{8B}}$ (Dubey et al., 2024) builds upon the $\text{Llama3}_{\text{8B}}$ by undergoing a post-training stage in which the model is specifically tuned to follow instructions and align with human preferences to improve specific capabilities.
- $\text{Phi-3-mini-4k-instruct}_{\text{3.8B}}$ (Abdin et al., 2024) is a lightweight widely-used LLM with 3.8 billion parameters, trained on the Phi-3 dataset featuring synthetic and high-quality filtered web data, focused on reasoning and quality.
- $\text{Mistral-Instruct}_{\text{7B}}$ (Jiang et al., 2023). We use Mistral-7B-Instruct-v0.3 LLM which is an instruct fine-tuned version of the Mistral-7B-v0.3.

B.3 Retrieval-augmented Generation Framework (RAG)

- RePlug (Shi et al., 2023) is a RAG framework using GPT-3 and Contriever. The retriever is specifically trained to use the first 128 tokens of a sequence as queries, with the goal of retrieving documents that maximize the probability of generating the subsequent 128 tokens when these retrieved documents are prepended to the query.

- Self-RAG (Asai et al., 2023) is a RAG framework designed to improve response quality by enabling on-demand retrieval and incorporating self-reflection mechanisms.

  The reproductions by Wang et al. (2024d) and Zhang et al. (2024), Self-RAG$_{\text{Mistral-7B}}$ and Self-RAG$_{\text{Llama3-8B}}$ respectively, involve tuning Mistral-7B and Llama3-8B as base language models using the open-source data provided by Self-RAG.

  Our reproduction, Self-RAG$_{\text{Llama3-8B}}$+SiDR$_{\text{MS}}$, utilizes the Self-RAG$_{\text{Llama3-8B}}$ checkpoint from Zhang et al. (2024) as LLM, while employing the same retriever SiDR$_{\text{MS}}$ and adapting it to our downstream setup.

B.4 Challenges and Prior Work

Major Challenges. There are two major challenges in training a RAG framework end-to-end via tuning retriever. (i) The primary challenge involves the extreme computational costs associated with deploying such a pipeline in training. These costs mainly arise from two sources: first, the LLMs generate sequences autoregressively, which is inherently resource-intensive; secondly, as θ updates, the retrieval index need to be rebuilt accordingly, adding further computational demands. (ii) The second challenge is ensuring stable and effective back-propagation of supervision signals from the final outcome of the RAG pipeline to the retriever.

Prior Practices. Prior research (Guu et al., 2020; Xu et al., 2023; Shi et al., 2023) has explored the joint training of retrievers with LLMs for RAG. Despite extensive efforts, they often default to learning a universal relevance, where the retrieved document aids in generating the continuation of a natural language input, while neglecting the specific downstream components $\mathcal{T}$, $\mathcal{D}$, $\mathcal{G}_\phi(x)$ and Eval. These general approaches lead to a significant discrepancy as the components used during training do not align with those employed during inference. As a result, these methods often fall short in meeting the specific, nuanced relevance needs of various downstream tasks.

B.5 Cost-Effectiveness Analysis

Regarding training costs, the primary expense comes from computing the RAG scores using the LLM. In Table 3, we report the number of documents required to compute RAG scores on-the-fly during training.

Throughout training, each query encounters between 15 to 128 unscored documents, depending on the task, requiring LLM forward passes to compute RAG scores on-the-fly. This process incurs a manageable cost, typically amounting to hours rather than days. We also observe a positive correlation between the number of documents processed and the performance improvements of OpenRag. Notably, the PubHealth dataset requires more documents to compute the RAG score online, resulting in the most significant improvement. This suggests that encountering more unscored documents indicates a larger gap in relevance between the initial and the learned retriever, highlighting the presence of more potentially useful documents in the datastore that could be leveraged by in-context retrieval learning.

	NQ	**TriviaQA**	**PubHealth**	**ARC**
nDoc	20	18	128	15
Improv.	+5.4%	+3.8%	+6.0%	+1.2%

Table 3: Number of documents required on-the-fly RAG score computation and the improvement for each task.

Table 4: Results of RAG framework using top-1 and top-10 documents in context, sorted by retrieval relevance and RAG scores.

Task Type (→)	Free-form				Closed-set			
Dataset (→)	**NQ**		**TriviaQA**		**PubHealth**		**ARC-C**	
Method (↓) Metrics (→)	**1-doc**	**10-doc**	**1-doc**	**10-doc**	**1-doc**	**10-doc**	**1-doc**	**10-doc**
Llama3$_{8B}$ + SiDR$_{MS}$ (doc with top relevance)	49.1	51.4	65.3	67.2	65.2	67.4	58.1	57.3
Llama3$_{8B}$ + SiDR$_{MS}$ (doc with top RAG scores)	85.1	76.2	88.7	84.2	87.4	77.4	95.6	83.6

C Effectiveness of RAG Scores on Task Accuracy

Given that our learning is based on using the RAG score as an indicator to identify positive and negative documents, we now investigate whether using documents with higher RAG scores leads to improved RAG response quality. For each dataset, we sample 1k samples from training split. For each query, we retrieve the top 100 documents, and then perform the RAG pipeline using only the top-1 and top-10 documents, sorted by retrieval relevance and RAG scores, respectively. The results, shown in Table 4, indicate that RAG scores are indicative of the final accuracy of the RAG framework. Furthermore, the high accuracy achieved using top RAG scores documents suggests that the datastore holds significant untapped potential, which current retrieval strategies have not yet fully exploited.

To our knowledge, using RAG scores to identify positives and negatives is a rough yet resource-efficient solution that could cover most existing knowledge-intensive tasks, aligning with their evaluation metrics that often utilize string matching. However, it may not be suitable for long-form generation, which requires different evaluation strategies. We believe it is possible to customize the identification of positive and negative examples based on the specific needs of each task. Ideally, if computational cost is not a concern or resources are sufficient, a strong proprietary LLM like GPT-4 can be used for contrastive identification on-the-fly.

Here are some additional observations: RAG scores are generally more indicative when using single document in context, likely because they are computed in this manner, ensuring more consistent evaluations. Furthermore, the improved performance seen in Table 4 compared to our main experiments may be attributed to the LLM having been pretrained on the training split of these datasets.

D Revisiting Semi-parametric Disentangled Retriever (SiDR)

Our work adopts the recently proposed retriever SiDR as the backbone for two main reasons. First, it supports the use of a non-parametric index, which enables in-training retrieval when the retriever's parameters change dynamically. Second, evaluating retriever checkpoints can be resource-intensive, as it requires embedding a large datastore with each new checkpoint. SiDR offers late parametric techniques that reduce this evaluation process from a full day on our resource to just a few minutes, significantly accelerating our research.

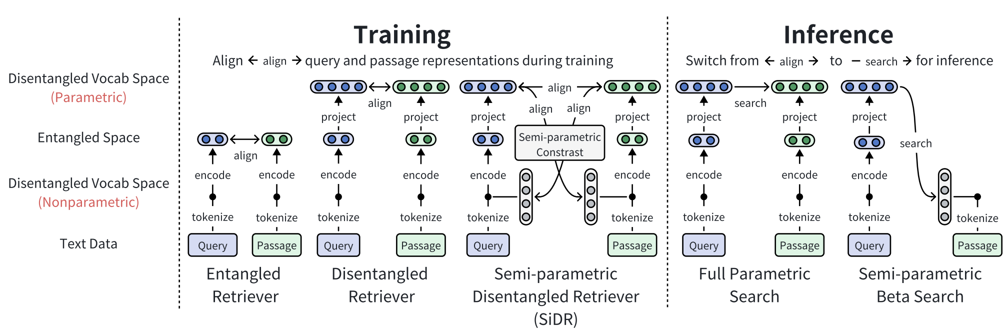

Figure 4: Illustration of semi-parametric disentangled retriever (SiDR) framework, adapted from Zhou et al. (2024).

SIDR is a sparse disentangled retriever (also known as a sparse lexical retriever) that encodes text chunks into a $|V|$-dimensional sparse representation, where each dimension represents the importance of a token within the language model vocabulary V. SIDR is then trained to align the $|V|$-dimensional parametric embedding, denoted as $V_\theta(x)$, with the $|V|$-dimensional bag-of-tokens representation, denoted as $V_{\text{BoT}}(x)$.

At downstream, a parametric query embedding $V_\theta(q)$ can perform search on both an embedding-based index $V_\theta(\mathcal{D})$ and a bag-of-tokens index $V_{\text{BoT}}(\mathcal{D})$, which leads to three distinct search schemes:

- **Full parametric search** utilizes a **parametric index** $V_\theta(\mathcal{D})$, which relies on embeddings derived from a neural encoder for the datastore. The relevance is defined as the inner product of the embeded query and embeded datastore:

$$f_\theta(q, \mathcal{D}) = \langle V_\theta(q), V_\theta(\mathcal{D}) \rangle$$

  This is the common indexing process for neural retrieval systems, which are effective but involve higher costs and longer latency for embedding the entire $\mathcal{D}$ to obtain the index $V_\theta(\mathcal{D})$.

- **Semi-parametric beta search** leverages a **non-parametric index** $V_{\text{BoT}}(\mathcal{D})$ based on BoT representations of the datastore, which are constructed solely by a tokenizer. The relevance is defined as:

$$f_\beta(q, \mathcal{D}) = \langle V_\theta(q), V_{\text{BoT}}(\mathcal{D}) \rangle$$

- **Late parametric with top-m re-rank** is a search pipeline that starts search with a non-parametric index to retrieve top-m passages, denote as $\mathcal{D}_m$, and then on-the-fly embeds them for re-ranking:

$$f_\beta(q, \mathcal{D}) = \langle V_\theta(q), V_{\text{BoT}}(\mathcal{D}) \rangle; \quad f_\theta(q, \mathcal{D}_m) = \langle V_\theta(q), V_\theta(\mathcal{D}_m) \rangle$$

In our framework, we primarily utilize the late parametric techniques provided by SIDR. For in-training retrieval, we use late parametric with top-20 re-ranking. For checkpoint evaluation and inspection in the ablation study, we use late parametric with top-100 re-ranking to accelerate results while managing limited resources. In our main experiments, we use full parametric search.

E Late Parametric vs. Periodic Re-indexing

A key distinction between our work and prior practices lies in our use of the late parametric mechanism to avoid re-indexing during training. In this section, we systematically evaluate these in-training retrieval approaches.

Baseline. We present ablation studies on different in-training retrieval approaches: (i) OPEN-RAG employs the late parametric method as proposed in SIDR, which uses a bag-of-token index for first-stage retrieval and re-ranks the top-20 documents on-the-fly using up-to-date parameters. (ii) OPEN-RAG (w/o re-rank) employs the bag-of-token index for retrieval, similar to the late parametric method but without the re-ranking process. This setup aims to assess the costs associated with re-ranking during training. (iii) OPEN-RAG (w/ re-index) involves periodic re-indexing using the most recently built but outdated index for retrieval, an in-training retrieval method that commonly used in prior studies. In this setup, we employ DPR_{MS} as the initial retriever. We avoid using SIDR_{MS}, which has high-dimensional embeddings of 30,522, in stark contrast to DPR's 768 dimensions. This significant discrepancy prevents our GPU cards from allocating the parametric index for SIDR_{MS}, although they manage DPR effectively.

Training. All models undergo the similar training pipeline: they are trained for 80 epochs with the first 40 epochs as a warm-up and the last 40 conducting in-training retrieval. They differ only in their in-training retrieval strategies: both OPEN-RAG and OPEN-RAG (w/o re-rank) do not require re-indexing; OPEN-RAG (w/ re-index) requires rebuilding index at every 15 epochs (around 5k steps), a rebuild interval commonly used in previous research (Xiong et al., 2020), resulting in a total of three rebuilds.

Results. We present the RAG accuracy on NQ and PubHealth test splits during in-training retrieval, with results reported every four epochs, as depicted in Figure 5. For the re-ranking setup, significant improvements are observed in the PubHealth data when re-ranking is employed, whereas the NQ dataset shows only minor improvements. Given that the costs associated with re-ranking are manageable in our setup, we continue to implement it. Regarding re-indexing, our findings indicate that despite requiring significant time and resources, it fails to yield improvements comparable to those of the late parametric approach and significantly lags behind. We attribute this to index staleness, where query embeddings must optimize against outdated document embeddings, rendering the learning process less effective. On the other hand, as presented in the study by Zhou et al. (2024), by re-ranking the top-20 retrieved documents, the late parametric method can recover more than 90% of the performance of a full parametric search across different tasks, representing a minor compromise. This also partially explains why the late parametric approach outperforms periodic re-indexing.

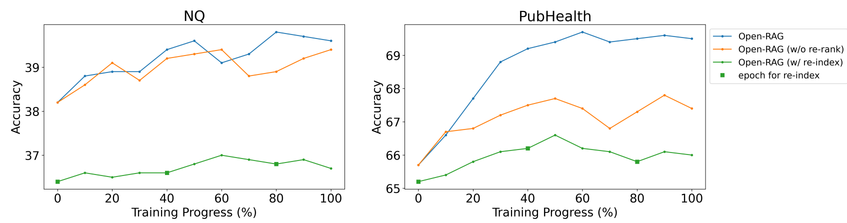

Figure 5: RAG accuracy of different in-training retrieval approaches.

F Inconsistencies between IR and RAG Scenarios

F.1 Performance Changes in IR Scenarios after Tuning

Table 5: Performance changes before and after tuning the retriever using the Open-Rag approach.

Dataset (→)	**NQ**		**TriviaQA**	
Method (↓) Metrics (→)	**IR**	**RAG**	**IR**	**RAG**
Llama3_{8B} + SiDR_{MS}	39.1	34.4	56.1	62.0
Llama3_{8B} + Open-Rag (SiDR_{MS})	40.8 (+1.7)	39.8 (+5.4)	53.9 (-2.2)	65.8 (+3.8)
Llama3_{8B} + SiDR_{NQ}	49.5	42.7	–	–
Llama3_{8B} + Open-Rag (SiDR_{NQ})	47.1 (-2.4)	44.1 (+1.4)	–	–

We evaluate the performance of our retriever in both IR and RAG scenarios before and after tuning. In IR scenarios, we measure top-1 retrieval accuracy by checking whether the top-1 retrieved document contains the answer. In RAG scenarios, we measure accuracy using a single document in the context window, evaluating whether the generated response contains the correct answer.

Our results indicate that while Open-Rag tunes the retriever to improve RAG performance, it results in inconsistent performance on traditional IR performance, with some degradation observed on certain datasets. This highlights a long-standing issue in the IR evaluation pipeline: a document containing the answer does not necessarily imply that it effectively addresses the query, and conversely, a document not containing the answer does not mean it is irrelevant or unhelpful.

Our conclusion also aligns with the findings and observations of other research. Cuconasu et al. (2024a) find that including more answer-containing documents in the context negatively impacts RAG performance. Similarly, Nian et al. (2024) observe that traditional relevance definitions for IR tasks do not enhance RAG response quality. Additional research emphasizes the need for further learning to bridge the preference gap (Ke et al., 2024) or re-ranking (Yu et al., 2024) for off-the-shelf retrievers to improve RAG performance.

F.2 Case Study

In this section, we present a case study using the NQ dataset where each query has a list of answer strings. This case study is designed to further explore the inconsistency issues inherent in RAG implementations. We specifically examine two scenarios: (i) cases where the retrieved document contains the correct answer but fails to produce the correct RAG output, and (ii) instances where the retrieved document does not directly address the query, yet the RAG model manages to generate the correct answer nonetheless. To enhance our analysis, we also ask GPT-4 to judge whether the documents address the question, helping readers quickly grasp the key issue.

```
======================================== Question ========================================
Who plays Big Momma in Big Momma’s House?
======================================== Answers =========================================
 [’Ella Mitchell’, ’Martin Lawrence’]
======================================== Document ========================================
Bounce with Me
Jermaine Dupri, Jagged Edge and Da Brat. Brief clips from "Big Momma’s House" are also
included. Bounce with Me "Bounce with Me" is a single by American rapper Lil’ Bow Wow
featuring Xscape. It is Lil’ Bow Wow’s debut single, taken from his debut album "Beware of
Dog". The single samples "Love Serenade (Part II)" by Barry White. It spent nine weeks at
number 1 on the U.S. Rap Chart and number 20 on "Billboard" Hot 100. The song was featured
in the film "Big Momma’s House" and the trailer of "Hardball".
===================================== RAG Generation =====================================
[1] Martin Lawrence
===================================== GPT4 Judgement =====================================
No, the document does not address the question. The document talks about the song "Bounce
with Me" and its connection to the movie Big Momma’s House, but it doesn’t mention who
plays Big Momma in the movie. The answer to the question is Martin Lawrence, who plays the
character of Big Momma in Big Momma’s House.
```

```
======================================== Question ========================================
Who sang "Buddy, Can You Spare a Dime?"
======================================== Answers =========================================
 [’Rudy Vallee’, ’Bing Crosby’]
======================================== Document ========================================
Brother, Can You Spare a Dime?
Brother, Can You Spare a Dime? "Brother, Can You Spare a Dime?", also sung as "Buddy, Can
You Spare a Dime?", is one of the best-known American songs of the Great Depression.
Written in 1930 by lyricist E. Y. "Yip" Harburg and composer Jay Gorney, "Brother, Can You
Spare a Dime?" was part of the 1932 musical revue "Americana"; the melody is based on a
Russian-Jewish lullaby Gorney’s mother had sung to him as a child. It was considered by
Republicans to be anti-capitalist propaganda, and almost dropped from the show; attempts
were made to ban it from the radio.
===================================== RAG Generation =====================================
The song "Brother, Can You Spare a Dime?" was written by E. Y. "Yip" Harburg and Jay Gorney
. It was first performed by Bing Crosby in 1932.
===================================== GPT4 Judgement =====================================
The document provides information about the song "Brother, Can You Spare a Dime?" (also
referred to as "Buddy, Can You Spare a Dime?") and its origins, but it does not directly
answer the question of who sang it. The document focuses more on the history of the song
and its cultural significance.

To answer your question: The song has been performed by several artists, including Bing
Crosby, Al Jolson, and Barbra Streisand, among others.
```

Figure 6: Example of RAG output correct answers while the retrieved document does not contain the correct answer, nor does GPT-4 consider the document address the question.

In Figure 6, we present examples where RAG outputs the correct answer, even though the retrieved document neither contains the answer nor is considered to address the question by GPT-4. In both cases, the document fails to provide the correct answer or relevant clues, yet RAG is still able to generate the correct response. We believe this is a common phenomenon, as LLMs possess a wealth of internal knowledge, particularly for public knowledge questions. In general, an incorrect or imperfect retrieved document is insufficient to mislead the LLM into producing an incorrect output.

```
======================================== Question ========================================
When was The Hitchhiker’s Guide to the Galaxy made?
======================================== Answers =========================================
[’1978’, ’2005’]
======================================== Document ========================================
The Hitchhiker’s Guide to the Galaxy (radio series)
The original series was the recipient of a number of awards including the Imperial Tobacco
Award (1978), The Sony Award (1979), The Society of Authors/Pye Awards Best Programme for
Young People (1980) and the Mark Time Awards ’Grand Master Award’ (Adams) and ’Hall of Fame
’ (1998). It was the only radio show ever to be nominated for the Hugo science fiction
awards, in 1979, in the ’Best Dramatic Presentation’ category. As a result of the series,
Douglas Adams was inducted into the Radio Academy’s Hall of Fame. In November 2003, two
years after Adams’s death and 23 years after the ...

===================================== RAG Generation =====================================
[1] The Hitchhiker’s Guide to the Galaxy (radio series)
===================================== GPT4 Judgement =====================================
No, the document does not directly address the query of when The Hitchhiker’s Guide to the
Galaxy was made. It provides information about the radio series, its awards, and
recognition but does not specify the year of its creation or first broadcast.

The original radio series of The Hitchhiker’s Guide to the Galaxy first aired in 1978.
```

```
======================================== Question ========================================
Who ordered the Forbidden City to be built?
======================================== Answers =========================================
[’Zhu Di’]
======================================== Document ========================================
History of the Forbidden City
The site of the Forbidden City was situated on the Imperial city during the Mongol Yuan
Dynasty. After the collapse of the Yuan Dynasty, the Hongwu Emperor of the Ming Dynasty
moved the capital from Beijing in the north to Nanjing in the south, and in 1369 ordered
that the Yuan palaces be razed. His son Zhu Di was created Prince of Yan with his seat in
Beijing. In 1402, Zhu Di usurped the throne and became the Yongle Emperor. He made Beijing
a secondary capital of the Ming empire ...

===================================== RAG Generation =====================================
The Yongle Emperor ordered the forbidden city to be built.
===================================== GPT4 Judgement =====================================
Yes, the document addresses the query. It mentions that Zhu Di, who later became the Yongle
 Emperor, ordered the construction of the Forbidden City. The passage describes how he
moved the capital to Beijing and commissioned the building of the Forbidden City in 1406.
```

Figure 7: Example of RAG failing to output the correct answer while the retrieved document contains the correct answer or GPT-4 considers the document as addressing the question.

In Figure 7, we present examples where RAG fails to output the correct answer, even though the retrieved document contains the correct answer or GPT-4 considers the document as addressing the question. In the first case, the document does not address the query, and the LLM tends to extract key phrases, such as the title, as the response, ignoring the query. In the second case, the document contains information that addresses the query, and the LLM generates the correct answer, but the answer’s alias name is not included in the pre-defined answer candidates, leading to a failure in the RAG scenario. These inconsistencies can be driven by many factors, including the LLM, instruction prompt, evaluation metrics, and relevance. All of these factors are intertwined, and we believe that end-to-end data-driven learning is more effective than analyzing their interplay in isolation.

G Examples of RAG Labels

For free-form generation tasks, we assess whether the generation contains any of the given answers. For closed-set generation tasks, we measure whether the generation contains the label. Below are examples that illustrate how different generations lead to different RAG labels given the same question and answers.

```
======================================== Question ========================================
Who is the sister of for king and country?
======================================== Answers ========================================
['Rebecca St. James']
====================================== Generation 1 ======================================
Rebecca St. James

### Explanation:
Rebecca St. James is the sister of for king and country. She is a Christian recording
artist and speaker. The Smallbones are the younger brothers of Rebecca St. James, and the
brothers-in-law of Jacob Fink, former bassist for the band Foster the People. They were
raised with another sister, Libby, and three other brothers, Ben, Dan, and Josh. On
======================================== Output 1 ========================================
RAG Label: 1
====================================== Generation 2 ======================================
Courtney Helm

### Explanation:
Courtney Helm is the sister of for king and country.
======================================== Output 2 ========================================
RAG Label: 0
```

Figure 8: Examples of RAG labels for free-form generation.

```
======================================== Question ========================================
Study: Vaccine for Breast, Ovarian Cancer Has Potential.
======================================== Answers ========================================
True
====================================== Generation 1 ======================================
True

### Explanation:
The study is about a vaccine for breast and ovarian cancer. The vaccine is
======================================== Output 1 ========================================
RAG Label: 1
====================================== Generation 2 ======================================
False

### Explanation:
The statement is false because the vaccine is not for breast and ovarian cancer.
======================================== Output 2 ========================================
RAG Label: 0
```

Figure 9: Examples of RAG labels for close-set generation.

H Prompt Formats

We demonstrate our prompts for different tasks such as OpenQA, fact-checking, and multi-choice reasoning in Figures 10, 11, and 12, respectively.

```
==================================== Prompt (OpenQA) ====================================
### Paragraph:
[1] For King & Country (band)
Their father was a music promoter, and Joel recalls, "going to these rock concerts, sitting
 on my father\'s shoulders, plugging my ears...Honestly, in a lot of ways I feel like music
 chose me and as I grew older, I made a clear decision to fully lean into it." The
Smallbones are the younger brothers of Christian recording artist and speaker Rebecca St.
James, and the brothers-in-law of Jacob Fink, former bassist for the band Foster the People
. They were raised with another sister, Libby, and three other brothers, Ben, Dan, and Josh
. On ...

### Instruction:
Who is the sister of for king and country?

### Response:
===================================== Continuation =====================================
Rebecca St. James
====================================== Generation ======================================
Rebecca St. James

### Explanation:
Rebecca St. James is the sister of for king and country. She is a Christian recording
artist and speaker. The Smallbones are the younger brothers of Rebecca St. James, and the
brothers-in-law of Jacob Fink, former bassist for the band Foster the People. They were
raised with another sister, Libby, and three other brothers, Ben, Dan, and Josh. On
======================================== Output ========================================
RAG Score: P('Rebecca St. James'|prompt) = 0.595
RAG Label: 1
```

Figure 10: Example prompt and outcomes of each step for NQ and TQA dataset.

```
================================ Prompt (Fact Checking) ================================
Below is an instruction that describes a task. Write a response that appropriately
completes the request.

### Paragraph:
[1] Gustav Gaudernack
potential of dendritic cells (DCs) and in 2005, Gaudernack's group published results from a
 phase I/II clinical trial in prostate cancer patients using autologous DCs loaded with
tumor mRNA as a vaccine. This study demonstrated that vaccination with autologous DCs
transfected with mRNA derived from three prostate cancer cell lines was safe and an
improved clinical outcome was significantly related to immune responses against the vaccine
. Furthermore, Gaudernack and colleagues initiated a phase I/II clinical trial for
treatment of malignant melanoma with autologous tumor-mRNA transfected DC vaccines. These
data clearly demonstrated vaccine-specific immune responses with a broad specter of ...

### Instruction:
Is the following statement correct or not? Say true if it's correct; otherwise say false.

### Input:
Study: Vaccine for Breast, Ovarian Cancer Has Potential

### Response:
===================================== Continuation =====================================
True
====================================== Generation ======================================
true

### Explanation:
The study is about a vaccine for breast and ovarian cancer. The study has ...
======================================== Output ========================================
P('true' |prompt) = 0.116
P('false'|prompt) = 0.109
RAG Label: 1
```

Figure 11: Example prompt and outcomes of each step for the Pubhealth dataset.

```
============================ Prompt (Multi-choice Reasoning) ============================
Below is an instruction that describes a task. Write a response that appropriately
completes the request.

### Paragraph:
[1] Rheumatic fever
Rheumatic fever may occur following an infection of the throat by the bacterium "
Streptococcus pyogenes". If the infection is untreated rheumatic fever can occur in up to
three percent of people. The underlying mechanism is believed to involve the production of
antibodies against a person\'s own tissues. Due to their genetics, some people are more
likely to get the disease when exposed to the bacteria than others. Other risk factors
include malnutrition and poverty. Diagnosis of RF is often based on the presence of signs
and symptoms in combination with evidence of a recent streptococcal infection. Treating
people who have strep ...

### Instruction:
Given four answer candidates, A, B, C and D, choose the best answer choice.

### Input:
Which factor will most likely cause a person to develop a fever?
A: a leg muscle relaxing after exercise
B: a bacterial population in the bloodstream
C: several viral particles on the skin
D: carbohydrates being digested in the stomach

### Response:
==================================== Continuation ======================================
B
===================================== Generation ======================================
B

### Explanation:
The bacteria Streptococcus pyogenes is a common cause of throat
======================================= Output ========================================
P('A'|prompt) = 0.121
P('B'|prompt) = 0.309
P('C'|prompt) = 0.061
P('D'|prompt) = 0.100
RAG Label: 1
```

Figure 12: Example prompt and outcomes of each step for the ARC-Challenge dataset.