RAG-LER: Retrieval Augmented Generation with LM Enhanced Reranker

Anonymous ACL submission

Abstract

Large language models (LLMs) have demonstrated impressive capabilities across diverse NLP tasks, yet they still struggle with hallucination due to limited parametric knowledge. Retrieval Augmented Generation (RAG) addresses this issue by integrating non-parametric data stores. However, straightforward integration of information retrieval or end-to-end training of these components often leads to suboptimal results or computational inefficiency. In this work, we introduce RAG-LER, a framework that enhances an LM's context understanding and improves the quality and accuracy of provided passages through an LM-supervised re-ranker. RAG-LER fine-tunes a pre-trained LM to follow instructions and discriminately use provided information. It then leverages this fine-tuned LM to generate ranking scores, which serve as supervised labels for training the re-ranker. By harnessing LLMs' strong capabilities, our approach eliminates the need for manual human labeling in re-ranker training while achieving improved performance. Experiments demonstrate that RAG-LER outperforms existing retrieval-augmented LMs on open-domain QA and fact-checking tasks, while exhibiting consistently improved performance when applied to different LMs, highlighting its versatility and effectiveness[1].

1 Introduction

Large language models have shown their capabilities on various tasks (Brown et al., 2020; Touvron et al., 2023b), but even with a huge number of parameters, they still have limited memorization of factual knowledge and are constrained by the outdated knowledge they were trained on. Retrieval Augmented Generation (RAG) (Lewis et al., 2020; Guu et al., 2020) augments the input with relevant passages retrieved from non-parametric corpus, reducing the hallucination in knowledge-intensive tasks and updating LLMs knowledge without training. Thus, the quality of retrieved passages becomes crucial during generation since irrelevant passages can lead to worse performance(Mallen et al., 2022; Shi et al., 2023a). Most previous approaches utilize several sub-modules like passage retrieval and re-ranking (Petroni et al., 2020a; Izacard and Grave, 2020; Izacard et al., 2023) for non-parametric knowledge selection. These approaches train the sub-modules with LLM end-to-end or combine them directly, training end-to-end improves the performance but reduces the modularity and costs large computational resources, combining these separately trained modules also underperforms as their different data representations.

This work presents RAG-LER, a retrieval-augmented framework that uses LLM's strong context understanding capability to enhance the reranker and takes the retriever as a replaceable submodule. RAG-LER is initialized with an arbitrary retriever, re-ranker, and a pre-trained LM. Given an input, RAG-LER retrieves relevant passages, each retrieved passage is prepended to the input individually and fed into the re-ranker, re-ranker re-scores these passages according to the relevance scores between these passages and the input, then the LM generates prediction given the input and re-ranked passages.

RAG-LER instruction-tunes an arbitrary LM on a diverse collection of instruction-following and text comprehension data. We augment the input with passages to train LM using the given passages' information. When irrelevant passages are presented, RAG-LER chooses to abstain (Zhou et al., 2023; Feng et al., 2024), we replace the label with a special token which indicates that the given passages have no useful information. This can be a signal to direct the passage retrieval and reduce the chance of hallucination. The key idea of our framework is to integrate the strong capability of LLM

[1] Our code and training data will be made publicly available after review.

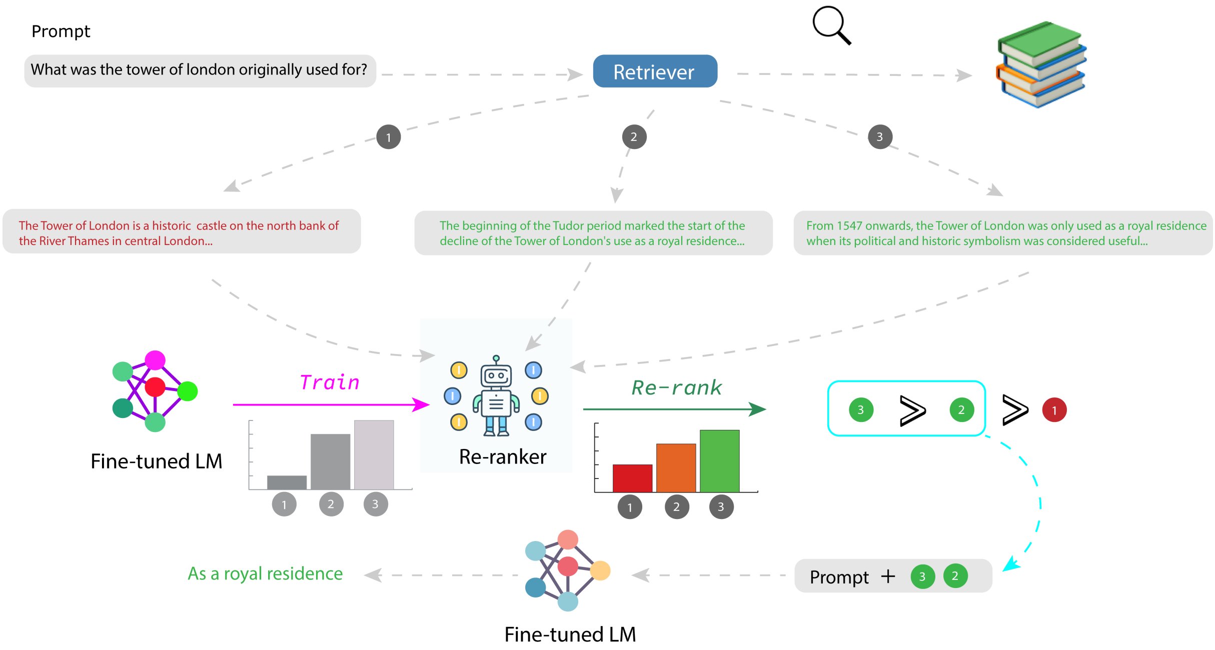
Prompt
What was the tower of london originally used for?
Retriever
1
2
3
The Tower of London is a historic castle on the north bank of the River Thames in central London...
The beginning of the Tudor period marked the start of the decline of the Tower of London's use as a royal residence...
From 1547 onwards, the Tower of London was only used as a royal residence when its political and historic symbolism was considered useful...
Fine-tuned LM
Train
1
2
3
Re-ranker
Re-rank
1
2
3
3 ⩾ 2 ⩾ 1
As a royal residence
Fine-tuned LM
Prompt + 3 2

Figure 1: Overview of RAG-LER. RAG-LER combines the strong performance of LLM on classification and efficiency of cross-encoder, re-ranking retrieved documents. RAG-LER firstly aligns the LLM and improves its capability of context understanding by instruction-tuning, then utilizes fine-tuned LLM to enhance the re-ranker.

with re-ranker, our trained LM can further be used to supervise the re-ranker. We use trained LM to generate the relevance score for the passages, training the re-ranker model with an objective aligns the distribution with LM. This eliminates the manual relevance labelling which requires huge human effort.

Our experiments demonstrate that RAG-LER improves performance on tasks under both open-book and closed-book settings, including Open-domain QA and Commonsense Reasoning. RAG-LER outperforms retrieval-augmented LMs which have larger sizes and fuse more passages. In particular, RAG-LER outperforms RA-DIT (Lin et al., 2023b) on three tasks, Atlas (Izacard et al., 2023) on all tasks. Our analysis shows the effectiveness of our approach across different retrieval methods and LLMs that vary in size and architecture, as well as the importance of each component and training strategy. Our contributions can be summarized as follows:

- We introduce RAG-LER, a framework that enhances the accuracy and factuality of Large Language Models (LLMs) through an innovative LM-enhanced re-ranker.
- We develop a unique approach to train a re-ranker using an LLM as a supervisor. This method eliminates the need for manual human labeling, making the training process more efficient and scalable.
- We demonstrate the consistent performance of our tuned re-ranker across LLMs of varying sizes and architectures, highlighting the versatility and robustness of our approach.

2 Related Work

Retrieval Augmented Generation. Retrieval Augmented Generation (RAG) augments the input with relevant information retrieved from an external knowledge base, improving the performance on various knowledge-intensive tasks (Lewis et al., 2020; Guu et al., 2020). Izacard et al. (2023) pre-trains the retriever and LM in an end-to-end style, followed by fine-tuning in few-shot setting on downstream tasks. Lin et al. (2023b) fine-tunes retriever with instruction-tuned LM. Most prior works focus on retrieving relevant documents once and feeding several of them into the LM. Some recent works present adaptive retrieval, Jiang et al. (2023) uses the upcoming prediction to retrieve relevant documents to regenerate the sentences with low-confidence tokens. Asai et al. (2023) trains LLM using synthetic data to decide whether retrieval is needed and the relevancy of retrieved passages. Although RAG makes great progress, it still has some disadvantages. Not all retrieved passages are useful, retrieval can even hurt the performance (Shi et al., 2023a; Maekawa et al., 2024). Mallen et al. (2022) investigates in what scenario retrieval is necessary for current LLMs. To deal with irrelevant passages, Yoran et al. (2023) trains LMs to be robust with irrelevant passages using mixed data containing relevant and irrelevant passages. Feng et al. (2024) uses multi-LLM collaboration to abstain when lacking the relevant information. Our method trains an LM to understand the input context and learn to abstain when irrelevant passages are given, we also train a re-ranker which re-scores the passages to reduce the length of input context.

Information Retrieval. Information retrieval involves identifying and retrieving information from knowledge resources, which has been used broadly in current NLP tasks. Chen et al. (2017) combines sparse retrieval method with reader component on Open-domain QA. Sparse retrieval represents text using term frequency, making it hard to capture the semantic meaning of the text. Dense retrieval uses dense vectors to embed the meaning of the text, retrieving by calculating the similarity like the inner product of two vectors. Lee et al. (2019) pre-trains the retriever with an unsupervised Inverse Cloze Task (ICT) and trains the retriever and reader jointly. Karpukhin et al. (2020) represents input and passages by a dual-encoder framework. Izacard et al. (2021) trains unsupervised dense retrievers with contrastive learning, showing competitive generality with BM25. Lin et al. (2023a) trains retriever using diverse query and relevance label augmentation, improving on both supervised and zero-shot retrieval. RAG-LER uses the sparse or dense retriever for relevant information retrieval, taking it as an interchangeable component.

Passage Re-ranking. Re-ranking retrieved passages aims to further improve the quality and accuracy of passages. Nogueira and Cho (2019) fine-tunes BERT (Devlin et al., 2019) to re-rank relevant passages, which is the first method using a transformer model for re-ranking tasks. Most works use a cross-encoder model to re-rank in pointwise style, in which documents are scored independently and ranked according to the scores, Ma et al. (2023) introduces a listwise ranking method that directly generates a reordered list. Nogueira et al. (2019) proposes monoBERT and duoBERT that formulate the ranking problem in pointwise and pairwise style respectively. Yet using an encoder-based model has been shown efficient, current works focus on leveraging the strong capability of LLMs for re-ranking tasks. Nogueira et al. (2020) uses a pre-trained sequence-to-sequence model to re-rank passages with generation-based method. Sun et al. (2023) studies the re-ranking capability of decoder-only LMs and proposes a distillation method that distills the passage re-ranking capability of ChatGPT into a smaller model. Our method leverages LLMs' strong capability to supervise the re-ranker training. We do not prompt LLM to generate ranking order, we take the distribution as supervised label, which is a more fine-grained and interpretable signal.

3 RAG-LER

A new framework RAG-LER is introduced, which combines the strong capability of LLM and efficiency of cross-encoder to enhance the LLM, as shown in Figure 1. We fine-tune an LLM to align it and improve its capability of context understanding. The fine-tuned LLM is capable of distinguishing the helpful passage and reflecting its confidence in the given passage, by utilizing this, we let fine-tuned LLM generate relevant labels to further train re-ranker. RAG-LER improves the performance of both LM and re-ranker without sacrificing their original capability.

3.1 Instruction-tuning Language Model

The passages given to the LM are retrieved by semantic similarity when utilizing the dense retriever (Karpukhin et al., 2020), which are not always relevant passages [2] that can help LM respond correctly. Therefore, LM should be robust for irrelevant passages and try to avoid using them. On the contrary, the LM should be capable of utilizing the relevant passages. Alternatively, our method contains a trained re-ranker (see Section 3.2) to help distinguish the relevant and irrelevant passages.

To make LM focus on the given passages, we fine-tune the LM on the datasets of Reading Comprehension task, which trains the LM to answer by utilizing the given passages. Formally, we take the input $(p \circ x)$ and output y, where x and p represent the user input and passages respectively, some items in the datasets offer the irrelevant passages which can not be used to answer the question, instead of answering based on the irrelevant passages or LM's parametric knowledge (Lin et al., 2023b), we add a special token No Answer which LM will

[2] It's important to note that while some passages may be semantically similar to the user input, they may not necessarily contain useful information for answering the query.

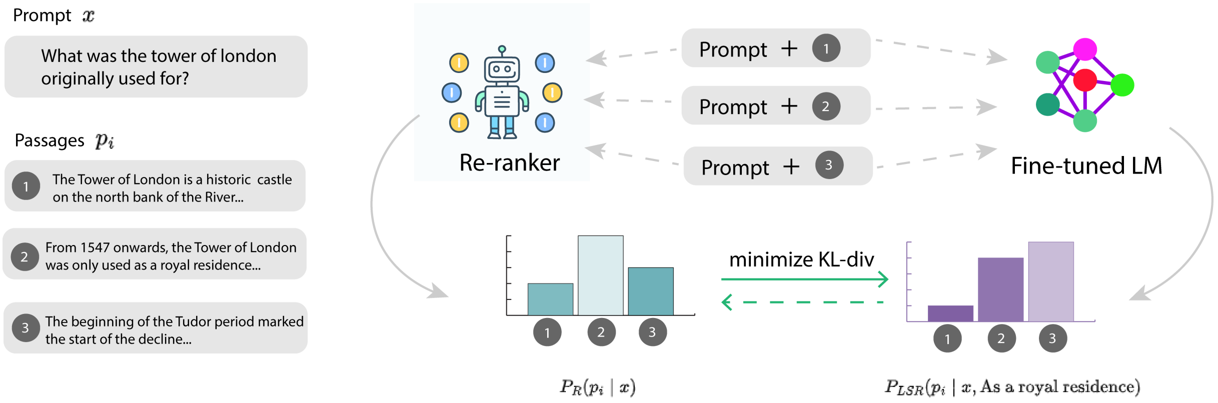
Prompt x
What was the tower of london originally used for?
Passages p_i
1 The Tower of London is a historic castle on the north bank of the River...
2 From 1547 onwards, the Tower of London was only used as a royal residence...
3 The beginning of the Tudor period marked the start of the decline...
Re-ranker
Prompt + 1
Prompt + 2
Prompt + 3
Fine-tuned LM
minimize KL-div
1
2
3
1
2
3
$P_R(p_i \mid x)$
$P_{LSR}(p_i \mid x$, As a royal residence)

Figure 2: RAG-LER re-ranker training process. Given prompt and retrieved passages, re-ranker's distribution approaches LLM's distribution by reducing the KL Divergence.

respond with, we think it will notify users to specify the query more for better retrieval or directly use the LM's parametric knowledge to answer rather than hallucinating. During training, $\mathcal{C}'$ contains the passages $\{p_i \mid i = 1, \cdots, n\}$ where n is the number of passages in the datasets. Additionally, we use instruction-tuning to make LM follow the human instructions. We fine-tune the LM with the standard next token prediction objective:

$$\max \mathbb{E}_{(x,p,y)\in\mathcal{D}_L} \log P_{LM}(y \mid p \circ x) \qquad (1)$$

where $\circ$ denotes the concatenation of sequences. We format our passage as Liu et al. (2023).

Although another method suggested by Longpre et al. (2021) is training LM with substituted entities to make it answer with passages, it will also train the LM to memorize the false facts, in real-world scenarios when retriever doesn't give LM the relevant passages, LM may try to answer based on their parametric knowledge which is factually false. Recent work (Kang et al., 2024) also shows that the unfamiliar data items used for fine-tuning affect how LLMs hallucinate. For these reasons, we choose to make LM respond No Answer instead of errors appearing in fine-tuning data during inference.

3.2 Re-ranker Enhanced by LM

In RAG architecture, LMs are provided with relevant passage $\mathcal{C}'$ retrieved by retriever (sparse or dense), however, these retrieved passages typically are from a large corpus $\mathcal{C}$, which makes retriever struggle to accurately find the relevant passage. Add another re-ranker has been an effective way (Nogueira and Cho, 2019; Xiong et al., 2020; Fang et al., 2024) to offer models more suitable information, which reorders the retrieved candidate passages $\mathcal{C}' \subset \mathcal{C}$ and gives the relevant passages a higher score. Inspired by leveraging the output from the Language Model (LM) as supervised labels to train the retriever(Shi et al., 2023b), we apply these labels to train the intermediate re-ranker instead, which we think it will benefit from: (i) Different from retriever fine-tuning, we separate the LM and the retriever to decrease the conjunction of LM to specific retriever, the middle re-ranker is typically smaller, cost-effective for training, and brings the similar improvements as fine-tuning retriever, fine-tuning the retriever needs to take the information about the entire corpus which is computing intensive. (ii) We don't need to train another retriever from the beginning when we want to use other retrievers. As the upper bound of our re-ranker's capability is dependent on the accuracy of retriever, we can easily leverage other capable retrievers seamlessly. (iii) As the retriever needs to retrieve from a large corpus, the relevant passage may not be given a high-ranking score, which puts the relevant passage in a later position, one typically needs to provide several passages for LM to make sure the provided information is given. It takes the issues of limited context window and lost in the middle (Liu et al., 2023). By utilizing the fine-tuned re-ranker, we give the relevant information a higher score, this reduces the number of passages given to the LM and eliminates the issue of distracting the LM with irrelevant information.

Some passages are relevant as they contain some general information (Fang et al., 2024), as these passages are relevant but not helpful for generating correct answer, our objective is to make middle re-ranker re-rank the relevant passages $\mathcal{C}'$ for the generative LM. As shown in Figure 2, we minimize the KL divergence between the re-rank score

$P_R(p \mid x)$ from the re-ranker:

$$P_R(p \mid x) = \frac{e^{s(x,p)}}{\sum_{p' \in \mathcal{C}'} e^{s(x,p')}} \tag{2}$$

and the LSR (LM-Supervised Reranking) score $P_{LSR}(p \mid x, y)$ from the LM:

$$P_{LSR}(p \mid x, y) = \frac{e^{P_{LM}(y|x,p)/\gamma}}{\sum_{p' \in \mathcal{C}'} e^{P_{LM}(y|x,p')/\gamma}} \tag{3}$$

where γ is a hyperparameter. $\mathcal{C}'$ denotes retrieved passages from corpus $\mathcal{C}$. $s(x, p)$ represents the relevance score assigned by the re-ranker to passage p given input x. Theoretically, the relevance score $s(x, p)$ can be computed as the cosine similarity between the input embedding $\mathbf{E}(x)$ and passage embedding $\mathbf{E}(p)$. In this work, we employ an encoder-based model with a specialized head layer, which is trained to directly generate a relevance score. We compute $P_{LM}(y \mid x, p)$, which represents the language model's confidence in generating output y given input x and passage p. The re-rank score indicates which passage the re-ranker deems most likely to assist the language model in generating the correct answer. The LSR score quantifies the language model's confidence that passage $p \in \mathcal{C}'$ will contribute to generating the correct answer.

Given a training sample (x, y) from the re-ranker fine-tuning dataset $\mathcal{D}_R$, we train the re-ranker by minimizing:

$$\mathcal{L} = \mathbb{E}_{(x,y)\in\mathcal{D}_R} D_{KL}(P_R(p \mid x) \parallel P_{LSR}(p \mid x, y))$$

Train re-ranker with KL Divergence loss helps it align more with the generative LM. We can then take the passages with the top-k highest score as the relevant evidence, significantly reducing the input context length for generative LM and relieving the issue of lost in the middle.

4 Experiments

4.1 Datasets

Training. We train the generative LM and re-ranker respectively. As shown in Table 4, we use datasets $\mathcal{D}_L$ and $\mathcal{D}_R$ to fine-tune the generative LM and re-ranker. For the generative LM training, we mainly focus on improving the model's capability to utilize the knowledge in the passages, we sample instances from Open-Instruct (Ivison et al., 2023), Hugging Face (Lhoest et al., 2021) and NewsQA (Trischler et al., 2016), we download NewsQA from Microsoft [3]. We add instruction-tuning data to make LM better follow the instructions. For the re-ranker training, we focus on making re-ranker distinguish between the passages that can help answer the question and the passages that can't. We sample items from Natural Questions (Kwiatkowski et al., 2019) and HotpotQA (Yang et al., 2018) [4]. To train re-ranker improving the score of relevant passages in a real way, we first use a retriever to retrieve 30 passage candidates, then strip the items containing 0 and 30 relevant passages, which results in 41k items (see Appendix A.1 for data details).

Evidence Corpus $\mathcal{C}$. During training, we use Dec 2018 Wikipedia dump released by Karpukhin et al. (2020) [5] as evidence corpus. For the rest of experiments, we use Dec 2021 Wikipedia dump released by Izacard et al. (2023) as our evidence corpus which contains 33M passages, each with fewer than 200 words.

4.2 Baselines

We compare our model to the base Llama2 models (Touvron et al., 2023b) in the 5-shot In-Context Learning setting, combined with re-ranker not trained on our $\mathcal{D}_R$, and state-of-the-art Retrieval-Augmented LMs including Atlas (Izacard et al., 2023) which jointly pre-trains encoder-decoder based LM and retriever, then fine-tunes with 64-shot downstream examples. RA-DIT (Lin et al., 2023b) uses decoder-only based Llama 65B (Touvron et al., 2023a) as its base model, combined with a retriever supervised by LM, it's similar to our method. However, it aims to improve the capacity of retriever by fine-tuned LM.

4.3 Settings

Training. We use Contriever-MS MARCO (Izacard et al., 2021) to retrieve top-30 passage candidates in our training data of re-ranker. For re-ranker, we use ms-marco-MiniLM-L-6-v2 [6] from Sentence Transformer (Reimers and Gurevych, 2019) as our re-ranker base, which is a distilled BERT model

[3] https://www.microsoft.com/en-us/research/project/newsqa-dataset

[4] We use HotpotQA dataset for both LM and re-ranker training, we sample items and there is no overlap of both sampled data.

[5] From observation of our early experiments and prior works, the 2018 Wikipedia dump works better for our re-ranker training.

[6] https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-6-v2

	ARC-C	OBQA	BoolQ	PIQA	WinoGrande	CSQA
Model	(acc)	(acc)	(acc)	(acc)	(acc)	(acc)
Llama2$_{7B}$	47.3	52.6	57.7	54.3	52.2	52.7
Llama2$_{13B}$	65.2	63.2	66.3	60.2	54.0	65.8
RAG-LER$_{7B}$	64.9	64.0	66.9	**74.8**	52.6	67.9
RAG-LER$_{13B}$	**72.6**	**69.8**	**78.0**	74.3	**54.9**	**72.8**

Table 1: Results on Commonsense Reasoning tasks without retrieval. We evaluate on dev splits of the datasets.

	NQ	TQA	HoPo	FEV
Model	(em)	(em)	(em)	(acc)
Llama2$_{7B}$	37.9	72.6	28.3	82.2
Llama2$_{13B}$	41.0	75.8	32.6	84.0
Atlas	42.2	74.5	34.7	87.1
RAG-LER$_{7B}$	41.8	77.6	34.0	64.7
RAG-LER$_{13B}$	**42.8**	**79.4**	**36.9**	**88.7**
RA-DIT$_{65B}$	35.2	75.4	**39.7**	80.7
RAG-LER$_{7B}$	42.0	78.2	34.9	65.3
RAG-LER$_{13B}$	**43.2**	**80.1**	37.3	**88.8**

Table 2: Results on KILT test sets of Open-domain QA and Fact Checking. em, acc denote exact match, accuracy respectively. Llama2 models are evaluated by using 5-shot In-Context Learning.

with 6 layers for efficiency, and further fine-tuned on MS-MARCO dataset (Campos et al., 2016) to improve the capability of passage retrieval. We take Llama2 7B and 13B models (Touvron et al., 2023b) as our base generative model. We set hyperparameter γ to 0.01 when getting LSR score.

Inference. For most of our experiments, we take off-the-shelf Contriever-MS MARCO as our retriever model, To compare with RA-DIT, we use Dragon+ (Lin et al., 2023a) as our retriever. We retrieve 30 passages for all retrieval-needed experiments. We take greedy search as our default generation strategy. We evaluate our method on both open-book and closed-book settings, we evaluate Llama2 models using 5-shot In-Context Learning, we sample examples from corresponding training split, using retrieved top-1 relevant passage for sampled items in the open-book setting.

5 Results

We report our results on Open-domain QA and Fact Checking benchmark from KILT (Petroni et al., 2020b) in Table 2. RAG-LER improves performance significantly on both 7B (from 6.9% to 20.1% on most of the datasets) and 13B (from 4.4% to 13.2% on all datasets) models, however, RAG-LER$_{7B}$ lags behind Llama2$_{7B}$ on FEVER (Thorne et al., 2018), we find that RAG-LER$_{7B}$ tends to explain the fact and rectify the statement, which is much easier for Llama2 to generate the formally correct label with In-Context Learning. In the meantime, our instruction-tuned model decreases the inference time and cost compared to the original Llama2 model. Our RAG-LER outperforms Atlas on all datasets, Atlas fine-tunes on each dataset and evaluates with task-specific models, while we evaluate on all datasets with the same model. RAG-LER also achieves better performance on most benchmarks compared to RA-DIT, but still performs worse on HotpotQA which requires multi-hop reasoning, it's likely due to that larger models with more parameters could better capture and process contextual information simultaneously (Wei et al., 2022).

In addition to knowledge-intensive tasks, we also evaluate our method on Commonsense Reasoning tasks with closed-book settings, including ARC-Challenge (Clark et al., 2018), OpenBookQA (Mihaylov et al., 2018), BoolQ (Clark et al., 2019), PIQA (Bisk et al., 2019), WinoGrande (Sakaguchi et al., 2019) and CommonsenseQA (Talmor et al., 2019). Table 1 presents our results compared to Llama2 models. RAG-LER outperforms both 7B and 13B models on all evaluation datasets, which demonstrates our models' strong capability on tasks with and without retrieval. Despite training our models with external passages, RAG-LER still shows good performance without it, we think it could be largely attributed to the benefit of fine-tuning with instruction (Chung et al., 2022), which improves the LM's ability to understand inputs.

6 Analysis

We conduct analysis studies based on our 7B model to get insights into our framework.

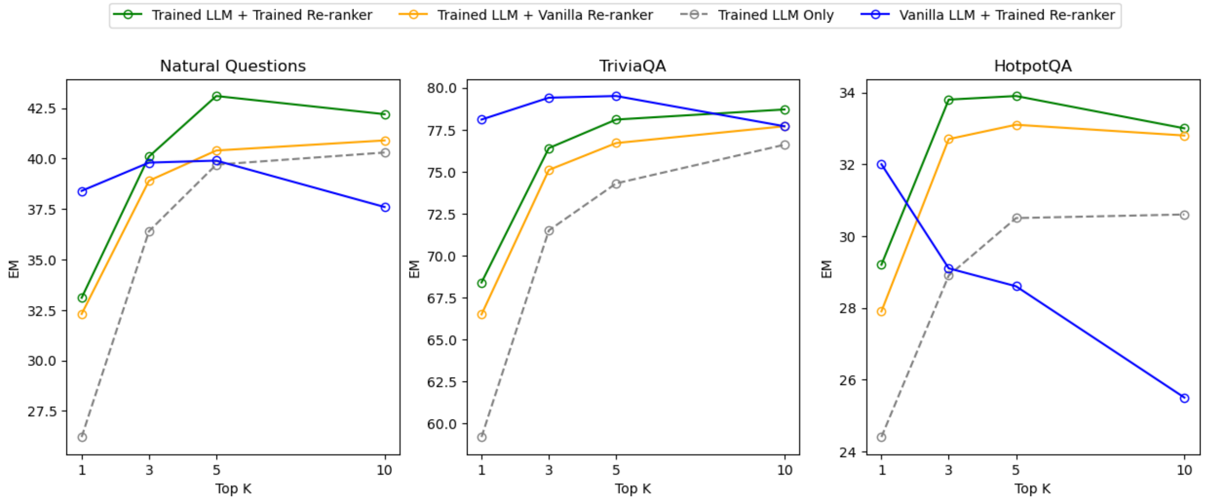

Figure 3: Ablation studies. We evaluate on dev splits of NQ, TQA, and HotpotQA in different settings. We take Llama2$_{7B}$ as our vanilla LLM, and ms-marco-MiniLM-L-6-v2 as our vanilla re-ranker. Llama2$_{7B}$ is evaluated with 5-shot In-Context Learning. All results use Contriever-MS MARCO as retriever.

6.1 Ablation Studies

We conduct a set of ablation experiments to study the effects of each component and training strategy of our framework. We conduct ablation studies on Natural Questions, TriviaQA and HotpotQA. Figure 3 illustrates the results.

Instruction-tuning LM. We study the effects of fine-tuning our generative LM with instruction, which is used to generate the training data for re-ranker. We evaluate pretrained Llama2$_{7B}$ model and RAG-LER$_{7B}$. We conduct the same training strategy for re-ranker combined with Llama2 model. We use 1-shot In-Context Learning to get re-ranker training probability [7] under the same conditions of RAG-LER. We evaluate both models with different numbers of retrieved chunks, as shown in Figure 3. Pretrained Llama2 model performs better with top-1 chunk, with more added chunks, RAG-LER gains more improvements. In the meantime, additional retrieved chunks even harm the capability of Llama2 model and degrade it monotonically, especially on multi-hop reasoning task which needs to combine multiple knowledge sources to answer the question. This aligns with the observation in prior works (Petroni et al., 2020a; Li et al., 2022; Maekawa et al., 2024). On the contrary, RAG-LER shows increasing performance with more chunks, it still degrades when given more than 5 chunks, but with less performance loss compared to pretrained Llama2 model, which demonstrates the RAG-LER's capability to reject the irrelevant passages.

Re-ranker supervised by LM. We conduct the analysis of how training re-ranker supervised by LM affects the performance of our model. We evaluate the model's performance with original ms-marco-MiniLM-L-6-v2 and further trained correspondent. As shown in Figure 3, combining with LM supervised re-ranker improves the performance on all cases with different numbers of chunks. Furthermore, with our trained re-ranker, model performs better with fewer chunks, which indicates that re-ranker successfully ranks up the helpful passages compared to the original.

6.2 Performance among Different Retrievers

As our framework focuses on improving the capabilities of generative LM and re-ranker, we hypothesize that greater retriever leads to better performance, and the ability should maintain consistency among different retrievers. We evaluate whether the performance changes when the retriever changes on Natural Questions, TriviaQA, and HotpotQA with our RAG-LER$_{7B}$. We take both sparse retriever (BM25)[8] and dense retriever (Contriever-MS MARCO and Dragon+). As shown in Figure 4, the variety of retrievers doesn't affect the relative effectiveness among different settings, using LM supervised re-ranker improves the performance with different retrievers. We use the performance without re-ranker as the norm of retriever

[7] Considering the inference cost, we use 1-shot ICL.

[8] We use Pyserini (Lin et al., 2021) for our BM25 retrieval.

	NQ			HoPo		
Trained by	Llama2$_{7B}$	Llama2$_{13B}$	Mistral$_{7B}$	Llama2$_{7B}$	Llama2$_{13B}$	Mistral$_{7B}$
No training	40.4	42.2	42.9	33.1	35.8	34.4
Llama2$_{7B}$	43.1	43.7	44.3	33.9	36.6	35.4
Llama2$_{13B}$	42.4	43.6	44.0	34.0	36.4	35.6
Mistral$_{7B}$	41.6	43.1	43.7	33.7	36.4	35.9

Table 3: Performance of LLMs with different re-rankers on NQ and HotpotQA. The models are base models used for instruction-tuning. Each column corresponds to the performance of an LLM, each row corresponds to the performance of a re-ranker.

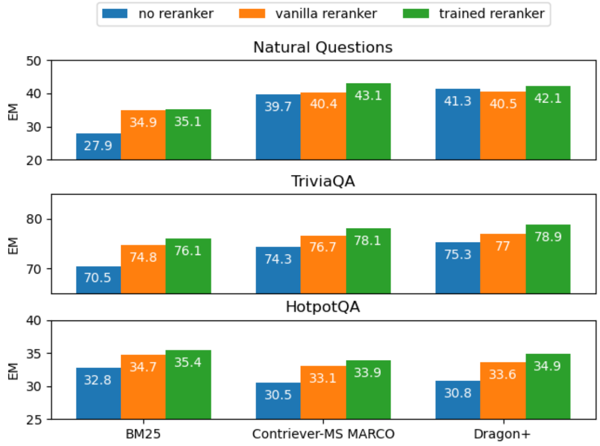

Figure 4: Performance across different retrievers based on our 7B model. Training re-ranker consistently benefits both Sparse and Dense retrievers.

performance, we can observe a better retriever combined with either untrained or LM-supervised re-ranker generally leads to better performance. Despite this, it shows worse on Natural Questions using Dragon+ combined with re-ranker, which means that re-ranker doesn't necessarily improve the performance in some cases, even with downgrades of untrained re-ranker, however, training with our method still gains improvement.

6.3 Transferability of Re-ranker

We further study whether our re-ranker fine-tuned by an LLM can be utilized by other LLMs. We evaluate on NQ and HotpotQA with LLMs that vary in size and architecture. More specifically, we evaluate on instruction fine-tuned Llama2$_{7B}$, Llama2$_{13B}$ and Mistral$_{7B}$[9], we keep the same configurations of training Mistral$_{7B}$ as two Llama2 models which are used as our RAG-LER base models. Table 3 shows the performance of different models combined with different re-rankers. The re-rankers supervised by different LLMs consistently improve the LLMs' performance on both tasks, it indicates that a re-ranker trained by an LLM can be used by other LLMs without repeating the process again. Interestingly, the model's performance doesn't necessarily link to re-ranker performance, re-ranker supervised by Llama2$_{7B}$ performs slightly higher on NQ, but for Mistral$_{7B}$, re-ranker supervised by Llama2$_{13B}$ performs better than its smaller respondent on HotpotQA, and also performs best with re-ranker supervised by itself. In addition, we observe that the performance improves when LLMs' performance improves across different trained re-rankers. These observations may also show us an effective way to improve the performance by training a re-ranker with a relatively smaller model, and transplanting it to a larger or more robust model.

7 Conclusion

In this work, we introduce RAG-LER, a novel framework that enhances the accuracy and factuality of LLMs through an LM-enhanced re-ranker. RAG-LER fine-tunes pre-trained LMs to discriminately utilize provided information and generate supervised labels for re-ranker training without human intervention. Our experiments demonstrate that RAG-LER consistently outperforms existing retrieval-augmented LMs across knowledge-intensive tasks while reducing the context length. Notably, we show the consistent performance of the tuned re-ranker across different LLMs, varying in size and architecture.

Limitations

Our work primarily enhances the second-stage re-ranking process. However, the overall performance is still constrained by the upstream retriever's effectiveness. Future work could explore joint optimization of the retriever and re-ranker to further improve the quality of retrieved passages. An intriguing

[9] https://huggingface.co/mistralai/Mistral-7B-v0.3

finding from our study indicates that the LLM used to generate training labels for the re-ranker doesn't consistently collaborate optimally with its trained re-ranker across all scenarios. This unexpected behavior warrants further investigation. Our current approach for re-ranker training involves presenting single passages to the LLM for relevance scoring. While effective for many tasks, this method may not be optimal for complex queries, particularly multi-hop QA tasks that require information synthesis from multiple sources. It's worth exploring methods for assessing the relevance of multiple passages simultaneously.

Ethics Statement

In the development and presentation of this work, we have carefully considered the ethical implications of our work. This work aims to enhance the accuracy of LLMs and reduce hallucination, potentially decreasing the spread of misinformation. However, it can still generate content that is not based on given non-parametric knowledge or that is misinformed. As we hypothesize the factuality of external knowledge store, it's not always the case in real-world scenarios, one can generate fake or harmful content with a modified knowledge store. We are committed to transparency in our research. Our methodology, including the specifics of the RAG-LER framework, code and model will be made available to the research community.

References

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

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2019. Piqa: Reasoning about physical commonsense in natural language. In *AAAI Conference on Artificial Intelligence*.

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, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Ma teusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *ArXiv*, abs/2005.14165.

Daniel Fernando Campos, Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, Li Deng, and Bhaskar Mitra. 2016. Ms marco: A human generated machine reading comprehension dataset. *ArXiv*, abs/1611.09268.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. *ArXiv*, abs/1704.00051.

Hyung Won Chung, Le Hou, S. Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Wei Yu, Vincent Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed Huai hsin Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *ArXiv*, abs/2210.11416.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *NAACL*.

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

Tri Dao. 2023. Flashattention-2: Faster attention with better parallelism and work partitioning. *ArXiv*, abs/2307.08691.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *ArXiv*, abs/2305.14314.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*.

Wei Fang, Yung-Sung Chuang, and James Glass. 2024. Joint inference of retrieval and generation for passage re-ranking. In *Findings*.

Shangbin Feng, Weijia Shi, Yike Wang, Wenxuan Ding, Vidhisha Balachandran, and Yulia Tsvetkov. 2024. Don't hallucinate, abstain: Identifying llm knowledge gaps via multi-llm collaboration. *ArXiv*, abs/2402.00367.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. *ArXiv*, abs/2002.08909.

Hamish Ivison, Yizhong Wang, Valentina Pyatkin, Nathan Lambert, Matthew E. Peters, Pradeep Dasigi, Joel Jang, David Wadden, Noah A. Smith, Iz Beltagy, and Hanna Hajishirzi. 2023. Camels in a changing climate: Enhancing lm adaptation with tulu 2. *ArXiv*, abs/2311.10702.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *Trans. Mach. Learn. Res.*, 2022.

Gautier Izacard and Edouard Grave. 2020. Leveraging passage retrieval with generative models for open domain question answering. *ArXiv*, abs/2007.01282.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, 24(251):1–43.

Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. *ArXiv*, abs/2305.06983.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Katie Kang, Eric Wallace, Claire Tomlin, Aviral Kumar, and Sergey Levine. 2024. Unfamiliar finetuning examples control how language models hallucinate. *ArXiv*, abs/2403.05612.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Yu Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. 2020. Dense passage retrieval for open-domain question answering. *ArXiv*, abs/2004.04906.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. 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 V. Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard H. Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. *ArXiv*, abs/1704.04683.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. *ArXiv*, abs/1906.00300.

Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Kuttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *ArXiv*, abs/2005.11401.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario vSavsko, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clement Delangue, Th'eo Matussiere, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander M. Rush, and Thomas Wolf. 2021. Datasets: A community library for natural language processing. *ArXiv*, abs/2109.02846.

Daliang Li, Ankit Singh Rawat, Manzil Zaheer, Xin Wang, Michal Lukasik, Andreas Veit, Felix X. Yu, and Surinder Kumar. 2022. Large language models with controllable working memory. *ArXiv*, abs/2211.05110.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: A Python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021)*, pages 2356–2362.

Sheng-Chieh Lin, Akari Asai, Minghan Li, Barlas Oğuz, Jimmy J. Lin, Yashar Mehdad, Wen tau Yih, and Xilun Chen. 2023a. How to train your dragon: Diverse augmentation towards generalizable dense retrieval. *ArXiv*, abs/2302.07452.

Xi Victoria Lin, Xilun Chen, Mingda Chen, Weijia Shi, Maria Lomeli, Rich James, Pedro Rodriguez, Jacob Kahn, Gergely Szilvasy, Mike Lewis, Luke Zettlemoyer, and Scott Yih. 2023b. Ra-dit: Retrieval-augmented dual instruction tuning. *ArXiv*, abs/2310.01352.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

S. Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. 2021. Entity-based knowledge conflicts in question answering. *ArXiv*, abs/2109.05052.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. 2023. The flan collection: Designing data and methods for effective instruction tuning. *arXiv preprint arXiv:2301.13688*.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *ArXiv*, abs/1711.05101.

Xueguang Ma, Xinyu Crystina Zhang, Ronak Pradeep, and Jimmy J. Lin. 2023. Zero-shot listwise document reranking with a large language model. *ArXiv*, abs/2305.02156.

Seiji Maekawa, Hayate Iso, Sairam Gurajada, and Nikita Bhutani. 2024. Retrieval helps or hurts? a deeper dive into the efficacy of retrieval augmentation to language models. *ArXiv*, abs/2402.13492.

Alex Troy Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Hannaneh Hajishirzi, and Daniel Khashabi. 2022. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Annual Meeting of the Association for Computational Linguistics*.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Conference on Empirical Methods in Natural Language Processing*.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with bert. *ArXiv*, abs/1901.04085.

Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy J. Lin. 2020. Document ranking with a pretrained sequence-to-sequence model. In *Findings*.

Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy J. Lin. 2019. Multi-stage document ranking with bert. *ArXiv*, abs/1910.14424.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*.

Fabio Petroni, Patrick Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2020a. How context affects language models' factual predictions. *ArXiv*, abs/2005.04611.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vassilis Plachouras, Tim Rocktaschel, and Sebastian Riedel. 2020b. Kilt: a benchmark for knowledge intensive language tasks. In *North American Chapter of the Association for Computational Linguistics*.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2019. Zero: Memory optimizations toward training trillion parameter models. *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. *ArXiv*, abs/1806.03822.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Winogrande. *Communications of the ACM*, 64:99 – 106.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed Huai hsin Chi, Nathanael Scharli, and Denny Zhou. 2023a. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*.

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

Weiwei Sun, Lingyong Yan, Xinyu Ma, Pengjie Ren, Dawei Yin, and Zhaochun Ren. 2023. Is chatgpt good at search? investigating large language models as re-ranking agent. *ArXiv*, abs/2304.09542.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *ArXiv*, abs/1811.00937.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. Fever: a large-scale dataset for fact extraction and verification. *ArXiv*, abs/1803.05355.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *ArXiv*, abs/2302.13971.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2016. Newsqa: A machine comprehension dataset. In *Rep4NLP@ACL*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed Huai hsin Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models. *ArXiv*, abs/2206.07682.

Wenhan Xiong, Xiang Lorraine Li, Srini Iyer, Jingfei Du, Patrick Lewis, William Yang Wang, Yashar Mehdad, Wen tau Yih, Sebastian Riedel, Douwe Kiela, and Barlas Oğuz. 2020. Answering complex open-domain questions with multi-hop dense retrieval. *ArXiv*, abs/2009.12756.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing*.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2023. Making retrieval-augmented language models robust to irrelevant context. *ArXiv*, abs/2310.01558.

Wenxuan Zhou, Sheng Zhang, Hoifung Poon, and Muhao Chen. 2023. Context-faithful prompting for large language models. In *Conference on Empirical Methods in Natural Language Processing*.

A Experimental Details

A.1 More details of training data

Details of LM training data. To improve the capability of Instruction following, we sample instances from Open-Instruct (Ivison et al., 2023) dataset. Particularly, we take items from their GPT-4 Alpaca (Peng et al., 2023), FLAN-V2, the CoT subset of the FLAN-V2 mixture (Longpre et al., 2023), ShareGPT. For Reading Comprehension, we sample instances from a couple of QA datasets including HotpotQA (Yang et al., 2018), SQuAD-V2 (Rajpurkar et al., 2018), RACE (Lai et al., 2017), NewsQA (Trischler et al., 2016). As shown in Table 4, we take 11643 instances in total. We use the golden evidence chunks in each Reading Comprehension data item. For most QA datasets, there is only 1 golden evidence chunk for each item, for HotpotQA, we use 2 golden evidence chunks for each item, some items in SQuAD contain passages that can't be used to answer the question, we replace the output with special token No Answer.

Details of re-ranker training data. We sample items from Natural Questions and HotpotQA datasets (no overlap with LM training data), As the re-ranker is bundled with the LM, we infer trained 7B model to get P_{LSR} in advance for re-ranker training on 1 Nvidia H100 with 80GB memory, for 13B model, we use 2 Nvidia H100 with 80GB memory. We use top-30 retrieved passages for each data item, we set Contriever-MS MARCO (Izacard et al., 2021) as our default retriever and get all retrieved passages for re-ranker training. We set hyperparameter γ to 0.01. We use FAISS-GPU (Johnson et al., 2019) for fast similarity search.

A.2 Fine-tuning details

LM fine-tuning. All generative models are basically trained under the same configuration on 2 Nvidia RTX 3090 with 24GB memory each. we fine-tune our models using QLoRA (Dettmers et al., 2023), the model is quantized to 4-bit during training, we take lora rank of 64, alpha of 16 and dropout of 0.1. We use linear learning rate scheduler with peak learning rate of 1e-4 and warmup ratio of 3% and AdamW (Loshchilov and Hutter, 2017) with weight decay of 0. We train our generative LMs for 3 epochs with a batch size of 128. We use FlashAttention-2 (Dao, 2023) for more efficient training.

Re-ranker fine-tuning. Our re-rankers are trained on the same hardware configuration as LM training. We do not use PEFT (Parameter-Efficient Fine-Tuning) method to fine-tune our re-rankers, which means we fine-tune with full parameters. We use linear learning rate scheduler with peak learning rate of 2e-5 and warmup ratio of 3%, and AdamW optimizer with weight decay of 0. We train re-ranking models for 2 epochs with a batch size of 128. We use DeepSpeed ZeRO stage 3 (Rajbhandari et al., 2019) for efficient training.

A.3 Inference details

For BoolQ, Our instruction-tuned models demonstrated a tendency to generate "yes" or "no" responses rather than the original "true" or "false" labels. To accommodate this, we adjusted the labels, mapping "true" to "yes" and "false" to "no". For FEVER, We replaced the original "REFUTES" and "SUPPORTS" labels with "false" and "true", respectively. During RAG-LER inference, we set the maximum length of newly generated tokens to 100 for all evaluations. We use the same instruction for most open-domain QA tasks. For pre-trained LLMs which use In-Context Learning, we set the maximum length of new tokens to 30 for Open-domain QA, and 10 for FEVER. For most of reasoning tasks, we unified the format to choice selection, limiting the maximum length to 10 tokens.

Dataset name	Task	Data source	Number of instances	Category
GPT-4 Alpaca	Instruction-following	Open-Instruct	22368	$\mathcal{D}_L$
FLAN-V2	Instruction-following	Open-Instruct	15316	$\mathcal{D}_L$
ShareGPT	Instruction-following	Open-Instruct	12567	$\mathcal{D}_L$
FLAN-V2-CoT	Instruction-following	Open-Instruct	9948	$\mathcal{D}_L$
HotpotQA $_{\mathcal{D}_L}$	Reading Comprehension	Hugging Face	17772	$\mathcal{D}_L$
SQuAD-V2	Reading Comprehension	Hugging Face	10326	$\mathcal{D}_L$
RACE	Reading Comprehension	Hugging Face	8256	$\mathcal{D}_L$
NewsQA	Reading Comprehension	NewsQA	15090	$\mathcal{D}_L$
HotpotQA $_{\mathcal{D}_R}$	Passage Re-ranking	Hugging Face	20895	$\mathcal{D}_R$
Natural Questions	Passage Re-ranking	KILT	19828	$\mathcal{D}_R$

Table 4: The statistics of LM and re-ranker training data. HotpotQA is used to train both LM and re-ranker, we split it into 2 non-overlap parts, and then sample items.

B Examples

We show the examples of our RAG-LER$_{\text{7B}}$ on open-domain QA tasks. As shown in Table 5 and Table 6. Due to the context length, we show examples with top-3 passages. For each example, we provide the corresponding instruction, the original question, the top-3 re-ranked passages, and RAG-LER$_{\text{7B}}$'s response. When the retrieved passages contain relevant information, RAG-LER synthesizes this information to provide accurate and comprehensive answers. For passages that don't contain useful information, RAG-LER is designed to generate a special token rather than producing hallucinated content. Table 7 shows the example of instruction-tuned Llama2 model combined with different re-rankers. We use top-2 passages for this example.

Instruction: Answer the following question based on the provided contexts. You may use one or more provided contexts.

Context:
Document 1: (Title: Got My Mind Set on You) "Got My Mind Set on You" (also written as "(Got My Mind) Set on You") is a song written and composed by Rudy Clark and originally recorded by James Ray in 1962, under the title "I've Got My Mind Set on You". An edited version of the song was released later in the year as a single on the Dynamic Sound label credited to James Ray with Hutch Davie Orchestra & Chorus. In 1987, George Harrison released a cover version of the song as a single, and released it on his album Cloud Nine, which he had recorded on his own Dark Horse Records label.
Document 2: (Title: Got My Mind Set on You) George Harrison – vocals, guitar ; Jeff Lynne – bass, keyboard ; Jim Keltner – drums ; Jim Horn – saxophone ; Ray Cooper – percussion This is the personnel as listed.
Document 3: (Title: Cloud Nine (George Harrison album)) Harrison's cover of Rudy Clark's little-known song "Got My Mind Set on You" quickly reached number 1 in the United States and number 2 in the United Kingdom. It was Harrison's first single to top the US charts since "Give Me Love (Give Me Peace on Earth)" in 1973. A few weeks later, Cloud Nine was released to high anticipation and a favourable critical reception. The album went to number 10 in the UK. In the US, it peaked at number 8 on the Billboard 200 chart and number 4 on the Cash Box Top 200, and achieved platinum status. It was also number 1 on Cash Boxs Top 40 Compact Discs chart. The success of the single "Got My Mind Set on You" and its accompanying video re-introduced Harrison to the mainstream.
Question: Who sings got my mind set on you?
Response: George Harrison

Context:
Document 1: (Title: It Happened at the World's Fair) It Happened at the World's Fair is a 1963 American musical film starring Elvis Presley as a crop-dusting pilot. It was filmed in Seattle, Washington, site of the Century 21 Exposition, along with the Seattle World's Fair. The governor of Washington at the time, Albert Rosellini, suggested the setting to Metro-Goldwyn-Mayer executives. The film made $2.25 million at the box office.
Document 2: (Title: It Happened at the World's Fair) The Seattle Center, including the Seattle Center Monorail and the Space Needle, serve as backdrops for several scenes. Security officers pursue Presley and the girl through the fountains at what is now the Pacific Science Center. The hitchhiking scene with Elvis and Gary Lockwood was filmed near Camarillo, California, as were some of the flying scenes. The entire hitchhiking scene, up to the point when Mike and Danny are picked up, was filmed on 5th Street near Pleasant Valley Road on the south side of Camarillo. While The Elvis Encyclopedia believes that the Wilburton Trestle was shown in the film, further evidence points to a different location.
Document 3: (Title: It Happened at the World's Fair (soundtrack)) It Happened at the World's Fair is the sixth soundtrack album by American singer and musician Elvis Presley, released by RCA Victor in mono and stereo, LPMLSP 2697, in April 1963. It is the soundtrack to the 1963 film of the same name starring Presley. Recording sessions took place at Radio Recorders in Hollywood on August 30 and September 22, 1962. It peaked at number four on the Billboard Top Pop Albums chart.
Question: Where was it happened at the world fair filmed?
Response: Seattle, Washington

Table 5: Examples of open-domain QA.

Instruction: Answer the following question based on the provided contexts. You may use one or more provided contexts.

Context:
Document 1: (Title: Insurance) Claims and loss handling is the materialized utility of insurance; it is the actual "product" paid for. Claims may be filed by insureds directly with the insurer or through brokers or agents. The insurer may require that the claim be filed on its own proprietary forms, or may accept claims on a standard industry form, such as those produced by ACORD. Insurance company claims departments employ a large number of claims adjusters supported by a staff of records management and data entry clerks. Incoming claims are classified based on severity and are assigned to adjusters whose settlement authority varies with their knowledge and experience.
Document 2: (Title: QC clause) commonly provide that the insurer will pay the legal costs of the defence. It is sometimes suggested that the clause is also designed to protect the public from the unjustified contesting of claims which have no real defence, but this may represent a charitable view of the way insurance companies conduct claims. Conflicts between the insurer and insured can put a solicitor who represents both in an invidious position, particularly where the solicitor has received confidential information from one party, and may result in multiplicity of legal representation. In practice however, there is an enormous reluctance to invoke such clauses, partly because of the expense of instructing Queen's Counsel.
Document 3: (Title: Insurance) beneficial to it and the insured in the claim handling process. An entity seeking to transfer risk (an individual, corporation, or association of any type, etc.) becomes the "insured" party once risk is assumed by an "insurer", the insuring party, by means of a contract, called an insurance policy. Generally, an insurance contract includes, at a minimum, the following elements: identification of participating parties (the insurer, the insured, the beneficiaries), the premium, the period of coverage, the particular loss event covered, the amount of coverage (i.e., the amount to be paid to the insured or beneficiary in the event of a loss), and exclusions (events not covered).
Question: Who represents the insured in the investigation of a claim?
Response: [No Answer]

Table 6: Examples of open-domain QA where passages do not contain useful information.

Passages	Has answer	Response
Prompt: Where does arsenic and old lace take place?		
Golden label: Brooklyn, New York		
Document 1: (Title: Arsenic and Old Lace (play)) The play is a farcical black comedy revolving around the Brewster family, descended from the Mayflower settlers but now composed of maniacs, most of them homicidal. The hero, Mortimer Brewster, is a drama critic who must deal with his crazy, murderous family and local police in Brooklyn, New York, as he debates whether to go through with his recent promise to marry the woman he loves, Elaine Harper, who lives next door and is the daughter of the local minister. His family includes two spinster aunts who have taken to murdering lonely old men by poisoning them with a glass of home-made elderberry wine laced with arsenic, strychnine, and "just a pinch" of cyanide;	✓	
Document 2: (Title: Arsenic and Old Lace (cocktail)) Arsenic and Old Lace (also called the Attention Cocktail or the Atty) is a classic cocktail with its origins in the 1910's made with gin, crème de violette, dry vermouth and absinthe. The first appearance of a cocktail with these four parts, albeit in equal quantities, was in Hugo Ensslin's Recipes for Mixed Drinks published in 1917, called the "Attention Cocktail". The 1930 edition of The Savoy Cocktail Book, a drink with those four ingredients, rebranded as the "Atty Cocktail" had ratios that more closely matched the modern Arsenic and Old Lace. "The Atty" first appears under the name "Arsenic and Old Lace" in 1941, published in the Cocktail Guide and Ladies' Companion by former Broadway producer Crosby Gaige. Around the same time, Joseph Kesselring's play Arsenic and Old Lace opened on Broadway in January 1941. The timing strongly implies a connection, though it is speculative to say whether Gaige was the one who renamed the cocktail.	✗	Brooklyn, New York
Document 1: (Title: Windsor, Connecticut) The Joseph Kesselring stage play and Frank Capra movie Arsenic and Old Lace was inspired by actual events that took place in a three-story brick house on Prospect Street, just off the north end of the Windsor green. Sixty men died between 1907 and 1917 while in the care of Amy Archer-Gilligan. Most were proven to be victims of arsenic poisoning. On historic Palisado Avenue, one can find the First Church in Windsor, Congregational, and adjacent graveyard. Across the street on the Palisado Green stands a statue of John Mason, a founder of Windsor and a colonial leader in the Pequot War. The historic plaque also honors Robert Seeley, Mason's second-in-command. Nearby stands Windsor's oldest structure, the Capt.	✗	
Document 2: (Title: Arsenic and Old Lace (play)) Arsenic and Old Lace is a play by American playwright Joseph Kesselring, written in 1939. It has become best known through the subsequent film adaptation starring Cary Grant and directed by Frank Capra. The play was produced by Lindsay and Crouse and directed by Bretaigne Windust, and opened on Broadway at the Fulton Theatre on January 10, 1941. On September 25, 1943, the play moved to the Hudson Theatre, closing there on June 17, 1944, having played 1,444 performances. The West End production - directed by Marcel Varnel and produced at London's Strand Theatre - enjoyed a similarly long run. Opening on December 23, 1942 and closing on March 2, 1946, it totalled 1,337 performances. Of the 12 plays written by Kesselring, Arsenic and Old Lace was much the most successful. According to the opening night review in The New York Times, the play was "so funny that none of us will ever forget it."	✗	Windsor, Connecticut
Document 1: (Title: Arsenic and Old Lace (play)) 'Arsenic and Old Lace' with Betty Garrett, Carole Cook & Michael Lee Stever.jpgArsenic and Old Lace resurfaced in 1998 with an Oklahoma City based revival from UCO Theatricals University Of Central Oklahoma starring Betty Garrett, Carole Cook and Michael Stever. ; A recent revival was mounted in February 2011 at the Dallas Theater Center starring Betty Buckley and Tovah Feldshuh. ; A Hebrew version was staged at the Habima Theatre in Tel Aviv with the opening night on October 29, 2012, with Lea Koenig and Dvora Kaydar in the main roles. ; On November 19, 2016, Independent Theatre Pakistan opened their new season with a rendition of the performance at Ali Auditorium in Lahore, Pakistan. ; The Maitland Repertory Theatre in Maitland, Australia, put on a production of the play from October 9th to 27th 2019 by director Colin Delane.	✗	
Document 2: (Title: Arsenic and Old Lace (film)) Arsenic and Old Lace is a 1944 American black comedy film directed by Frank Capra and starring Cary Grant. It was based on Joseph Kesselring's 1941 play, Arsenic and Old Lace. The script adaptation was written by Julius J. Epstein and Philip G. Epstein. The contract with the play's producers stipulated that the film would not be released until the Broadway run ended. The original planned release date was September 30, 1942. The play was a tremendous hit, running for three and a half years, so the film was not released until 1944.	✗	Oklahoma City

Table 7: Examples of using instruction-tuned Llama2$_{7B}$ with different re-rankers. The top combines an LM-enhanced re-ranker, the middle combines a vanilla re-ranker, and the bottom uses no re-ranker.