Unsupervised Information Refinement Training of Large Language Models for Retrieval-Augmented Generation

Shicheng Xu[1,2*] **Liang Pang**[1†] **Mo Yu**[3†] **Fandong Meng**[3] **Huawei Shen**[1]
Xueqi Cheng[1] **Jie Zhou**[3]
[1]CAS Key Laboratory of AI Safety, Institute of Computing Technology, CAS
[2]University of Chinese Academy of Sciences
[3]Pattern Recognition Center, WeChat AI
{xushicheng21s,pangliang,shenhuawei,cxq}@ict.ac.cn
moyumyu@global.tencent.com {fandongmeng,withtomzhou}@tencent.com

Abstract

Retrieval-augmented generation (RAG) enhances large language models (LLMs) by incorporating additional information from retrieval. However, studies have shown that LLMs still face challenges in effectively using the retrieved information, even ignoring it or being misled by it. The key reason is that the training of LLMs does not clearly make LLMs learn how to utilize input retrieved texts with varied quality. In this paper, we propose a novel perspective that considers the role of LLMs in RAG as "Information Refiner", which means that regardless of correctness, completeness, or usefulness of retrieved texts, LLMs can consistently integrate knowledge within the retrieved texts and model parameters to generate the texts that are more concise, accurate, and complete than the retrieved texts. To this end, we propose an information refinement training method named **INFO-RAG** that optimizes LLMs for RAG in an unsupervised manner. INFO-RAG is low-cost and general across various tasks. Extensive experiments on zero-shot prediction of 11 datasets in diverse tasks including Question Answering, Slot-Filling, Language Modeling, Dialogue, and Code Generation show that INFO-RAG improves the performance of LLaMA2 by an average of 9.39% relative points. INFO-RAG also shows advantages in in-context learning and robustness of RAG.

1 Introduction

Retrieval-augmented generation (RAG) is a popular framework in modern NLP systems that equips neural with retrieved information for text generation like open-domain question answering, dialogue (Lewis et al., 2020; Guu et al., 2020) etc. Recently, RAG has been applied to large language models (LLMs) to provide additional knowledge and mitigate issues such as hallucination (Peng et al., 2023; Shi et al., 2023; Ren et al., 2023).

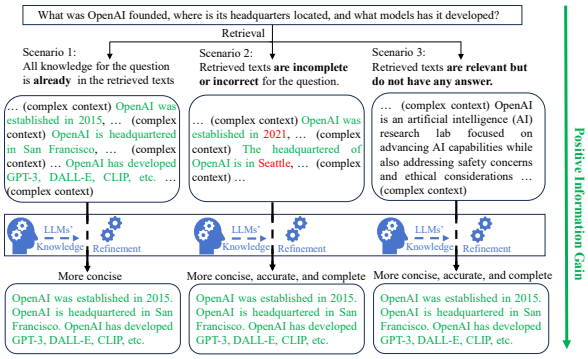

Figure 1: We consider the role of LLMs in RAG as "Information Refiner" that can generate more concise, accurate, and complete texts than the input retrieved texts. In this way, LLM can consistently make RAG system produce positive information gain.

Despite the improved performance of retrieval models, the internet continues to be inundated with fake news, rumors, and fragmented, noisy information, posing challenges for retrieval models to reliably identify and shield against such content (Sun et al., 2022; Thakur et al., 2021). Consequently, not all retrieved texts are beneficial, necessitating that LLMs determine how to judiciously utilize them. However, pre-training tasks do not explicitly enable LLMs to learn how to utilize the retrieved texts with varied quality for generation. For a question and its retrieved texts as input sequence, RAG aims to minimize the negative log-likelihood (NLL) of sub-sequence (question and generated answer) by referring to the retrieved texts. However, mainstream pre-training for LLMs with decoder-only architecture is language modeling based on the prefix (Radford et al., 2018; Touvron et al., 2023a), the training objective aims to minimize the negative log-likelihood (NLL) of the entire input sequence (retrieved texts, question, and generated answer) (Mikolov et al., 2012). This gap causes

*Work done during the Tencent Rhino-bird Research Elite Program at WeChat.
†Corresponding authors.

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 133–145
August 11-16, 2024

LLMs to only regard the input retrieved texts as a part of the prefix for language modeling rather than additional reference, which leads to the following problems. Firstly, for the long and complex retrieved texts, LLMs struggle to extract the correct answers (Deng et al., 2023) accurately. Secondly, in situations where the retrieved texts cannot address the task, LLMs lack the capability to integrate the knowledge within model parameters with the retrieved texts to generate improved texts. Thirdly, LLMs are susceptible to incorrect and noisy information in retrieved texts, posing a risk of being misled (Chen et al., 2023; Yoran et al., 2023).

To solve above problems, some previous methods explore strategies for how or when to perform retrieval for LLMs by prompt techniques (Press et al., 2023; Khattab et al., 2022; Xu et al., 2023; Asai et al., 2023). However, prompt cannot materially change the ability of LLMs to utilize retrieved texts because model parameters are not updated for this ability. Some methods fine-tune LLMs on the constructed RAG data for a specific task such as QA (Yoran et al., 2023; Yu et al., 2023). However, under the trend that LLMs are regarded as foundation models for various tasks in zero-shot setting, fine-tuning LLMs only on a few tasks make LLMs limited to the RAG of training tasks and lose their generalizability. Because catastrophic forgetting still exists in supervised fine-tuning of LLMs (Luo et al., 2023). Although constructing data for a large number of tasks can alleviate this, it is hard to design the data in various RAG tasks and requires high data annotation costs. Our paper aims to fundamentally improve the ability of LLMs to utilize retrieved texts while preserving the generalizability of LLMs for various RAG tasks in zero-shot setting, which is orthogonal to prompt techniques and can be combined with them to get better performance.

In this paper, considering that LLMs have a certain ability to use their own knowledge to examine information (Dhuliawala et al., 2023), we introduce a novel perspective to reassess the role of LLMs in RAG. Specifically, we propose considering LLMs as **"Information Refiner"**. The key idea behind this is to continue training the pre-trained LLMs with an Information Refinement objective that regardless of the correctness, completeness, or usefulness of the input retrieved texts, LLMs can consistently integrate knowledge within the retrieved texts and model parameters to generate the texts that are more concise, accurate, and complete than the retrieved texts (Figure 1). We term this process "Positive Information Gain". This enables LLMs to extract correct information from complex texts as well as resist and rectify retrieved erroneous information and noise, thereby improving the information bottleneck of the RAG and allowing the knowledge capacity of RAG to approximate the combined knowledge of IR and LLMs.

We make the information refinement training work in a completely unsupervised manner, such that it is easy to obtain large-scale training data and maintain the generalizability of the trained LLMs that can be used in various RAG tasks in zero-shot setting. Specifically, we propose an unsupervised training method named INFO-RAG. INFO-RAG classifies the retrieved texts into three scenarios (shown in Figure 1) and proposes the unsupervised training task for each scenario. For the first scenario that all knowledge for the question is already in the retrieved texts, LLMs need to accurately extract relevant knowledge from complex retrieved texts and generate more concise texts. For the second scenario that retrieved texts are incomplete or incorrect for the question, LLMs need to combine the knowledge within model parameters to verify the retrieved texts, correct the wrong knowledge, and complete the missing knowledge. For the third scenario that retrieved texts are relevant but do not have any answer, LLMs need to find the knowledge within model parameters based on relevant context to generate correct answers. We mix the above three tasks to train INFO-RAG unsupervisedly.

Main contributions of this paper are as follows:
(1) We introduce a novel perspective to reassess the role of LLMs in the RAG system that considers LLMs as **"Information Refiner"** that can produce positive information gain in RAG scenarios.
(2) We propose an unsupervised training method named INFO-RAG that enables LLMs to perform information refinement in RAG. INFO-RAG is low-cost and general for various RAG tasks.
(3) Extensive experiments show INFO-RAG enhances the zero-shot RAG of LLaMA2 across Question Answering, Slot-Filling, Language Modeling, Dialog, and Code Generation. INFO-RAG also shows advantages in in-context learning and robustness of RAG. Code is released at https://github.com/xsc1234/INFO-RAG/.

2 Related Work

Retrieval Augmented Generation Retrieval augmented generation (RAG) aims to provide addi-

tional knowledge for language models by retrieving information from external databases (Lewis et al., 2020; Guu et al., 2020; Borgeaud et al., 2022; Izacard et al., 2022). RAG makes the text generated by LLM more accurate and credible, and is widely used in Open-domain QA (Karpukhin et al., 2020; Trivedi et al., 2022a), dialogue (Cai et al., 2018, 2019) and Code Generation (Parvez et al., 2021). Recently, RAG has also been widely applied in LLMs (Peng et al., 2023; Shi et al., 2023; Ren et al., 2023). The form of RAG in LLMs is using the retrieved texts as contexts (Ram et al., 2023).

Some studies have noted that noise in retrieved texts will interfere with the performance of the language model or even mislead it (Xu et al., 2023; Wang et al., 2023; Chen et al., 2023; Xu et al., 2024). These works try to solve this problem from the interactive framework between IR and LM, while our work points out a more essential view. That is, previous studies on RAG do not define the role of LLMs in RAG clearly. Our paper introduces a novel perspective to reassess the role of LLMs in RAG that considers LLMs as "Information Refiner".

Unsupervised Learning of RAG Unsupervised learning of RAG can be divided into the training of retrievers and language models. As for retrievers, REALM (Guu et al., 2020) proposes using masked language modeling to pre-train a knowledge retriever. REPLUG (Shi et al., 2023) trains the retriever according to the feedback from black-box LM. As for language models, RETRO (Borgeaud et al., 2022) improves language models by retrieving tokens. Atlas proposes pretext tasks to jointly train the retriever and language model. However, these two methods focus on the model of encoder-decoder architecture, which is inconsistent with the current mainstream LLMs based on decoder-only.

Previous unsupervised training methods do not consider the specific role that language models should play in RAG. In this paper, we focus on training language model as an "Information Refiner" that can further improve the information bottleneck of RAG and be robust to retrieved texts.

3 Our INFO-RAG

This section introduces our **INFO-RAG**, an unsupervised training method to enable LLMs to perform information refinement in RAG. Firstly, we summarize the retrieved texts in RAG into three scenarios and define the positive information gain for each scenario. Secondly, we construct sample pairs in which the output has information gain compared to the input for these three scenarios and design three training tasks. Thirdly, we train LLMs under our designed tasks on the unsupervised samples. Unsupervised training makes INFO-RAG low-cost and general for RAG in various tasks.

3.1 Positive Information Gain in RAG

In this paper, we introduce a novel perspective to reassess the role of LLMs in RAG that LLMs should be the "Information Refiner" that can produce "Positive Information Gain" in the information flow of RAG. This section details the scenarios of retrieved texts and defines specific information gain LLMs should produce in each scenario.

Scenario 1. The first scenario is that all knowledge for the question is already in the retrieved texts. Even if the correct knowledge already exists in the retrieved texts, complex and lengthy retrieved texts are not conducive for users to directly obtain the knowledge. Therefore, the positive information gain in this scenario means that LLMs extract correct knowledge as much as possible while removing irrelevant information, thereby generating more direct and concise texts for users.

Scenario 2. The second scenario is that although the retrieved texts contain some usable knowledge, they still contain some incomplete or incorrect knowledge. This scenario is very common, especially with the current proliferation of fake news, misinformation, and fragmented knowledge on the Internet. There has been study proving that noise and erroneous knowledge in retrieved texts greatly mislead the generation of LLMs (Xu et al., 2023). The positive information gain in this scenario is that LLMs can exploit the knowledge within their parameters to verify the knowledge in the retrieved texts. Utilize accurate knowledge, rectify incorrect knowledge, and complete missing knowledge

Scenario 3. The third scenario is that the retrieved texts do not have any answer that can used to solve the question. This scenario means that the question is very difficult or the target knowledge is very long-tail for information retrieval systems. Even in this case, the retrieval model's ability to model semantics allows it to provide texts that are semantically related to the question (Karpukhin et al., 2020). Therefore, the positive information gain in this scenario is that LLMs can stimulate the knowledge within their parameters based on semantically relevant context to solve the question.

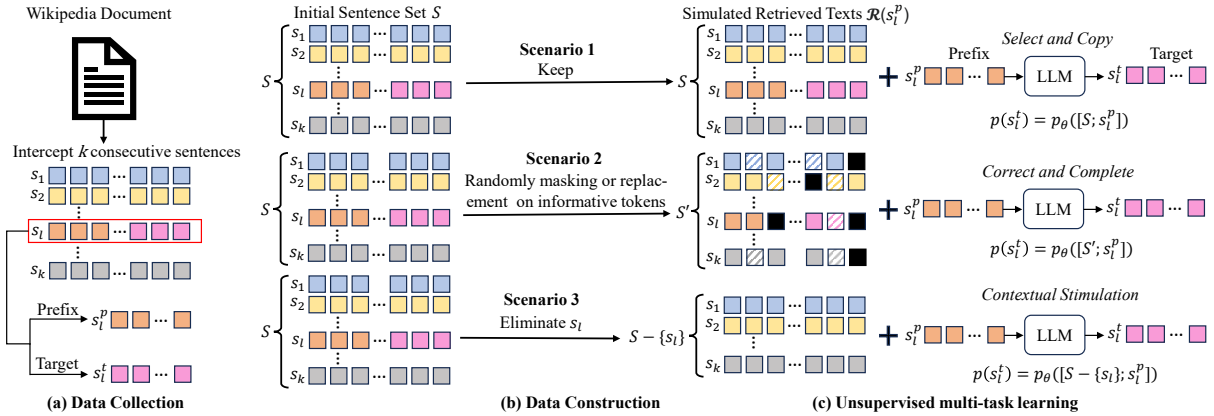

Figure 2: Overview of our INFO-RAG. Each sample is only processed for a single scenario to avoid data leakage.

3.2 Unsupervised Learning

This section introduces unsupervised learning in INFO-RAG. We construct the input-output pairs that satisfy the information gain in the above three scenarios on Wikipedia. We continue to train pre-trained LLMs on the constructed data to perform information refinement in the form of next token prediction in prefix language modeling, which is general for various tasks. Pipeline is in Figure 2.

3.2.1 Data Collection

The data construction is performed on English Wikipedia. Specifically, for each document d in Wikipedia, we intercept k consecutive sentences from d and get the sentence set $S = [s_1, s_2, ..., s_k]$. Our method randomly selects s_l from S and uses it as the object for language modeling. The first $\frac{1}{3}$ to $\frac{2}{3}$ of the tokens of s_l are randomly intercepted as the prefix (s_l^p) and the other tokens of s_l are used as the prediction target (s_l^t). We also perform the process (Section 3.2.2) on sentence set S so that it can be used to simulate the retrieved texts $\mathcal{R}(s_l^p)$ for prefix s_l^p in three scenarios for conditioning the generation of s_l^t. Then, we can get an unsupervised training sample for prefix language modeling that predicts s_l^t given the prefix s_l^p and the retrieved texts $\mathcal{R}(s_l^p)$. This can be formulated as:

$$p(s_l^t) = p_\theta([\mathcal{R}(s_l^p); s_l^p]), \tag{1}$$

θ are parameters of LLMs, $[\mathcal{R}(s_l^p); s_l^p]$ is the concatenation of $\mathcal{R}(s_l^p)$ and s_l^p by a special token.

3.2.2 Data Construction and Training Tasks

This section details our data construction and training tasks for three scenarios in Section 3.1.

For **Scenario 1** that needs LLMs to extract the correct knowledge from the complex texts, we propose the training task named *Select and Copy*. Specifically, given the sentence set S for a sample, *Select and Copy* directly uses all sentences in S as retrieved texts for conditioning LLMs to predict s_l^t for the given prefix s_l^p. This can be formulated as:

$$p(s_l^t) = p_\theta([S; s_l^p]). \tag{2}$$

In *Select and Copy*, s_l (both s_l^p and s_l^t) has been contained in the retrieved texts S, this needs LLMs to select the texts matching the prefix s_l^p from the complex retrieved texts S and directly copy the target s_l^t for generation. The information gain between s_l^t and input retrieved texts S is that s_l^t is more concise to be used as the postfix for s_l^p.

For **Scenario 2** that needs LLMs to verify the knowledge in the retrieved texts, utilize accurate knowledge, rectify incorrect knowledge, and complete missing knowledge. We propose the training task named *Correct and Complete*. Given a sentence set S, firstly, this task uses the stability of word distribution between layers to get informative tokens. The intention for this is that the more unstable the word distribution of the token among the topmost layers is, the more it indicates that the token is an informative token. We follow (Chuang et al., 2023) to achieve this. Specifically, for each sentence s_i in S, our method obtains the next word distribution of the a-th token $s_i^{[a]}$ given prefix $s_i^{<a}$ of s_i in each layer of LLM as:

$$d_j(s_i^{[a]}|s_i^{<a}) = \text{softmax}(\mathbf{W}\mathbf{H}_j^{[a]}), \tag{3}$$

in which j indicates the j-th layer of LLMs, $\mathbf{H}_j^{[a]} \in \mathbb{R}^h$ is the hidden states for token $s_i^{[a]}$ in the j-th layer, $\mathbf{W} \in \mathbb{R}^{h \times v}$ is the vocabulary head that maps the hidden states $\mathbf{H}_j^{[a]}$ to the word distribution with

vocabulary size v. Then, for the LLM with N layers, our method uses Jensen-Shannon Divergence (JSD) to measure the differences in word distribution between layers and gets the word distribution stability of token $s_i^{[a]}$ as:

$$O_i^{[a]} = \max_{j \in J} \text{JSD}(d_N(s_i^{[a]}|s_i^{<a})||d_j(s_i^{[a]}|s_i^{<a})),$$

in which J is the set of candidate layers (0-th to $\frac{N}{2}$-th layers), $d_N(s_i^{[a]}|s_i^{<a})$ is the word distribution of the last layer. This design aims to find the layer with the largest word distribution difference between the last layer and use the JSD of the two as the word distribution stability of the token $s_i^{[a]}$ (Chuang et al., 2023). For each token of s_i, we obtain its word distribution stability in parallel and get the set of word distribution stability for s_i as:

$$\mathbb{O}_i = \{O_i^{[0]}, O_i^{[1]}, ..., O_i^{[n]}\}. \quad (4)$$

We choose the tokens corresponding to the top 50% of the elements in $\mathbb{O}_i$ as informative tokens within the sentence s_i. Subsequently, we apply a specific percentage (30%) of random masking and replacement to these tokens. For the randomly selected token, we replace it with [MASK] with a 50% probability to simulate the incomplete knowledge, and randomly replace it with another token with a 40% probability to simulate the incorrect knowledge, while keeping it unchanged with a 10% probability to simulate the correct knowledge. We do the above pipeline for each sentence in the set S and get the processed set S'. RAG in *Correct and Complete* can be formulated as:

$$p(s_l^t) = p_\theta([S'; s_l^p]). \quad (5)$$

In *Correct and Complete*, the broken s_l with noise is already in S'. The information gain in this task requires LLM to extract, correct, and complete the knowledge in s_l from S' to generate s_l^t.

For **Scenario 3** that needs LLMs to find answers from their knowledge based on relevant texts in context. We propose the training task named *Contextual Stimulation*. *Contextual Stimulation* eliminates s_l (both s_l^p and s_l^t) from the set S and uses the remaining sentences as retrieved tests for generation, which can be formulated as:

$$p(s_l^t) = p_\theta([S - \{s_l\}; s_l^p]). \quad (6)$$

In *Contextual Stimulation*, each sentence in retrieved texts $S - \{s_l\}$ is semantically relevant to s_l^p but cannot help LLMs to directly generate s_l^t. LLMs need to be stimulated by relevant information to generate s_l^t based on their own knowledge.

3.2.3 Training Strategy

After the data construction for three training tasks, we mix them for multi-task training. Specifically, we use LoRA (Hu et al., 2021) to train the pre-trained LLMs on the mixed dataset of three tasks. Three tasks are trained alternately in batches. Since *Select and Copy* is relatively simple for LLMs, it only accounts for 20% of the batches, while *Correct and Complete* and *Contextual Stimulation* each account for 40% of the batches. Using LoRA not only reduces training costs but also makes our method plug-and-play. The trained LoRA parameters are loaded when LLMs need to perform RAG and unloaded when RAG is not needed.

4 Experiments

4.1 Datasets and Evaluation Metrics

To demonstrate the generality of our unsupervised training method, we evaluate the performance of INFO-RAG on eleven datasets across seven tasks.

Open-domain Question Answering Open-domain QA is a typical knowledge-intensive task that can directly evaluate the knowledge of LLMs. We use Natural Questions (Kwiatkowski et al., 2019) (NQ) and WebQuestions (Berant et al., 2013) (WebQ) as the datasets. We use cover Exact Match (EM) to determine whether the ground truth exactly appears in the output and the accuracy is used as the evaluation metric, following (Schick et al., 2023)

Soft Filling Soft filling requires LLMs to output the object entities for the input subject entity and relation. We use two knowledge-intensive datasets including Zero Shot RE (Levy et al., 2017) (ZS) and T-REx (Elsahar et al., 2018). We use the same evaluation metric as Open-domain QA.

Long-Form Question Answering Compared with open-domain QA, LFQA is the QA task whose ground truth answer is a relatively long text. We use ELI5 (Fan et al., 2019), a knowledge-intensive dataset for LFQA. We use ROUGE-L as the evaluation metric (Petroni et al., 2020).

Dialogue Dialogue in our experiment focuses on the factual knowledge. We use Wizard of Wikipedia (Dinan et al., 2018) (WoW), a knowledge-powered dialogue dataset whose conversation is grounded with knowledge. We use F1 as the evaluation metric (Petroni et al., 2020).

Language Modeling We use WikiText-103 (Merity, 2016), a popular dataset for language modeling. We use ROUGE-L as the evaluation metric.

	Soft-Filling		ODQA		Multi-Hop QA		LFQA	Dialog	LM	Code Gen		
	Accuracy		Accuracy		Accuracy		ROUGE	F1	ROUGE	CodeBLEU		Overall
	T-REx	ZS	NQ	WebQ	Hotpot	Musique	Ell5	Wow	WikiText	Python	Java	
LLaMA-2-7B	55.60	54.08	**46.82**	43.52	39.40	25.95	15.18	7.85	60.77	21.44	22.99	35.78
+ INFO-RAG	**65.91**	**57.01**	45.74	**44.68**	**46.56**	**30.19**	**17.18**	**9.09**	**62.91**	**26.75**	**32.06**	**39.83**
LLaMA-2-7B-chat	60.63	55.03	49.42	46.72	50.03	42.69	27.81	10.21	60.26	22.46	23.90	40.83
+ INFO-RAG	**65.77**	**58.32**	**53.93**	**49.13**	**52.01**	**44.45**	**28.15**	**10.49**	**63.24**	**27.25**	**28.79**	**43.78**
LLaMA-2-13B	60.08	50.77	47.40	44.62	42.12	25.78	14.80	7.04	62.20	21.52	29.16	36.86
+ INFO-RAG	**62.80**	**55.63**	**47.82**	**45.42**	**51.48**	**35.02**	**17.48**	**7.20**	**64.14**	**29.00**	**35.50**	**41.04**
LLaMA-2-13B-chat	62.53	56.81	50.36	45.47	61.23	47.06	27.07	11.19	60.52	22.34	30.96	43.23
+ INFO-RAG	**65.39**	**59.05**	**54.04**	**51.07**	**61.91**	**47.93**	**27.24**	**11.38**	**63.92**	**31.98**	**38.12**	**46.55**

Table 1: Overall performance on retrieval-augmented generation on 11 datasets across 7 tasks in zero-shot setting.

Multi-Hop Question Answering Multi-hop QA measures the ability of LLMs to perform combined reasoning on multiple knowledge. We use HotpotQA (Yang et al., 2018) and Musique (Trivedi et al., 2022b) for this task. We use the same evaluation metric as Open-domain QA.

Code Generation Code generation aims to generate the code for the given natural language. We use Java and Python in CodeXGLUE (Iyer et al., 2018) for this task. We use CodeBLEU (Ren et al., 2020) as the evaluation metric.

4.2 Experimental Settings

LLMs in our paper include LLaMA-2-7B, 13B and their chat version (Touvron et al., 2023b). We use LoRA to fine-tune these pre-trained LLMs on four A100 GPUs with the learning rate of 1e-5, per-gpu batch size of 4 (for 7B) and 2 (for 13B) for 5K steps. As for the training data, we intercept 15 consecutive sentences for each example.

As for the retrieval model and retrieval database, for Open-domain QA, Soft Filling and Language Modeling, we use ColBERTv2 (Santhanam et al., 2022), a late-interaction model with excellent generalization ability as the retriever, and use Wikipedia consisting of 21,015,324 passages (Karpukhin et al., 2020) as retrieval database. For Code Generation, we SCODE-R (Parvez et al., 2021) as code retriever and use deduplicated source codes in CodeSearchNET (Husain et al., 2019) as retrievel database. For all the above tasks, we give Top-5 retrieved passages to each example. For LFQA, Dialog, and Multi-Hop QA, we use the list of contextual passages provided in the datasets as the retrieved list (distractor setting). **In each experiment, all baselines and our method share the same retrieved documents.**

4.3 Experimental Results

Main Results (Zero-Shot Setting) Experimental results in Table 1 show the improvement (the average is 9.39%) of our method on the utilization of retrieved knowledge from four aspects.

(1) Short and Direct Knowledge. Our method can significantly improve the RAG performance of LLaMA on ODQA and Slot-Filling tasks. The answer in ODQA and Slot-Filling is short and direct, it can directly reflect the ability of LLMs to utilize the knowledge in retrieved texts.

(2) Reasoning on Multiple Knowledge. Our INFO-RAG has advantages in cross-passage reasoning on multiple knowledge of retrieval lists. Questions in both HotpotQA and Musique are complex and need multiple knowledge from different passages. These questions not only require LLMs to extract correct knowledge from the retrieved passage list but also to combine the knowledge of different passages in the list for reasoning to give the final answer.

(3) Long and Complex Knowledge. Our INFO-RAG can improve the RAG performance of LLaMA on LFQA, Dialogue and Language Modeling. These tasks require LLaMA to output long and complex texts grounded with intensive knowledge.

(4) Code Knowledge. Our INFO-RAG can also improve the RAG performance of LLaMA on Code Generation. This further demonstrates the cross-task generality of INFO-RAG. Our method is only trained on natural language but can also show advantages in programming language tasks, which demonstrates that INFO-RAG successfully enables LLMs to learn how to exploit the retrieved information rather than just fitting the data. Unsupervised and prefix language modeling training paradigms make INFO-RAG general in various tasks.

	T-REx			ZS			NQ			WebQ		
	has-ans.	replace	no-ans.	has-ans.	replace	no-ans.	has-ans.	replace	no-ans.	has-ans.	replace	no-ans.
LLaMA-2-7B	67.19	38.37	6.49	64.41	12.78	2.44	**65.54**	16.91	3.41	60.64	25.68	7.90
+ INFO-RAG	**79.80**	**41.79**	**7.04**	**68.10**	**13.55**	**3.26**	64.43	**22.68**	**4.70**	**62.70**	**26.48**	**8.96**
LLaMA-2-7B-chat	73.79	40.56	4.87	66.71	14.19	1.63	68.72	20.81	4.50	66.86	28.63	5.62
+ INFO-RAG	**80.01**	**42.92**	**5.42**	**69.64**	**15.02**	**2.65**	**70.99**	**23.14**	**5.62**	**68.73**	**29.74**	**9.12**
LLaMA-2-13B	72.26	39.47	7.76	60.14	19.71	4.69	**65.94**	18.45	4.42	62.09	26.63	9.27
+ INFO-RAG	**75.80**	**44.08**	**8.48**	**65.94**	**23.21**	**4.90**	64.98	**27.60**	**8.02**	**63.51**	**28.24**	**9.88**
LLaMA-2-13B-chat	75.96	43.79	5.59	67.03	16.58	1.42	69.37	30.72	6.16	65.07	31.88	5.47
+ INFO-RAG	**79.25**	**48.59**	**6.67**	**70.26**	**25.02**	**3.87**	**73.73**	**33.85**	**8.39**	**70.59**	**37.48**	**11.25**

Table 2: Experimental results on three scenarios. "has-ans." is the first scenario that correct answers are in retrieved texts. "replace" is the second scenario that correct answers are randomly replaced with other phrases to simulate the incorrect and incomplete knowledge. "no-ans." is the third scenario that retrieval cannot find any answers.

Results on In-context Learning for RAG Besides, our INFO-RAG allows further improvement cooperating with in-context learning (ICL). ICL (Brown et al., 2020) works by prepending a few examples of the target task before the query, which helps LLMs understand the task. However, ICL may not always help in the RAG setting, mainly due to the confusion between the retrieved texts of the query and the few-shot examples. As shown in Table 3, LLaMA-2 cannot further improve the RAG performance from ICL, even sometimes hurt by the few-shot examples while INFO-RAG can further improve RAG by ICL. This is mainly because INFO-RAG enables LLaMA to understand the task form of RAG, thereby better learning the general task pattern from ICL examples. In this experiment, we construct the ICL example consisting of a query, a relevant passage, and an answer. For a fair comparison, we need to ensure that the performance of our method and the baseline are close in non-ICL setting. Therefore, we select queries for which the baseline gives the same answer as our method (both correct or both incorrect) and evaluate the ICL performance on these queries.

Enhancing Previous SOTA in Open-Retrieval Setting We further show that our INFO-RAG can cooperate well with the recent prompting techniques that perform multi-step reasoning to combine with retrieval to solve questions (Xu et al., 2023; Khattab et al., 2022; Press et al., 2023; Yao et al., 2022). To make a fair comparison, we follow SearChain (Xu et al., 2023) that runs on Multi-Hop QA and Slot-Filling in open-retrieval setting that retrieves passages from the full Wikipedia in each reasoning step. SearChain and other baselines use LLaMA-2-13B-chat as the backbone. Then, we

Data	Model	Number of Examples in ICL					
		0	2	4	8	12	16
NQ	LLaMA-2	43.36	23.34	16.60	39.22	44.32	43.00
	+INFO-RAG	43.36	**44.35**	**45.88**	**44.45**	**47.75**	**46.25**
WebQ	LLaMA-2	43.20	18.36	9.40	36.71	44.80	44.81
	+INFO-RAG	43.20	**48.03**	**49.82**	**48.25**	**47.86**	**47.29**
T-REx	LLaMA-2	59.83	47.05	49.11	56.51	55.23	56.31
	+INFO-RAG	59.83	**63.08**	**63.45**	**63.54**	**63.57**	**63.38**
ZS	LLaMA-2	52.41	42.71	37.05	50.40	50.20	51.01
	+INFO-RAG	52.41	**56.53**	**60.37**	**59.86**	**59.75**	**59.85**

Table 3: RAG performance changes with number of examples in In-context learning.

	Multi-Hop QA		Slot-Filling	
	HotpotQA	Musique	T-REx	zsRE
Previous SOTA	28.19	10.03	63.10	57.09
SearChain	31.21	11.27	64.58	58.91
+ INFO-RAG	**33.04**	**12.10**	**66.95**	**60.72**

Table 4: Enhancement to the state-of-the-art RAG framework. Previous SOTA includes DSP, Self-Ask, React.

perform SearChain based on LLaMA-2-13B-chat trained by INFO-RAG to show the enhancement to SearChain by INFO-RAG. Results in Table 4 show that INFO-RAG can make SearChain achieve better performance. This provides additional support that our unsupervised INFO training fundamentally improves the RAG performance of LLMs.

4.4 Analysis

Fine-grained Analysis for Three Scenarios As shown in Table 2, our INFO-RAG is effective in all three RAG scenarios and shows better robustness to incorrect, incomplete, and noisy retrieved texts. We propose corresponding unsupervised training tasks for the three scenarios of RAG. This section introduces the fine-grained analysis for each sce-

	T-REx	ZS	NQ	WebQ	Hotpot	Musique	ElI5	Wow	WikiText	Python	Java	Overall
LLaMA-2 w/o RAG	35.60	10.99	32.67	39.13	29.16	5.83	26.05	10.71	41.80	20.67	25.87	25.32
LLaMA-2 w/ RAG	62.53	56.81	50.36	45.47	61.23	47.06	27.07	11.19	60.52	22.34	30.96	43.23
+ training on wiki	62.55	56.79	49.23	45.05	61.00	46.95	26.31	11.05	60.84	22.05	30.28	42.92
+ INFO-RAG	**65.39**	**59.05**	**54.04**	**51.07**	**61.91**	**47.93**	**27.24**	**11.38**	**63.92**	**31.98**	**38.12**	**46.55**

Table 5: Analysis on the best-performed model LLaMA-2-13B-chat.

Method	NQ			
	original	has-ans.	replace	no-ans.
Baseline	50.36	69.37	30.72	6.16
S1: *Select and Copy*	48.77	69.59	25.40	0.11
S2: *Correct and Complete*	51.59	70.42	32.71	4.48
S3: *Contextual Stimulation*	52.75	72.50	31.77	8.86
S2&S3	53.73	73.01	32.50	9.01
INFO-RAG (S1& S2&S3)	54.04	73.73	33.85	8.39

Table 6: Effects of three training tasks.

Datasets	Method	Max Δ ratio	Max Δ position	Max Δ number
NQ	LLaMA-2	-51.94%	-16.18%	-25.43%
	+ INFO-RAG	**-43.48%**	**-15.80%**	**-17.25%**
WebQ	LLaMA-2	-50.57%	**-5.63%**	-22.13%
	+ INFO-RAG	**-45.48%**	-8.72%	**-11.91%**
T-REx	LLaMA-2	-46.57%	-9.45%	-5.95%
	+ INFO-RAG	**-44.38%**	**-8.61%**	**-2.99%**
ZS	LLaMA-2	-59.25%	-13.40%	-12.37%
	+ INFO-RAG	**-50.08%**	**-11.11%**	**-11.43%**

Table 7: Maximum relative performance change caused by changes in retrieval results.

nario. For **Scenario 1**, we use cover EM to select those samples that already contain the correct answers in the retrieval list. For **Scenario 2**, we randomly replace the correct answers in the retrieved texts with another phrase with the same properties. For **Scenario 3**, we use cover EM to select those samples that retrieved texts do not contain any correct answers. We count the accuracy of LLaMA on samples of these three scenarios respectively. Questions in the third scenario are more difficult than in the second scenario because retrieval models cannot find anything to solve them. Table 2 indicates that our method shows advantages in each scenario and is more robust regardless of whether the retrieved texts contain the correct answer.

Ablation Study We conduct ablation study to explore the effects of the following factors.

(1) Additional Training on Wikipedia. We study whether our improvement is from helping the model to achieve information refinement, or simply because of additional training on Wikipedia. To this end, we train LLaMA-2 on Wikipedia with standard language modeling objective, by setting the same hyperparameters as our INFO-RAG. The results in Table 5 show that this baseline leads to no improvement over the backbone LLaMA-2, confirming the effectiveness of our training method rather than additional training on Wikipedia.

(2) Training tasks. We perform three training tasks proposed in INFO-RAG separately on original data and data constructed for each scenario to explore their effects respectively. Table 6 shows that both S2 and S3 have gains in their scenarios. Although S1 has negative effects when performed alone, it can achieve the best results when trained together with S2 and S3. This is mainly because S1 alone is so simple that causes LLM to overfit the data. Adding S2 and S3 allows LLM to learn the task paradigm of information refinement, making LLM better extract the correct answer for **Scenario 1**.

Robustness to Retrieval Results Table 7 shows INFO-RAG is more robust to changes in retrieval results including the ratio and position of positive passages and number of retrieved passages. More details can be found in Section A of Appendix.

Avoid Catastrophic Forgetting Experiment on MMLU (Hendrycks et al., 2020) without RAG shows that INFO-RAG performs very close to the original LLaMA-2 (7B: 45.0 vs. 45.3; 13B: 54.3 vs. 54.8), which indicates that INFO-RAG enhances RAG while avoiding catastrophic forgetting. More details can be found in Section A.6 of Appendix.

5 Conclusion

This paper proposes a novel perspective to reassess the role of LLMs in RAG that considers LLMs as "Information Refiner". This means that regardless of the correctness, completeness, or usefulness of the retrieved texts, LLMs can consistently integrate knowledge within model parameters and the retrieved texts to generate texts that are more concise, accurate, and complete. To achieve it, we propose an information refinement training method named

INFO-RAG in an unsupervised manner, which is low-cost and general across various tasks. Extensive experiments across 11 datasets of 7 tasks in zero-shot setting show that INFO-RAG improves the performance of LLMs for RAG. INFO-RAG also shows advantages in ICL and robustness of RAG and can be combined with the SOTA RAG framework to further improve its performance.

Limitations

This paper aims to enable LLMs to perform information refinement in RAG by unsupervised training, so as to accurately extract correct information and avoid the interference of incorrect information. The main limitation of this paper is that due to the lack of computing resources, we only conduct experiments on models with 7B and 13B parameter sizes. In the future, we consider using more computing resources to explore the performance of models with larger parameter sizes.

Ethics Statement

After careful consideration, we believe that our paper does not introduce additional ethical concerns. We declare that our work complies with the ACL Ethics Policy.

Acknowledgements

This work was supported by the National Key R&D Program of China (2022YFB3103700, 2022YFB3103704), the National Natural Science Foundation of China (NSFC) under Grants No. 62276248 and U21B2046, and the Youth Innovation Promotion Association CAS under Grants No. 2023111.

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 preprint arXiv:2310.11511*.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the EMNLP 2013*, pages 1533–1544.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. 2022. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR.

Tom B. Brown, Benjamin Mann, Nick Ryder, et al. 2020. Language models are few-shot learners.

Deng Cai, Yan Wang, Victoria Bi, Zhaopeng Tu, Xiaojiang Liu, Wai Lam, and Shuming Shi. 2018. Skeleton-to-response: Dialogue generation guided by retrieval memory. *arXiv preprint arXiv:1809.05296*.

Deng Cai, Yan Wang, Wei Bi, Zhaopeng Tu, Xiaojiang Liu, and Shuming Shi. 2019. Retrieval-guided dialogue response generation via a matching-to-generation framework. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1866–1875.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2023. Benchmarking large language models in retrieval-augmented generation. *arXiv preprint arXiv:2309.01431*.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James Glass, and Pengcheng He. 2023. Dola: Decoding by contrasting layers improves factuality in large language models. *arXiv preprint arXiv:2309.03883*.

Jingcheng Deng, Liang Pang, Huawei Shen, and Xueqi Cheng. 2023. Regavae: A retrieval-augmented gaussian mixture variational auto-encoder for language modeling. *arXiv preprint arXiv:2310.10567*.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2023. Chain-of-verification reduces hallucination in large language models.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2018. Wizard of wikipedia: Knowledge-powered conversational agents. *arXiv preprint arXiv:1811.01241*.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. 2018. T-rex: A large scale alignment of natural language with knowledge base triples. In *Proceedings of LREC 2018*.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. Eli5: Long form question answering. *arXiv preprint arXiv:1907.09190*.

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

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Hamel Husain, Ho-Hsiang Wu, Tiferet Gazit, Miltiadis Allamanis, and Marc Brockschmidt. 2019. Codesearchnet challenge: Evaluating the state of semantic code search. *arXiv preprint arXiv:1909.09436*.

Srinivasan Iyer, Ioannis Konstas, Alvin Cheung, and Luke Zettlemoyer. 2018. Mapping language to code in programmatic context. *arXiv preprint arXiv:1808.09588*.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022. Few-shot learning with retrieval augmented language models. *arXiv preprint arXiv:2208.03299*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781.

Omar Khattab, Keshav Santhanam, Xiang Lisa Li, David Hall, Percy Liang, Christopher Potts, and Matei Zaharia. 2022. Demonstrate-search-predict: Composing retrieval and language models for knowledge-intensive nlp. *arXiv preprint arXiv:2212.14024*.

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

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. *arXiv preprint arXiv:1706.04115*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2023. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *arXiv preprint arXiv:2308.08747*.

Stephen Merity. 2016. The wikitext long term dependency language modeling dataset. *Salesforce Metamind*, 9.

Tomáš Mikolov et al. 2012. Statistical language models based on neural networks. *Presentation at Google, Mountain View, 2nd April*, 80(26).

Md Rizwan Parvez, Wasi Uddin Ahmad, Saikat Chakraborty, Baishakhi Ray, and Kai-Wei Chang. 2021. Retrieval augmented code generation and summarization. *arXiv preprint arXiv:2108.11601*.

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, et al. 2023. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *arXiv preprint arXiv:2302.12813*.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, et al. 2020. Kilt: a benchmark for knowledge intensive language tasks. *arXiv preprint arXiv:2009.02252*.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *arXiv preprint arXiv:2302.00083*.

Ruiyang Ren, Yuhao Wang, Yingqi Qu, Wayne Xin Zhao, Jing Liu, Hao Tian, Hua Wu, Ji-Rong Wen, and Haifeng Wang. 2023. Investigating the factual knowledge boundary of large language models with retrieval augmentation. *arXiv preprint arXiv:2307.11019*.

Shuo Ren, Daya Guo, Shuai Lu, Long Zhou, Shujie Liu, Duyu Tang, Neel Sundaresan, Ming Zhou, Ambrosio Blanco, and Shuai Ma. 2020. Codebleu: a method for automatic evaluation of code synthesis. *arXiv preprint arXiv:2009.10297*.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. Colbertv2: Effective and efficient retrieval via lightweight late interaction.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *arXiv preprint arXiv:2302.04761*.

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

Tiening Sun, Zhong Qian, Sujun Dong, Peifeng Li, and Qiaoming Zhu. 2022. Rumor detection on social media with graph adversarial contrastive learning. In *Proceedings of the WWW 2022*, pages 2789–2797.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022a. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. *arXiv preprint arXiv:2212.10509*.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022b. Musique: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Yile Wang, Peng Li, Maosong Sun, and Yang Liu. 2023. Self-knowledge guided retrieval augmentation for large language models. *arXiv preprint arXiv:2310.05002*.

Shicheng Xu, Liang Pang, Huawei Shen, Xueqi Cheng, and Tat-seng Chua. 2023. Search-in-the-chain: Towards the accurate, credible and traceable content generation for complex knowledge-intensive tasks. *arXiv preprint arXiv:2304.14732*.

Shicheng Xu, Liang Pang, Jun Xu, Huawei Shen, and Xueqi Cheng. 2024. List-aware reranking-truncation joint model for search and retrieval-augmented generation. *arXiv preprint arXiv:2402.02764*.

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. *arXiv preprint arXiv:1809.09600*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2023. Making retrieval-augmented language models robust to irrelevant context. *arXiv preprint arXiv:2310.01558*.

Wenhao Yu, Hongming Zhang, Xiaoman Pan, Kaixin Ma, Hongwei Wang, and Dong Yu. 2023. Chain-of-note: Enhancing robustness in retrieval-augmented language models.

A More Analysis

A.1 Robustness to ratio of Positive Passages

Our INFO-RAG improves the robustness of RAG performance to retrieval performance. The performance of the retriever greatly affects the performance of LLM in RAG (Chen et al., 2023). We explore this in this section. Specifically, we simulate changes in retrieval performance by varying the ratio of positive and negative passages in the retrieved list and report the RAG performance with different ratios. Table 8 shows INFO-RAG performs better when the ratio is low and the performance is more stable than baseline when the ratio changes from 100% to 0% (Max Δ). The model in this experiment is LLaMA-2-13B-chat.

A.2 Robustness to Positive Passage Position

Experimental results in Table 9 show that our INFO-RAG consistently outperforms the baseline (LLaMA-2) regardless of where the positive passage (passage contains the correct answers) appears in the retrieved list. Specifically, we mix positive and negative passages in a ratio of 1:9 to simulate the retrieved passage list, vary the position of the positive passage in the retrieved list from 0 to 9, and evaluate the corresponding RAG performance respectively. The model in this experiment is LLaMA-2-13B-chat. Experimental results show that our INFO-RAG not only outperforms the baseline at every position but also achieves more stable performance varying with the position (Max Δ).

A.3 Robustness to Number of Retrieved Passages

Experimental results in Table 10 show that our INFO-RAG consistently outperforms the baseline with the different number of retrieved passages (from 1 to 10) and is robust to the change of the number. In this experiment, we use LLaMA-2-13B-chat as the base model, change the number of retrieved passages from 1 to 10, and evaluate the corresponding performance.

Data	Model	ratio of Positive Passages						Max Δ
		100%	80%	60%	40%	20%	0%	
NQ	LLaMA-2	88.11	82.71	80.81	77.62	69.73	42.35	-51.94%
	+ INFO-RAG	**90.31**	**83.72**	**81.72**	**79.72**	**71.52**	**51.04**	**-43.48%**
WebQ	LLaMA-2	79.41	75.43	71.63	65.53	63.39	39.25	-50.57%
	+ INFO-RAG	**83.66**	**76.23**	**74.23**	**69.05**	**65.74**	**45.61**	**-45.48%**
T-REx	LLaMA-2	80.01	70.05	71.52	68.53	66.23	42.75	-46.57%
	+ INFO-RAG	**83.52**	**73.22**	**74.93**	**72.32**	**70.12**	**46.45**	**-44.38%**
ZS	LLaMA-2	69.52	65.48	63.81	60.95	57.14	28.33	-59.25%
	+ INFO-RAG	**72.50**	**72.62**	**67.62**	**67.86**	**60.48**	**36.19**	**-50.08%**

Table 8: RAG performance changes with the ratio of positive passages (randomly select 500 samples).

Datasets	Method	Position of Positive Passage										Max Δ
		0	1	2	3	4	5	6	7	8	9	
NQ	LLaMA-2	54.94	48.05	46.05	46.45	46.35	48.30	48.35	47.15	51.64	50.44	-16.18%
	+ INFO-RAG	**63.23**	**58.34**	**54.54**	**54.44**	**53.54**	**53.24**	**53.84**	**54.44**	**53.34**	**53.34**	**-15.80%**
WebQ	LLaMA-2	66.13	63.21	62.54	62.68	64.01	62.41	63.21	64.54	63.87	64.14	**-5.63%**
	+ INFO-RAG	**71.58**	**68.39**	**66.26**	**65.34**	**67.19**	**65.73**	**65.73**	**65.81**	**65.54**	**66.72**	-8.72%
T-REx	LLaMA-2	64.43	60.13	58.34	60.23	58.54	59.14	59.74	60.53	63.53	63.23	-9.45%
	+ INFO-RAG	**70.72**	**66.23**	**64.93**	**65.23**	**65.43**	**64.83**	**66.03**	**67.23**	**64.63**	**66.83**	**-8.61%**
ZS	LLaMA-2	63.04	59.04	54.59	55.03	55.17	57.15	56.42	57.89	58.04	59.47	-13.40%
	+ INFO-RAG	**66.42**	**63.33**	**59.04**	**60.23**	**61.42**	**61.66**	**60.00**	**61.19**	**60.23**	**62.14**	**-11.11%**

Table 9: RAG performance changes with the position of positive passage (randomly select 500 samples).

	T-REx	ZS	NQ	WebQ
Baseline	51.47	40.26	45.05	41.78
+ INFO-RAG	**55.67**	**43.29**	**49.76**	**44.02**

Table 11: Works based on BM25.

Method	T-REx	ZS	NQ	WebQ
Baseline	62.53	56.81	50.36	45.47
Simple Mask	**64.05**	**58.91**	**53.80**	**50.55**
Our method	**65.39**	**59.05**	**54.04**	**51.07**

Table 13: Ablation study of masking strategy.

A.4 Ablation Study on Masking Strategy

In general, Table 13 and 12 show our masking strategy in Scenario 3 is more effective than simple and straightforward masking. Specifically, our method is more significantly effective in the scenarios that correct answers are randomly replaced with other phrases (replace) and retrieval cannot find any answers (no answer).

A.5 Works with Different Retriever

We evaluate our method and baseline (LLaMA2-13B-chat) with BM25 as the retriever, the experimental results shown in Table 11 indicate that our method still performs better than baseline when the retriever as BM25.

A.6 Performance on MMLU

Experimental results on MMLU benchmark in the setting without RAG shown in Table 14 show that our INFO-RAG significantly improves the performance of LLMs in RAG, while still maintaining its versatility and avoiding catastrophic forgetting. MMLU is a benchmark that measures massive multitask language understanding ability of LLMs. It covers 57 subjects across STEM, the humanities, the social sciences, and more. It ranges in difficulty from an elementary level to an advanced professional level, and it tests both world knowledge and problem-solving ability (Hendrycks et al., 2020). Experiments show that our INFO-RAG performs very close to the original LLaMA-2 on MMLU, which shows that our INFO-RAG does not damage the basic language understanding ability of LLMs. This is mainly because the prefix language model-

Datasets	Method	Number of Retrieved Passages										Max Δ
		1	2	3	4	5	6	7	8	9	10	
NQ	LLaMA-2	38.80	43.21	46.62	47.84	48.61	49.42	52.03	50.23	50.40	50.20	-25.43%
	+ INFO-RAG	**45.18**	**46.80**	**51.44**	**51.23**	**51.00**	**53.21**	**54.03**	**53.44**	**53.82**	**54.60**	**-17.25%**
WebQ	LLaMA-2	40.22	43.63	48.20	46.61	48.32	49.11	49.40	50.22	51.65	50.43	-22.13%
	+ INFO-RAG	**50.21**	**53.84**	**54.41**	**55.07**	**55.25**	**55.27**	**57.00**	**55.45**	**56.62**	**56.03**	**-11.91%**
T-REx	LLaMA-2	66.20	63.45	**67.22**	64.45	64.43	65.40	64.41	65.22	63.22	65.01	-5.95%
	+ INFO-RAG	**66.25**	**66.03**	66.31	**65.80**	**67.23**	**67.22**	**66.65**	**67.83**	**67.03**	**67.40**	**-2.99%**
ZS	LLaMA-2	49.25	50.01	52.38	54.09	56.12	56.20	56.13	56.05	55.95	56.11	-12.37%
	+ INFO-RAG	**53.17**	**54.08**	**56.35**	**58.01**	**59.45**	**59.12**	**59.40**	**58.55**	**60.03**	**59.08**	**-11.43%**

Table 10: RAG performance changes with the number of retrieved passages (randomly select 500 samples).

	T-REx			ZS			NQ			WebQ		
	has-ans.	replace	no-ans.	has-ans.	replace	no-ans.	has-ans.	replace	no-ans.	has-ans.	replace	no-ans.
Baseline	75.96	43.79	5.59	67.03	16.58	1.42	69.37	30.72	6.16	65.07	31.88	5.47
Simple Mask	78.43	44.05	5.75	**70.30**	19.45	1.96	73.59	31.05	6.51	70.55	32.96	6.83
Our method	**79.25**	**48.59**	**6.67**	70.26	**25.02**	**3.87**	**73.73**	**33.85**	**8.39**	**70.59**	**37.48**	**11.25**

Table 12: Ablation study of masking strategy on three scenarios. "has-ans." is the first scenario that correct answers are in retrieved texts. "replace" is the second scenario that correct answers are randomly replaced with other phrases to simulate the incorrect and incomplete knowledge. "no-ans." is the third scenario that retrieval cannot find any answers.

ing training paradigm of our method is consistent with the pre-training task of LLMs. The difference is that in the training of prefix language modeling, our method learns to perform information refinement that utilizes the retrieved texts for the next token prediction.

	Humanities	STEM	Social-Sciences	Other	Average
LLaMA-2-7B w/o RAG	42.9	36.4	51.2	52.2	45.3
+ INFO-RAG w/o RAG	42.8	36.1	50.8	52.0	45.0
LLaMA-2-13B w/o RAG	52.8	44.1	62.6	61.1	54.8
+ INFO-RAG w/o RAG	52.5	43.7	62.1	60.9	54.3

Table 14: Performance on MMLU in the setting without retrieval-augmented generation.