Small Models, Big Insights: Leveraging Slim Proxy Models To Decide When and What to Retrieve for LLMs

Anonymous ACL submission

Abstract

The integration of large language models (LLMs) and search engines represents a significant evolution in knowledge acquisition methodologies. However, determining the knowledge that an LLM already possesses and the knowledge that requires the help of a search engine remains an unresolved issue. Most existing methods solve this problem through the results of preliminary answers or reasoning done by the LLM itself, but this incurs excessively high computational costs. This paper introduces a novel collaborative approach, namely SlimPLM, that detects missing knowledge in LLMs with a slim proxy model, to enhance the LLM's knowledge acquisition process. We employ a proxy model which has far fewer parameters, and take its answers as heuristic answers. Heuristic answers are then utilized to predict the knowledge required to answer the user question, as well as the known and unknown knowledge within the LLM. We only conduct retrieval for the missing knowledge in questions that the LLM does not know. Extensive experimental results on five datasets with two LLMs demonstrate a notable improvement in the end-to-end performance of LLMs in question-answering tasks, achieving or surpassing current state-of-the-art models with lower LLM inference costs.[1]

1 Introduction

Large language models (LLMs) have demonstrated significant prowess in various natural language processing (NLP) tasks (Achiam et al., 2023), attributed to their advanced language comprehension and generation capabilities. Despite being trained on extensive text corpora, these models occasionally produce hallucinated content (Zhou et al., 2020; Maynez et al., 2020). To tackle this problem, the integration of retrieval systems with LLMs has been proposed, enabling access to external knowledge bases for more accurate and reliable text generation.

Retrieval-augmented generation (RAG) involves using a retrieval system to supplement LLMs with relevant external information, thereby improving text generation quality (Peng et al., 2023; He et al., 2022). Yet, recent studies have suggested that retrieval may not always be beneficial. In cases where LLMs can adequately respond without external knowledge, retrieval may introduce irrelevant information, potentially degrading performance (Kadavath et al., 2022; Wang et al., 2023b; Shi et al., 2023a; Petroni et al., 2020a). Therefore, it is critical to determine when retrieval is necessary for user questions (Shuster et al., 2021). The challenge lies in identifying questions that exceed the LLMs' intrinsic knowledge and require external retrieval, due to the prevalence of content hallucination. Efforts to address this challenge can be categorized into two groups: (1) The first group of methods involves fine-tuning LLMs for RAG scenarios, allowing them to autonomously signal the need for external knowledge (Nakano et al., 2021; Liu et al., 2023b; Qin et al., 2023b). This method, while effective, demands substantial computational resources and risks diminishing the LLMs' general capabilities due to potential catastrophic forgetting (Kotha et al., 2023; Zhai et al., 2023). (2) The second category avoids direct tuning of LLMs, assessing the necessity for retrieval based on the quality of the generated content or specific indicators within it (Ram et al., 2023; Min et al., 2022). However, this approach still has its drawbacks, as it requires multiple inferences, thereby increasing both the inference costs and the latency of responses to user questions.

In light of this, we put forward a question: *Is it feasible to employ a proxy model with a relatively smaller parameter size to facilitate effective retrieval results for an LLM?* Theoretically, existing

[1] Anonymous code and datasets are available at https://anonymous.4open.science/r/SlimPlm-2434

decoder-only language models share similar Transformer structures, and they are pre-trained on some common text corpora, such as Common Crawl web pages, books, and Wikipedia pages (Touvron et al., 2023; Bai et al., 2023; Scao et al., 2022; Almazrouei et al., 2023; Zhang et al., 2024). Therefore, it is possible for them to reach a consensus on relative mastery over different knowledge and the necessity of retrieval. Our preliminary quantitative analysis, shown in Section 4.6, also supports this hypothesis. The experimental results show that on questions well understood by the LLM, the relatively smaller language model also has considerable knowledge. The gap between larger and smaller LLMs mainly manifests in questions they do not understand. This further validates the possibility of employing a proxy model to help determine the necessity of retrieval.

Based on our analysis, in this paper, we introduce a novel approach, called **SlimPLM** (**Slim** **P**roxy **L**anguage **M**odel), which leverages a relatively smaller language model as a "proxy model" to help determine when and how to perform retrieval for LLMs. Specifically, for a user question, SlimPLM first uses the proxy model to generate a preliminary "heuristic answer". This heuristic answer serves two purposes. First, it is evaluated by a lightweight model designed to assess the necessity for retrieval. If this evaluation shows that the heuristic answer is of high quality, it implies that the question may be addressed directly by LLMs without additional information retrieval. In contrast, a lower-quality answer triggers the retrieval process to identify and supplement missing knowledge. To facilitate this, SlimPLM utilizes the heuristic answer again to generate multiple queries, each reflecting a specific aspect of the initial response. These queries are then individually assessed for their need for retrieval, filtering out queries that do not require retrieval. By this means, the remaining queries can retrieve more relevant knowledge that is lacking in LLMs. The integration of SlimPLM into existing RAG frameworks offers a flexible and effective enhancement without notably increasing computational costs or response latency. Experimental results across five commonly used question-answering datasets validate SlimPLM's effectiveness in determining the necessity for retrieval and improving retrieval results.

Our contributions are threefold: (1) We propose a novel approach that leverages a small proxy model to generate heuristic answers, helping determine when and how to perform retrieval for LLMs. (2) We devise a retrieval necessity judgment model based on the heuristic answer. It is capable of accurately identifying which queries necessitate further information retrieval. (3) We formulate a query rewriting strategy that decomposes the heuristic answer into distinct claims. This is complemented by a claim-based filtering mechanism to enhance the relevance of the retrieval results for LLMs' text generation.

2 Related Work

2.1 Retrieval-augmented Generation (RAG)

Before the rise of LLMs, small language models (Raffel et al., 2019; Lewis et al., 2019) possess limited knowledge, so retrieval-augmentation is used to assist knowledge-incentive QA tasks. Previous works often design a fixed module for retrieval-augmentation (Sun et al., 2022; Lewis et al., 2020; Izacard and Grave, 2020). Since the release of ChatGPT, LLMs have been developing rapidly (Zhao et al., 2023). LLMs possess abundant intrinsic knowledge, but hallucination issue means they also have to rely on retrieval-augmentation to improve factuality (Shuster et al., 2021; White, 2023).

Since LLMs have a rather fixed architecture, retrieval-augmented LLMs generally leave LLMs' architecture unchanged while pre-pending grounding documents to the input (Ram et al., 2023; Yu et al., 2023b). Generally, a typical RAG pipeline (Zhu et al., 2023; Liu et al., 2023a; Shi et al., 2023b) consists of a query rewriter (Wang et al., 2023a; Gao et al., 2022), a retriever (Guu et al., 2020; Neelakantan et al., 2022), a filter or reranker (Yoran et al., 2023; Yu et al., 2023a; Xu et al., 2023), and an LLM as reader. Some methods attempt to enhance preposition components for overall performance improvement. Others fine-tune the LLM to fit into RAG tasks (Asai et al., 2023; Kadavath et al., 2022).

2.2 Retrieval Necessity Judgment

Except for modifications to RAG pipeline components, retrieval necessity judgment is a more closely related topic. There are several streams to solve this issue.

(1) Fine-tuning LLMs is effective, but incurs significant computational costs (Qin et al., 2023a; Lin et al., 2022). Some work fine-tune the LLM to imitate a human-like web-browsing behavior (Schick

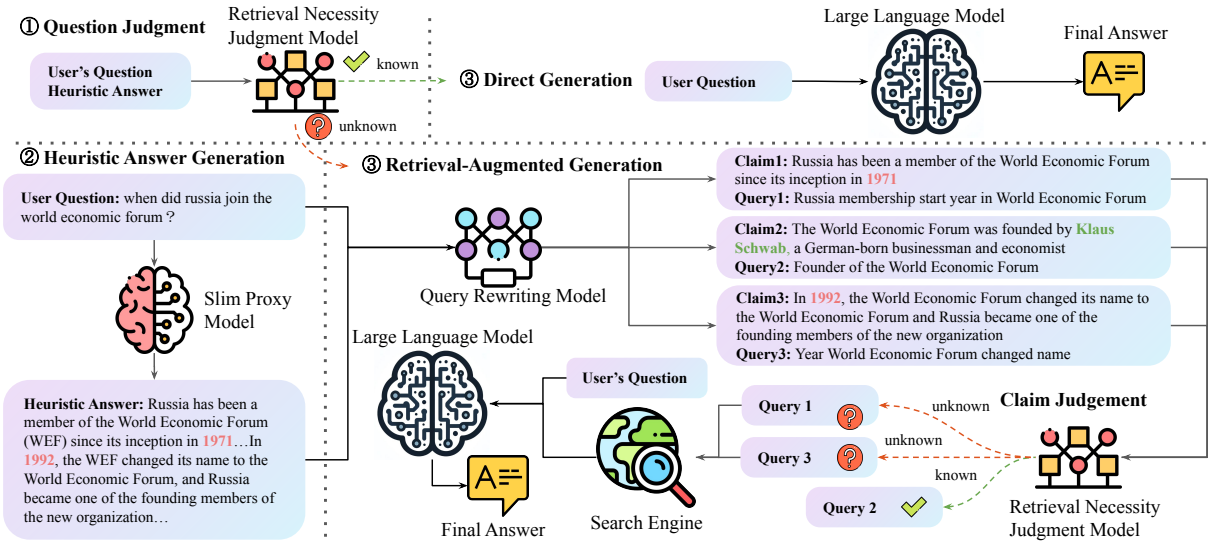

Figure 1: A display of the main process of SlimPLM. Solid lines with arrows represent the flow of data, while dashed lines with arrows signify control signals from the retrieval necessity judgment model. Step 1 and step 2 are mandatory in the pipeline, but step 3 involves choosing between direct generation and RAG.

et al., 2023; Nakano et al., 2021). Self-RAG (Asai et al., 2023) introduces special tokens called reflection tokens to control retrieval behavior.

(2) Another intuitive approach involves assessing the LLM's confidence based on logits produced by the model (Jiang et al., 2020; Guo et al., 2017). FLARE (Jiang et al., 2023) dynamically triggers RAG if logits fall below a certain threshold.

(3) Other work use iterative prompts to decide if extra information is required (Wei et al., 2022; Liu et al., 2021; Rubin et al., 2021), or synergy Chain-of-Thought prompting (Wei et al., 2022) with RAG (Press et al., 2022; Khattab et al., 2022). For example, ReAct (Yao et al., 2022) alternates between generating thoughts and actions, creating a series of thought-action-observation steps.

(4) Assessing the difficulty or popularity of user questions to judge retrieval necessity is also a viable approach (Mallen et al., 2022). SKR (Wang et al., 2023b) refers to similar questions they have previously encountered to judge retrieval necessity.

By contrast, SlimPLM conducts retrieval necessity judgment by assessing the answer generated by a smaller LLM. SlimPLM does not increase LLM inference times while improving judgment accuracy.

2.3 Query Formulation

Many methods conduct retrieval necessity judgment and also involve query formulation. FLARE (Jiang et al., 2023) masks the low logits tokens in the LLM output sequence and concatenates the rest to formulate the query. ReAct (Yao et al., 2022) and Self-Ask (Press et al., 2022) prompt the LLM to generate queries itself. By contrast, SlimPLM conducts query formulation by scrutinizing the answer generated by a smaller LLM providing an accurate perception of the necessary knowledge.

3 Methodology

In this paper, we aim to leverage a relatively smaller model as the *proxy model* to determine whether the user-issued question requires supplementary retrieval results and further provide clues for retrieving relevant knowledge. Our method, SlimPLM, can be flexibly used as a plug-in to various retrieval-augmented generation scenarios, without additional training requirements. The illustration of our method is shown in Figure 1.

3.1 Problem Formulation & Framework

Before diving into the details of our method, we first formulate the concept and notations involved in this paper.

Given a user input x and a text corpus (*e.g.*, a Wikipedia dump) $D = \{d_i\}_{i=1}^{N}$ of size N, models are expected to generate the annotated answer y. To obtain the information in D that is relevant to x, a retriever (R) is employed. This retriever takes a query q as input and returns a relevant text list $D_{\text{ref}} = R(q)$. Typically, the user input x is used as the query, namely $q = x$. However, existing studies have demonstrated that using refined queries

for retrieval can improve the final generation quality (Gao et al., 2022; Wang et al., 2023a). Therefore, we denote the refined queries as $\{q_1, \ldots, q_n\}$. With these refined queries, a collection of relevant retrieval results $D_{\text{ref}} = R(q_1), \ldots, R(q_n)$ is assembled to support the generation process, formalized as $\hat{y} = \text{LLM}(D_{\text{ref}}, x)$. Note that, when $D_{\text{ref}} = \varnothing$, the process degenerates to normal generation without retrieval.

We define a proxy model (PM), which is implemented by a relatively smaller LLM. The proxy model generates an answer for the input x as:

$$\hat{a} = \text{PM}(x), \tag{1}$$

where $\hat{a}$ is called a *heuristic answer* in this paper. This heuristic answer serves two purposes: (1) It is used for determining whether the retrieval is necessary for the current input x. The determination is made by a retrieval necessity judgment model (introduced in Section 3.2). (2) It also provides clues for query rewriting. The query rewriting results will help identify knowledge gaps within the LLM that necessitate further retrieval (introduced in Section 3.3).

3.2 Retrieval Necessity Judgment

Because existing LLMs are typically trained on common corpora (such as CommonCrawl and Wikipedia (Touvron et al., 2023; Penedo et al., 2023; Gao et al., 2020)) and employ a similar Transformer decoder-based architecture, it is promising to leverage a smaller LLM for judging the knowledge mastered by larger LLMs and determining the need for additional retrieval. Thus, we propose a retrieval necessity judgment component.

Judgment Model Given the heuristic answer $\hat{a}$ generated by the proxy model (Equation 1), we fine-tune a judgment model RJ (implemented by Llama2-7B in our experiments) by using both the user input x and the heuristic answer $\hat{a}$. We use the following instructions for fine-tuning:

```
Input:
<SYS> You are a helpful assistant. Your
task is to parse user input into structured
formats and accomplish the task according to
the heuristic answer. </SYS>
Heuristic answer: {Heuristic Answer}
Question: {user question}
Retrieval Necessity Judgment Output:
Output:
Known (True / False)
```

After fine-tuning, the RJ model can predict whether a user question needs further retrieval with the help of the heuristic answer.

Judgment Label Collection To fine-tune the RJ model, we need to collect training samples with reliable labels. Existing studies (Wang et al., 2023b) have proposed an annotation strategy that compares the models' outputs generated with and without retrieval. In our preliminary study, we find that this strategy is highly influenced by the capability of the retriever and the completeness of the corpus, leading to annotations that cannot accurately reflect the model's necessity for search. To tackle this problem, we propose to leverage the quality of our heuristic answers, *i.e.*, if the quality of the heuristic answer is higher than a predefined threshold, we infer that the question can be well answered without retrieval; otherwise, we consider retrieval necessary.

Specifically, we collect samples with short answers from existing question-answering datasets and employ the matching ratio between the heuristic answers and the ground-truth answers as the metric. Compared to rouge scores (Lin, 2004) or perplexity (Huyen, 2019), this metric can better align with the evaluation and reflect the generation quality. Notably, while we only use short answers for label collection, the obtained model can well generalize to different datasets, such as long-form QA datasets. Formally, for a question with multiple short answers $Y = \{y_1, y_2, \ldots, y_n\}$, we compute the matching ratio r between $\hat{a}$ and Y as:

$$r = \frac{|\{y \mid y \in \hat{a} \wedge y \in Y\}|}{|Y|}. \tag{2}$$

Then, we set a threshold θ and obtain the label as:

$$\text{Label}(\hat{a}, x) = \begin{cases} \text{Known (True)}, & \text{if } r > \theta; \\ \text{Known (False)}, & \text{otherwise.} \end{cases}$$

3.3 Retrieval Target Determination

After determining the necessity of retrieval, the next question is how to perform effective retrieval. A straightforward method is using user input x as the query to retrieve relevant information from the corpora D. However, many studies have reported that the information retrieved by x may lose details and introduce redundant content (Wang et al., 2023a). To address this issue, we propose a query rewrite method based on the heuristic answers and a query filter method to refine these rewritten queries.

Heuristic Answer-Driven Query Rewrite Restricted by parameter scale, the proxy model often hallucinates during the process of answering questions, but the direction in which they answer questions is heuristic (Dhuliawala et al., 2023; Gao et al., 2022). They can extend related aspects and sub-topics of thought when analyzing questions. Inspired by claim decomposition operation intended for factual evaluation (Min et al., 2023; Kryscinski et al., 2019), we perform query rewriting based on each fact mentioned in the heuristic answer given by the proxy models. The specific operations are as follows: we decompose the heuristic answer $\hat{a}$ into multiple claims related to the question, $\{c_1, c_2, \ldots, c_n\}$, where each claim related to the question can lead to a query, $\{q_{c_1}, q_{c_2}, \ldots, q_{c_n}\}$. In addition, we combine the query rewrites directly derived from the user's input $\{q_{x_1}, q_{x_2}, \ldots, q_{x_n}\}$. Our query rewriting model QR takes the user question and the heuristic answer as input and outputs all query rewrite results, $\text{QR}(x, \hat{a}) = \{q_{x_1}, \ldots, q_{x_n}, q_{c_1}, \ldots, q_{c_n}\}$.

To train the query rewriting model, we collect and annotate a dataset with the help of GPT-4(Achiam et al., 2023). In each dataset used in our experiments, we sample 1,000 user questions. We utilize the method of instruction fine-tuning (Ouyang et al., 2022; Chung et al., 2022) to fine-tune a decoder-only generative model, accomplishing the task of claim extraction and query rewriting in a single round. Our instructions and the model output are displayed as follows.

```
Input:
<SYS> You are a helpful assistant. Your
task is to parse user input into structured
formats and accomplish the task according to
the heuristic answer. </SYS>
Heuristic answer: {Heuristic Answer}
Question: {User Question}
Query Rewrite Output:
Output:
<Claim> Claim 1 <Query> Query 1 <Claim> Claim
2 <Query> Query 2, ...
```

Claim-based Query Filter In the previous step, our method generates several rewritten queries $\text{QR}(x, \hat{a})$, which correspond to the claims in the heuristic answers.

To achieve this, we reuse the judgment model RJ trained in Section 3.2. Specifically, we replace the input of the heuristic answer by the extracted claim and the input of user questions by the rewritten query. Then, the model can predict whether the rewritten query requires external knowledge from retrieval. We only perform retrieval when the result is Known (False), namely, we have:

$$D_{\text{ref}} = \{R(q_{c_i}) | \text{RJ}(c_i, q_{c_i}) = \text{Known (False)}\}.$$

By this means, we can obtain the retrieved result set D_{ref} that only contains the knowledge missing by the LLM.

4 Experiments

We conduct experiments on five widely used question-answering (QA) datasets and compare the performance of our method with several baselines.

4.1 Datasets

We use the following five QA datasets: (1) Natural Questions (NQ) (Kwiatkowski et al., 2019): a dataset consisting of real user questions from Google search. (2) Trivia-QA (Joshi et al., 2017): a realistic text-based question answering dataset. (3) ASQA (Stelmakh et al., 2022): a dataset targeting ambiguous questions requiring answers that integrate factual information from various sources. (4) MuSiQue (Trivedi et al., 2021): a synthetic multi-hop question-answering dataset. (5) ELI5 (Fan et al., 2019): a long-form question answering dataset originated from the Reddit forum. Due to our limited resources, we randomly sample 400 questions from the test set (if any) or validation set of each dataset as the test set for evaluation.

4.2 Evaluation Metrics

For all QA tasks, LLMs can freely generate any answers. For datasets annotated with long-form answers, we employ the Rouge Score (Lin, 2004) (ROUGE) to evaluate the quality of the generated answers by comparing them with the ground-truth ones. For datasets with short answers, we use the Exact Match (EM) metric to compare the generated answer with the golden one. If the dataset provides multiple optional short answers, we also report the proportion of instances where at least one short answer matches (Hit@1).

4.3 Baselines

We first select two baselines without retrieval:

(1) **Vanilla Chat**: This method directly inputs the user question into LLMs to get the answer. (2) **CoT Prompting** (Wei et al., 2022): This method introduces a prompt method that lets LLMs think step-by-step to derive the final answer.

Method	#API	ASQA		NQ		Trivia-QA		MuSiQue	ELI5		
		EM	Hit@1	EM	Hit@1	EM	Hit@1	EM	ROUGE-1	ROUGE-2	ROUGE-L
Llama2-70B-Chat without Retrieval											
Vanilla	1	29.68	62.50	40.49	55.00	27.44	90.75	11.50	28.66	4.88	14.27
CoT	1	26.21	54.50	35.36	48.75	23.50	79.00	11.50	28.12	4.73	14.06
Llama2-70B-Chat with Retrieval											
Direct	1	27.63	58.00	42.40	56.00	28.07	92.25	10.50	28.61	4.76	**15.76**
FLARE	2.10	30.08	63.50	41.36	55.75	27.41	89.50	11.25	27.95	4.72	13.91
Self-Eval	2	29.45	60.75	42.15	55.75	27.58	91.50	10.25	28.70	4.83	15.39
Self-Ask	2.67	26.37	60.25	38.56	53.00	26.56	89.50	9.50	-	-	-
ITER-RETGEN	3	30.15	60.50	42.85	55.50	28.31	91.00	13.00	28.44	4.74	15.72
SKR-KNN	1	29.38	61.75	41.90	55.75	28.16	**92.25**	10.25	28.71	4.80	15.73
SlimPLM (Ours)	1	**30.73**	**65.00**	**47.43**	**62.25**	**28.35**	92.00	**13.00**	**29.97**	**5.61**	15.13
Qwen-72B-Chat without Retrieval											
Vanilla	1	26.65	58.50	40.38	53.75	27.82	90.25	11.75	**30.61**	5.21	15.90
CoT	1	27.74	59.50	40.49	53.75	27.62	91.75	12.75	29.94	4.94	14.75
Qwen-72B-Chat with Retrieval											
Direct	1	25.85	57.00	41.27	52.75	26.39	87.75	7.75	25.93	4.55	**16.74**
FLARE	2.29	27.68	59.00	40.89	54.50	27.10	88.50	**12.75**	30.31	5.20	15.77
Self-Eval	2	27.64	60.00	42.43	56.00	27.13	90.50	7.75	29.19	5.14	16.05
Self-Ask	2.76	22.82	52.25	36.16	49.25	25.29	87.50	9.75	-	-	-
ITER-RETGEN	3	**29.38**	61.50	43.51	57.50	27.16	89.75	12.25	26.15	4.41	16.52
SKR-KNN	1	28.08	61.50	43.08	56.00	26.38	88.50	11.25	27.29	4.75	16.31
SlimPLM (Ours)	1	27.97	**62.25**	**44.07**	**57.75**	**28.03**	**92.25**	9.75	29.56	**5.91**	16.36

Table 1: Evaluation results of SlimPLM and baselines on five QA benchmarks. #API is the average LLM inference times. Hit@1 is the proportion of instances where at least one short answer matches.

We also consider several retrieval-augmented generation methods. They differ in time and approach for retrieval necessity judgment and construction of retrieval queries. We include more baseline implementation details in Appendix B.

(1) **Direct Search**: This approach uses search for all questions and directly utilizes the user question as the search query. (2) **FLARE** (Jiang et al., 2023): This method examines the content of each sentence generated by the LLM, and uses retrieval if the generation logits are below a threshold. FLARE uses the masked sentence as a query, wherein tokens associated with low logits are masked. (3) **Self-Eval** (Kadavath et al., 2022): This method uses prompts and few-shot learning to let LLM itself decide whether it needs retrieval or not. (4) **Self-Ask** (Press et al., 2022): This method iteratively prompts the LLM to decide whether to generate follow-up questions as queries or generate the final answer directly. (5) **SKR-KNN** (Wang et al., 2023b). It uses a dense retriever to retrieve top-k nearest neighbor questions from the training set. The necessity of retrieval is determined by the number of neighboring questions that require or do not require retrieval.

4.4 Implementation Details

We conduct experiments on two open-source LLMs, Llama2-70B-Chat (Touvron et al., 2023) and Qwen-72b-Chat (Bai et al., 2023). The default proxy model, fine-tuned query rewriting model, and retrieval necessity judgment model are built on Llama2-7B-Chat. We build a search engine on the KILT dataset's document library, which is based on the 2019 Wikipedia mirror (Petroni et al., 2020b). BM25 (Robertson and Zaragoza, 2009) is used as the retriever and $E5_{base}$ (Wang et al., 2022) is employed as the reranker. More implementation details are provided in Appendix A.

4.5 Experimental Results

The evaluation results are shown in Table 1, where we uniformly chose Llama2-7B-Chat as the proxy model, a fine-tuned query rewriting model, and a fine-tuned retrieval necessity judgment model. Generally, our SlimPLM achieves superior or competitive performance on all datasets. This clearly demonstrates the effectiveness of our method. Besides, we have the following observations:

(1) On most datasets, retrieval-augmented generation methods can outperform the methods with-

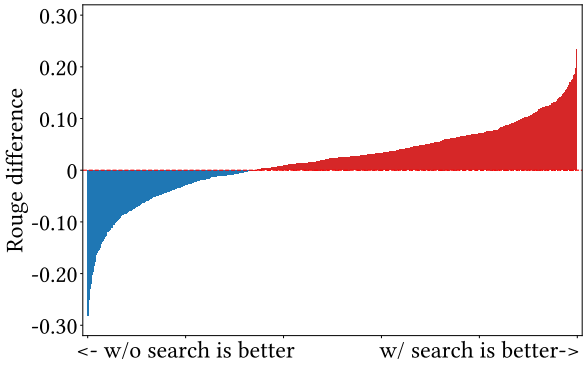

Figure 2: The rouge performance for with and without retrieval on ELI5 dataset

Figure 3: The proportion of samples (y-axis) with EM scores higher than certain values (x-axis).

out using retrieval. This clearly demonstrates the benefit of incorporating external knowledge into open-domain QA tasks. (2) Compared to methods that initiate retrieval based on the results or logits generated by LLMs (*i.e.*, Self-Eval, Self-Ask, and FALRE), our method yields better results. This validates the superiority of our method, which employs a proxy model to determine when and what the LLM needs to retrieve. Notably, our method requires the LLM to infer only once, significantly reducing the cost of inference. (3) Comparing methods that judge retrieval necessity merely based on user questions (SKR-KNN), our method also has advantages. By using heuristic answers, it can more accurately assess the LLM's knowledge capability and formulate queries that are more precisely tailored to the question, thereby improving overall performance. (4) Intriguingly, we notice that retrieval does not uniformly benefit all user questions. For example, in the ELI5 dataset, approximately 66.4% of samples show improvement with retrieval, as shown in Figure 2. This observation highlights the critical need to judge the necessity of retrieval. More cases where retrieval has negative impact are shown in Appendix C.

4.6 Further Analysis

We further conduct a series of experiments to investigate the impact of different settings in our method.

Ablation Study We first examine the effectiveness of different modules in our method by an ablation study. This experiment is conducted by removing the heuristic-answer-driven query rewriting (*w/o* QR), question-Level retrieval necessity judgment (*w/o* RJ), and Claim-based Query Filter (*w/o* QF), respectively. From the results are shown in Table 2, we can see: (1) If query rewriting is removed, then retrieval necessity judgment between vanilla chat and direct search is applied. Performing query rewriting can both enhance the comprehensiveness and relevance of retrieved references. (2) When retrieval necessity judgment is removed, all questions will use retrieval results for generation. LLMs will be led astray on questions that they can perform well on their own knowledge. (3) If claim-based query filter is removed, then retrieval is applied to every query derived from the heuristic answer. Not filtering queries which contain contents that do not require retrieval worsens the search results.

Knowledge Ability Consensus between Proxy Models and LLMs In this experiment, we compared the knowledge capabilities of LLMs and proxy models, and confirmed their consensus. Our findings can be summarized as follows: (1) The difference in capabilities between the proxy model and the LLM is primarily manifested in the knowledge of lower mastery levels. As illustrated in Figure 3, on the ASQA dataset, the difference between the 70B and 7B language models is very slight for samples with an EM score greater than 0.5. Their differences are primarily evident in samples with an EM score less than 0.5. (2) The higher the level of some knowledge mastered by the proxy model, the higher the level of mastery by the LLM. Further experiments on ASQA shows over 82.19% of the samples with an EM score greater than 0.5 for the 7B model overlaps with those of the 70B model.

The experimental results above offer a theoretical basis for our method. If the proxy model can correctly answer the question, then the LLM is very likely to answer it correctly as well. Applying vanilla chat for them can better leverage the inherent knowledge capabilities of LLMs.

Method	ASQA		NQ		Trivia-QA		MuSiQue
	EM	Hit@1	EM	Hit@1	EM	Hit@1	EM
SlimPLM	30.73	65.00	47.43	62.25	28.35	92.00	13.00
w/o QR	29.19 (5.0%↓)	61.75 (5.0%↓)	45.16 (4.8%↓)	59.75 (4.0%↓)	28.07 (1.0%↓)	92.50 (0.5%↑)	11.50 (1.2%↓)
w/o QJ	29.43 (4.2%↓)	61.75 (5.0%↓)	43.03 (9.3%↓)	57.25 (8.0%↓)	27.91 (1.6%↓)	90.25 (1.9%↓)	12.75 (1.9%↓)
w/o QF	30.73 (0.0%)	64.75 (0.4%↓)	46.62 (1.7%↓)	61.25 (1.6%↓)	28.27 (0.3%↓)	91.75 (0.3%↓)	12.50 (3.9%↓)

Table 2: Ablation study on Llama2-70B-Chat. "QR", "QJ", and "QF" denote the query rewriting, question-level retrieval necessity judgment, and claim-based query filter, respectively.

Method	ASQA		NQ		Trivia-QA		MuSiQue	ELI5		
	EM	Hit@1	EM	Hit@1	EM	Hit@1	EM	ROUGE-1	ROUGE-2	ROUGE-L
Llama2-70B-Chat										
Llama2-7B-Chat	**30.73**	**65.00**	**47.43**	**62.25**	**28.35**	**92.75**	13.00	29.97	5.61	15.13
Qwen-7B-Chat	29.62	60.25	42.53	56.25	27.93	92.25	13.00	29.95	5.57	**16.16**
TinyLlama-1.1B-Chat	30.47	60.50	44.24	56.75	28.02	91.00	11.50	30.05	5.56	15.37
Phi-2 (2.7B)	28.96	60.50	43.33	57.50	27.99	91.50	**13.75**	**30.34**	**5.82**	15.48
Qwen-72B-Chat										
Llama2-7B-Chat	**27.97**	**62.25**	**44.07**	**57.75**	**28.03**	**92.25**	9.75	**29.56**	**5.91**	**16.36**
Qwen-7B-Chat	27.76	59.75	42.54	55.75	27.22	90.25	8.75	29.44	5.74	16.33
TinyLlama-1.1B-Chat	27.61	58.25	42.36	55.25	27.67	91.25	9.25	28.80	5.64	16.12
Phi-2 (2.7B)	26.95	59.50	42.22	54.25	27.10	89.00	**10.75**	29.17	5.83	16.32

Table 3: Performance Comparison of Various Proxy Methods.

Dataset	Chat	Proxy	Rewrite	Judge	Total
ASQA	192.86	24.42	35.27	3.38	63.07
NQ	249.74	29.61	38.65	3.65	71.91
TQA	114.07	15.73	28.13	2.47	46.33
MuSiQ	168.47	19.00	30.84	2.36	52.20
ELI5	471.22	47.82	46.82	4.23	98.87

Table 4: The number of tokens used by LLM, and the additional tokens brought by SlimPLM.

Impact of Various Proxy Models We also explore the impact of using different proxy models in our method. This experiment is conducted by using four open-source LLMs with different sizes as the proxy model, including Llama2-7B-Chat (Touvron et al., 2023), Qwen-7B-Chat (Bai et al., 2023), TinyLlama-1.1B-Chat (Zhang et al., 2024), and Phi-2 (Li et al., 2023). Experimental results are shown in Table 3. We can see that in most datasets, Llama2-7B can provide the best results. Furthermore, Llama2-7B contributes a greater improvement to Llama2-70B than to Qwen-72B, we attribute this to the better knowledge alignment between Llama models.

Computational Cost Analysis In our method, we use a proxy model, a query rewriting model, and a retrieval necessity judgment model based on relatively smaller LLMs (7B). To investigate their computational efficiency, we analyze the average number of tokens generated by each model and calculate the associated costs. This calculation is based on the assumption that the computational expense per token for a 7B model is roughly 1/10 that of a 70B model——a conservative estimate, given that the actual cost differential is likely to exceed this ratio (Kaplan et al., 2020). Table 4 lists the additional computational costs required by each component and the total cost. The analysis reveals that the additional costs are substantially lower (1/4 to 1/3) compared to the costs of a single inference by an LLM. This observation validates the economic advantages of our method.

5 Conclusion

In conclusion, our research proposes a new paradigm for RAG, utilizing a smaller LLM as proxy model. Based on the heuristic answer by proxy model, we conduct query rewriting, retrieval necessity judgment, and claim-based query filtering. This approach enables accurate perception for when and what to retrieve for LLMs. Experiments across various datasets show a marked improvement in the end-to-end performance of LLM question-answering, achieving or exceeding state-of-the-art results. Moreover, this enhancement is attained with little additional computational cost.

Limitations

In scenarios where almost all user questions are primarily outside the scope of the LLM's pre-training corpus, or where almost all the questions do not require external knowledge, our method proves challenging to utilize. In these situations, opting either for a full retrieval or without retrieval at all may be a more suitable approach. Additionally, we acknowledge a gap in the knowledge capabilities between proxy models and LLMs. Heuristic answers are unable to fully reflect the true knowledge capability of the LLMs. Moreover, our current method employs three models: a proxy model, a query rewriting model, and a retrieval necessity judgment model. The pipeline appears somewhat complex; integrating these functions into a single generative framework would be preferable.

References

OpenAI Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mo Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Benjamin Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Sim'on Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Raphael Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Lukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Hendrik Kirchner, Jamie Ryan Kiros, Matthew Knight, Daniel Kokotajlo, Lukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Adeola Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel P. Mossing, Tong Mu, Mira Murati, Oleg Murk, David M'ely, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Ouyang Long, Cullen O'Keefe, Jakub W. Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alexandre Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Pondé de Oliveira Pinto, Michael Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario D. Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin D. Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas A. Tezak, Madeleine Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cer'on Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2023. Gpt-4 technical report.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra-Aimée Cojocaru, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. The falcon series of open language models. *ArXiv*, abs/2311.16867.

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.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang,

Xiaodong Deng, Yang Fan, Wenhang Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, K. Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Jian Yang, Shusheng Yang, Shusheng Yang, Bowen Yu, Yu Bowen, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xing Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report. *ArXiv*, abs/2309.16609.

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.

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. *ArXiv*, abs/2309.11495.

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

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The pile: An 800gb dataset of diverse text for language modeling. *ArXiv*, abs/2101.00027.

Luyu Gao, Xueguang Ma, Jimmy J. Lin, and Jamie Callan. 2022. Precise zero-shot dense retrieval without relevance labels. *ArXiv*, abs/2212.10496.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *International Conference on Machine Learning*.

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

Hangfeng He, Hongming Zhang, and Dan Roth. 2022. Rethinking with retrieval: Faithful large language model inference. *ArXiv*, abs/2301.00303.

Chip Huyen. 2019. Evaluation metrics for language modeling. *The Gradient*.

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

Zhengbao Jiang, J. Araki, Haibo Ding, and Graham Neubig. 2020. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977.

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.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *ArXiv*, abs/1705.03551.

Saurav Kadavath, Tom Conerly, Amanda Askell, T. J. Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zachary Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, John Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom B. Brown, Jack Clark, Nicholas Joseph, Benjamin Mann, Sam McCandlish, Christopher Olah, and Jared Kaplan. 2022. Language models (mostly) know what they know. *ArXiv*, abs/2207.05221.

Jared Kaplan, Sam McCandlish, T. J. Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeff Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *ArXiv*, abs/2001.08361.

O. Khattab, Keshav Santhanam, Xiang Lisa Li, David Leo Wright Hall, Percy Liang, Christopher Potts, and Matei A. Zaharia. 2022. Demonstrate-search-predict: Composing retrieval and language models for knowledge-intensive nlp. *ArXiv*, abs/2212.14024.

Suhas Kotha, Jacob M. Springer, and Aditi Raghunathan. 2023. Understanding catastrophic forgetting in language models via implicit inference. *ArXiv*, abs/2309.10105.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Evaluating the factual consistency of abstractive text summarization. In *Conference on Empirical Methods in Natural Language Processing*.

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.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdel rahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Annual Meeting of the Association for Computational Linguistics*.

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.

Yuan-Fang Li, Sébastien Bubeck, Ronen Eldan, Allison Del Giorno, Suriya Gunasekar, and Yin Tat Lee. 2023. Textbooks are all you need ii: phi-1.5 technical report. *ArXiv*, abs/2309.05463.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Annual Meeting of the Association for Computational Linguistics*.

Stephanie C. Lin, Jacob Hilton, and Owain Evans. 2022. Teaching models to express their uncertainty in words. *Trans. Mach. Learn. Res.*, 2022.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2021. What makes good in-context examples for gpt-3? In *Workshop on Knowledge Extraction and Integration for Deep Learning Architectures; Deep Learning Inside Out*.

Jiongnan Liu, Jiajie Jin, Zihan Wang, Jiehan Cheng, Zhicheng Dou, and Ji rong Wen. 2023a. Reta-llm: A retrieval-augmented large language model toolkit. *ArXiv*, abs/2306.05212.

Xiao Liu, Hanyu Lai, Hao Yu, Yifan Xu, Aohan Zeng, Zhengxiao Du, P. Zhang, Yuxiao Dong, and Jie Tang. 2023b. Webglm: Towards an efficient web-enhanced question answering system with human preferences. *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Alex 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*.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan T. McDonald. 2020. On faithfulness and factuality in abstractive summarization. *ArXiv*, abs/2005.00661.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. *ArXiv*, abs/2305.14251.

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

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Ouyang Long, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. 2021. Webgpt: Browser-assisted question-answering with human feedback. *ArXiv*, abs/2112.09332.

Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas A. Tezak, Jong Wook Kim, Chris Hallacy, Johannes Heidecke, Pranav Shyam, Boris Power, Tyna Eloundou Nekoul, Girish Sastry, Gretchen Krueger, David P. Schnurr, Felipe Petroski Such, Kenny Sai-Kin Hsu, Madeleine Thompson, Tabarak Khan, Toki Sherbakov, Joanne Jang, Peter Welinder, and Lilian Weng. 2022. Text and code embeddings by contrastive pre-training. *ArXiv*, abs/2201.10005.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke E. Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Francis Christiano, Jan Leike, and Ryan J. Lowe. 2022. Training language models to follow instructions with human feedback. *ArXiv*, abs/2203.02155.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra-Aimée Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for falcon llm: Outperforming curated corpora with web data, and web data only. *ArXiv*, abs/2306.01116.

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Lidén, Zhou Yu, Weizhu Chen, and Jianfeng Gao. 2023. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *ArXiv*, abs/2302.12813.

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*.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. 2022. Measuring and narrowing the compositionality gap in language models. *ArXiv*, abs/2210.03350.

Yujia Qin, Zihan Cai, Di Jin, Lan Yan, Shi Liang, Kunlun Zhu, Yankai Lin, Xu Han, Ning Ding, Huadong Wang, Ruobing Xie, Fanchao Qi, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 2023a. Webcpm: Interactive web search for chinese long-form question answering. In *Annual Meeting of the Association for Computational Linguistics*.

Yujia Qin, Shengding Hu, Yankai Lin, Weize Chen, Ning Ding, Ganqu Cui, Zheni Zeng, Yufei Huang, Chaojun Xiao, Chi Han, Yi Ren Fung, Yusheng Su, Huadong Wang, Cheng Qian, Runchu Tian, Kunlun Zhu, Shi Liang, Xingyu Shen, Bokai Xu, Zhen Zhang, Yining Ye, Bo Li, Ziwei Tang, Jing Yi, Yu Zhu, Zhenning Dai, Lan Yan, Xin Cong, Ya-Ting Lu, Weilin Zhao, Yuxiang Huang, Jun-Han Yan, Xu Han, Xian Sun, Dahai Li, Jason Phang, Cheng Yang, Tongshuang Wu, Heng Ji, Zhiyuan Liu, and Maosong Sun. 2023b. Tool learning with foundation models. *ArXiv*, abs/2304.08354.

Colin Raffel, Noam M. Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331.

Stephen E. Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3:333–389.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2021. Learning to retrieve prompts for in-context learning. *ArXiv*, abs/2112.08633.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ili'c, Daniel Hesslow, Roman Castagn'e, Alexandra Sasha Luccioni, Franccois Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurenccon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa Etxabe, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris C. Emezue, Christopher Klamm, Colin Leong, Daniel Alexander van Strien, David Ifeoluwa Adelani, Dragomir R. Radev, Eduardo Gonz'alez Ponferrada, Efrat Levkovizh, Ethan Kim, Eyal Natan, Francesco De Toni, Gérard Dupont, Germán Kruszewski, Giada Pistilli, Hady ElSahar, Hamza Benyamina, Hieu Trung Tran, Ian Yu, Idris Abdulmumin, Isaac Johnson, Itziar Gonzalez-Dios, Javier de la Rosa, Jenny Chim, Jesse Dodge, Jian Zhu, Jonathan Chang, Jorg Frohberg, Josephine L. Tobing, Joydeep Bhattacharjee, Khalid Almubarak, Kimbo Chen, Kyle Lo, Leandro von Werra, Leon Weber, Long Phan, Loubna Ben Allal, Ludovic Tanguy, Manan Dey, Manuel Romero Muñoz, Maraim Masoud, María Grandury, Mario vSavsko, Max Huang, Maximin Coavoux, Mayank Singh, Mike Tian-Jian Jiang, Minh Chien Vu, Mohammad Ali Jauhar, Mustafa Ghaleb, Nishant Subramani, Nora Kassner, Nurulaqilla Khamis, Olivier Nguyen, Omar Espejel, Ona de Gibert, Paulo Villegas, Peter Henderson, Pierre Colombo, Priscilla Amuok, Quentin Lhoest, Rheza Harliman, Rishi Bommasani, Roberto L'opez, Rui Ribeiro, Salomey Osei, Sampo Pyysalo, Sebastian Nagel, Shamik Bose, Shamsuddeen Hassan Muhammad, Shanya Sharma Sharma, S. Longpre, So maieh Nikpoor, S. Silberberg, Suhas Pai, Sydney Zink, Tiago Timponi Torrent, Timo Schick, Tristan Thrush, Valentin Danchev, Vassilina Nikoulina, Veronika Laippala, Violette Lepercq, Vrinda Prabhu, Zaid Alyafeai, Zeerak Talat, Arun Raja, Benjamin Heinzerling, Chenglei Si, Elizabeth Salesky, Sabrina J. Mielke, Wilson Y. Lee, Abheesht Sharma, Andrea Santilli, Antoine Chaffin, Arnaud Stiegler, Debajyoti Datta, Eliza Szczechla, Gunjan Chhablani, Han Wang, Harshit Pandey, Hendrik Strobelt, Jason Alan Fries, Jos Rozen, Leo Gao, Lintang Sutawika, M Saiful Bari, Maged S. Al-Shaibani, Matteo Manica, Nihal V. Nayak, Ryan Teehan, Samuel Albanie, Sheng Shen, Srulik Ben-David, Stephen H. Bach, Taewoon Kim, Tali Bers, Thibault Févry, Trishala Neeraj, Urmish Thakker, Vikas Raunak, Xiang Tang, Zheng-Xin Yong, Zhiqing Sun, Shaked Brody, Y Uri, Hadar Tojarieh, Adam Roberts, Hyung Won Chung, Jaesung Tae, Jason Phang, Ofir Press, Conglong Li, Deepak Narayanan, Hatim Bourfoune, Jared Casper, Jeff Rasley, Max Ryabinin, Mayank Mishra, Minjia Zhang, Mohammad Shoeybi, Myriam Peyrounette, Nicolas Patry, Nouamane Tazi, Omar Sanseviero, Patrick von Platen, Pierre Cornette, Pierre Franccois Lavall'ee, Rémi Lacroix, Samyam Rajbhandari, Sanchit Gandhi, Shaden Smith, Stéphane Requena, Suraj Patil, Tim Dettmers, Ahmed Baruwa, Amanpreet Singh, Anastasia Cheveleva, Anne-Laure Ligozat, Arjun Subramonian, Aur'elie N'ev'eol, Charles Lovering, Daniel H Garrette, Deepak R. Tunuguntla, Ehud Reiter, Ekaterina Taktasheva, Ekaterina Voloshina, Eli Bogdanov, Genta Indra Winata, Hailey Schoelkopf, Jan-Christoph Kalo, Jekaterina Novikova, Jessica Zosa Forde, Xiangru Tang, Jungo Kasai, Ken Kawamura, Liam Hazan, Marine Carpuat, Miruna Clinciu, Najoung Kim, Newton Cheng, Oleg Serikov, Omer Antverg, Oskar van der Wal, Rui Zhang, Ruochen Zhang, Sebastian Gehrmann, Shachar Mirkin, S. Osher Pais, Tatiana Shavrina, Thomas Scialom, Tian Yun, Tomasz Limisiewicz, Verena Rieser, Vitaly Protasov, Vladislav Mikhailov, Yada Pruksachatkun, Yonatan Belinkov, Zachary Bamberger, Zdenvek Kasner, Zdeněk Kas-

ner, Amanda Pestana, Amir Feizpour, Ammar Khan, Amy Faranak, Ananda Santa Rosa Santos, Anthony Hevia, Antigona Unldreaj, Arash Aghagol, Arezoo Abdollahi, Aycha Tammour, Azadeh HajiHosseini, Bahareh Behroozi, Benjamin Ayoade Ajibade, Bharat Kumar Saxena, Carlos Muñoz Ferrandis, Danish Contractor, David M. Lansky, Davis David, Douwe Kiela, Duong Anh Nguyen, Edward Tan, Emi Baylor, Ezinwanne Ozoani, Fatim Tahirah Mirza, Frankline Ononiwu, Habib Rezanejad, H.A. Jones, Indrani Bhattacharya, Irene Solaiman, Irina Sedenko, Isar Nejadgholi, Jan Passmore, Joshua Seltzer, Julio Bonis Sanz, Karen Fort, Lívia Dutra, Mairon Samagaio, Maraim Elbadri, Margot Mieskes, Marissa Gerchick, Martha Akinlolu, Michael McKenna, Mike Qiu, Muhammed Ghauri, Mykola Burynok, Nafis Abrar, Nazneen Rajani, Nour Elkott, Nourhan Fahmy, Olanrewaju Samuel, Ran An, R. P. Kromann, Ryan Hao, Samira Alizadeh, Sarmad Shubber, Silas L. Wang, Sourav Roy, Sylvain Viguier, Thanh-Cong Le, Tobi Oyebade, Trieu Nguyen Hai Le, Yoyo Yang, Zachary Kyle Nguyen, Abhinav Ramesh Kashyap, Alfredo Palasciano, Alison Callahan, Anima Shukla, Antonio Miranda-Escalada, Ayush Kumar Singh, Benjamin Beilharz, Bo Wang, Caio Matheus Fonseca de Brito, Chenxi Zhou, Chirag Jain, Chuxin Xu, Clémentine Fourrier, Daniel Le'on Perin'an, Daniel Molano, Dian Yu, Enrique Manjavacas, Fabio Barth, Florian Fuhrimann, Gabriel Altay, Giyaseddin Bayrak, Gully Burns, Helena U. Vrabec, Iman I.B. Bello, Isha Dash, Ji Soo Kang, John Giorgi, Jonas Golde, Jose David Posada, Karthi Sivaraman, Lokesh Bulchandani, Lu Liu, Luisa Shinzato, Madeleine Hahn de Bykhovetz, Maiko Takeuchi, Marc Pàmies, María Andrea Castillo, Marianna Nezhurina, Mario Sanger, Matthias Samwald, Michael Cullan, Michael Weinberg, M Wolf, Mina Mihaljcic, Minna Liu, Moritz Freidank, Myungsun Kang, Natasha Seelam, Nathan Dahlberg, Nicholas Michio Broad, Nikolaus Muellner, Pascale Fung, Patricia Haller, R. Chandrasekhar, Renata Eisenberg, Robert Martin, Rodrigo L. Canalli, Rosaline Su, Ruisi Su, Samuel Cahyawijaya, Samuele Garda, Shlok S Deshmukh, Shubhanshu Mishra, Sid Kiblawi, Simon Ott, Sinee Sang-aroonsiri, Srishti Kumar, Stefan Schweter, Sushil Pratap Bharati, Tanmay Laud, Théo Gigant, Tomoya Kainuma, Wojciech Kusa, Yanis Labrak, Yashasvi Bajaj, Y. Venkatraman, Yifan Xu, Ying Xu, Yu Xu, Zhee Xao Tan, Zhongli Xie, Zifan Ye, Mathilde Bras, Younes Belkada, and Thomas Wolf. 2022. Bloom: A 176b-parameter open-access multilingual language model. *ArXiv*, abs/2211.05100.

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*, abs/2302.04761.

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. *ArXiv*, abs/2305.15294.

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.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. In *Conference on Empirical Methods in Natural Language Processing*.

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. 2022. Asqa: Factoid questions meet long-form answers. *ArXiv*, abs/2204.06092.

Zhiqing Sun, Xuezhi Wang, Yi Tay, Yiming Yang, and Denny Zhou. 2022. Recitation-augmented language models. *ArXiv*, abs/2210.01296.

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. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288.

H. Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2021. Musique: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *ArXiv*, abs/2212.03533.

Liang Wang, Nan Yang, and Furu Wei. 2023a. Query2doc: Query expansion with large language

models. In *Conference on Empirical Methods in Natural Language Processing*.

Yile Wang, Peng Li, Maosong Sun, and Yang Liu. 2023b. Self-knowledge guided retrieval augmentation for large language models. *ArXiv*, abs/2310.05002.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Huai hsin Chi, F. Xia, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903.

Ryen W. White. 2023. Navigating complex search tasks with ai copilots. *ArXiv*, abs/2311.01235.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. 2023. Recomp: Improving retrieval-augmented lms with compression and selective augmentation. *ArXiv*, abs/2310.04408.

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*, abs/2210.03629.

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

W. Yu, Hongming Zhang, Xiaoman Pan, Kaixin Ma, Hongwei Wang, and Dong Yu. 2023a. Chain-of-note: Enhancing robustness in retrieval-augmented language models. *ArXiv*, abs/2311.09210.

W. Yu, Zhihan Zhang, Zhenwen Liang, Meng Jiang, and Ashish Sabharwal. 2023b. Improving language models via plug-and-play retrieval feedback. *ArXiv*, abs/2305.14002.

Yuexiang Zhai, Shengbang Tong, Xiao Li, Mu Cai, Qing Qu, Yong Jae Lee, and Y. Ma. 2023. Investigating the catastrophic forgetting in multimodal large language models. *ArXiv*, abs/2309.10313.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024. Tinyllama: An open-source small language model. *ArXiv*, abs/2401.02385.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Z. Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jianyun Nie, and Ji rong Wen. 2023. A survey of large language models. *ArXiv*, abs/2303.18223.

Chunting Zhou, Jiatao Gu, Mona T. Diab, Paco Guzmán, Luke Zettlemoyer, and Marjan Ghazvininejad. 2020. Detecting hallucinated content in conditional neural sequence generation. *ArXiv*, abs/2011.02593.

Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Zhicheng Dou, and Ji rong Wen. 2023. Large language models for information retrieval: A survey. *ArXiv*, abs/2308.07107.

A SlimPLM Implementation Details

Model Fine-tuning Our query rewriting model and the retrieval necessity judgment model are both obtained by instruction fine-tuning from Llama2-7B-Chat. We find that models fine-tuned with data collected from datasets annotated with multiple short answers possess better generalization abilities. They can adapt to various tasks including ambiguous QA, natural questions, long-form QA, and rewritten queries. We collect 5000 samples each from the training sets of ASQA, Natural Questions, and Trivia-QA. Through rule-based filtering, we formed the fine-tuning data for the retrieval necessity judgment model, as shown in Table 5. Because the number of unknown samples significantly exceeds that of known samples, we downsample the unknown samples to make their proportions roughly equal. For the query rewriting model, we collecte 1000 samples each from ASQA, Natural Questions, Trivia-QA, MuSiQue, ELI5, and then use GPT-4 for auxiliary annotation. The prompt we use to induce GPT-4 annotation is displayed in Table 8.

RAG Prompts RAG prompts concatenate the reference document in front of the question for enhanced retrieval generation. For datasets annotated with short-form answers and long-form answers, we use different RAG prompts. This is because short-form QA requires the completeness of answers, while long-answer QA demands the fluency of answers. Prompts we use are demonstrated in Table 9. We apply the same prompt strategy across all baselines unless some methods have very strict requirements for prompts, such as Self-Ask (Press et al., 2022) and FLARE (Jiang et al., 2023).

B Baseline Implementation Details

For methods that require multiple rounds of large language model reasoning, we observe that three rounds of reasoning can already solve most of the problems in our dataset. Methods with an indefinite number of reasoning rounds (Self-Ask (Press et al., 2022), FLARE (Jiang et al., 2023)) mostly stop iterating after three rounds. Considering the limitations of computational resources, we set the maximum number of iterations to three rounds. We also set the iteration count of 3 for ITER-RETGEN (Shao et al., 2023).

The results of Self-Ask (Press et al., 2022) on the ELI5 dataset are not compared is that Self-Ask can only output short answers due to prompt limi-

Dataset	Known	Unknown	Dropped
ASQA	593	592	3168
NQ	1873	1874	1253
TQA	588	588	3824

Table 5: The number of unknown, known and dropped samples for retrieval necessity judgment model.

RAG Prompt for FLARE
Search results:
[1] **doc 1**
[2] **doc 2**
...
question

Table 6: RAG prompt for FLARE.

tations, which does not meet the ELI5 setting for long text annotations.

The special prompt for FLARE is demonstrated in Table 6.

The special prompt for Self-Ask is demonstrated in Table 10. Specifically, we use the LLM itself as a reader to extract concise answers as intermediate answers from the documents found in search. This was implemented using the Google API in the original paper, but we use our own Wiki document search library, hence the need for this approach.

C Case Study

We provide some cases of misleading references in Table 7. There are mainly two scenarios where searching can have adverse effects: (1) The references retrieved is misleading, the LLM is provided incorrect references; (2) The references retrieved is incomplete, causing the language model to focus on the answer found and overlook other possible answers.

Question: Where are the winter Olympics and when do they start?
reference: Åre and Östersund, Sweden will host the next World Winter Games between February 2 to 13, 2021. It will mark the first time that Sweden has ever hosted the Special Olympics World Games. **Error Type:** irrelevant reference **reference:** The EOC launched the bid process on 20 September 2018 after a meeting of the constituent National Olympic Committees in Stockholm. **Error Type:** irrelevant reference
Question: When did the golden state warriors win the finals?
reference: The 2017 NBA playoffs began on April 15, 2017. It concluded with the Golden State Warriors defeating the Cleveland Cavaliers 4 games to 1 in the NBA Finals, their third consecutive meeting at the Finals. **Error Type:** incomplete reference **reference:** This Finals was the first time in NBA history the same two teams had met for a third consecutive year. The Cavaliers sought to repeat as champions after winning the championship in 2016, while the Warriors won the first meeting in 2015. **Error Type:** incomplete reference

Table 7: Cases of misleading and incomplete references.

GPT-4 Prompt for Annotating Query Rewrite from User Question
Your task is to perform text analysis on user conversations, and complete the last json item. You need to follow the following rules: 1. Classify user conversations into the following categories: text rewriting, mathematical problems, knowledge questions, text creation, table processing, translation, summarization, logical reasoning, open qa, coding, text classification, information extraction, brainstorming, exams, role-playing, others. The format should be a string and stored in the task field. 2. Determine whether the answer of user input is closely related to current datetime, and store it in the timeliness field in boolean format. 3. If the user's request involves reasoning, each reasoning process should be described as questions and split into as many sub-questions as possible. 4. The sub-questions after splitting should be placed in the question field in questions, and the sub-questions should be fully described without using pronouns such as "he", "this", or "that". 5. If the sub-question involves very strict factual information such as personal relationships, time, location, policies, regulations, etc., which requires the use of a search engine to answer, then it needs to be marked as needSearch=true, and the generated search term should be placed in searchWord. 6. If the sub-question is a chit-chat question such as "how are you" or a pure mathematical problem, coding, logical reasoning, creative thinking, or common sense problem, then no search is needed. 7. Extract the entities and events involved in the user's request and store them in the entities and events fields respectively. The format is a list of strings. Note that the entities and events should be higly informative, and should not be a user instruction or a question.
GPT-4 Prompt for Annotating Query Rewrite from User Question
«SYS»You are asked to first separate a given text by claims and then provide a search query to verify each claim if needed. Here are some requirements: 1. The separation is conducted according to the meaning and each claim should be be brief and contain as one key claim. 2. Do not add any hallucinated information or miss any information. 3. The claims should be independent and self-contained, and the claims should be fully described without using pronouns such as "he", "this", or "that". 4. The query is derived from it's corresponding claim and the original user question, and should be useful to check the factuality of the claim. 5. If the claim does not contain any fact relevant with the original user question, or only contains simple commen senses, then search is not required. 6. The final return should strictly follow the given format. Like this: <Claims> <Claim(claim1)> <Search(True/False)> <Query(query1)> <Claim(claim2)> <Search(True/False)> <Query(query2)> <Claim(claim3)><Search(True/False)><Query(query3)>.....</Claims> «/SYS»

Table 8: The prompt to induce GPT-4 auxiliary annotation for query rewriting model.

RAG Prompt for Short-Form QA
«SYS» Now, based on the following reference and your knowledge, please answer the question more succinctly and professionally. The reference is delimited by triple brackets [[[]]]. The question is delimited by triple parentheses ((())). You should include as many possible answers as you can. «/SYS» Reference: [[[**reference**]]], question: (((**question**)))
RAG Prompt for Long-form QA
«SYS» Now, based on the following reference and your knowledge, please answer the question more succinctly and professionally. The reference is delimited by triple brackets [[[]]]. The question is delimited by triple parentheses ((())). You are not allowed to add fabrications or hallucinations. «/SYS» Reference: [[[**reference**]]], question: (((**question**)))

Table 9: RAG prompt for different tasks.

RAG Prompt for Self-Ask
«SYS» Given the following question, answer it by providing follow up questions and intermediate answers. If no follow up questions are necessary, answer the question directly. «SYS» Question: Who lived longer, Muhammad Ali or Alan Turing? Are follow up questions needed here: Yes. Follow up: How old was Muhammad Ali when he died? Intermediate answer: Muhammad Ali was 74 years old when he died. Follow up: How old was Alan Turing when he died? Intermediate answer: Alan Turing was 41 years old when he died. So the final answer is: Muhammad Ali Question: When was the founder of craigslist born? Are follow up questions needed here: Yes. Follow up: Who was the founder of craigslist? Intermediate answer: Craigslist was founded by Craig Newmark. Follow up: When was Craig Newmark born? Intermediate answer: Craig Newmark was born on December 6, 1952. So the final answer is: December 6, 1952 Question: **question**
RAG Prompt for Self-Ask Reference Reader
Given the following reference, answer it by a brief sentence. You are not allowed to add fabrications or hallucinations. **reference** Question: How old was Muhammad Ali when he died? Answer: Muhammad Ali was 74 years old when he died. Question: Who was the founder of craigslist? Answer: Craigslist was founded by Craig Newmark. Question: Who was the father of Mary Ball Washington? Answer: The father of Mary Ball Washington was Joseph Ball. Question: Who is the director of Casino Royale? Answer: The director of Casino Royale is Martin Campbell. Question: **question** Answer:

Table 10: RAG prompt for Self-Ask.