Query Rewriting for Retrieval-Augmented Large Language Models

Xinbei Ma[1,2,*] , **Yeyun Gong**[3, #, †], **Pengcheng He**[4, #], **Hai Zhao**[1,2,†], **Nan Duan**[3]
[1]Department of Computer Science and Engineering, Shanghai Jiao Tong University
[2]Key Laboratory of Shanghai Education Commission for Intelligent Interaction
and Cognitive Engineering, Shanghai Jiao Tong University
[3]Microsoft Research Asia [4]Microsoft Azure AI
sjtumaxb@sjtu.edu.cn, zhaohai@cs.sjtu.edu.cn,
{yegong, nanduan}@microsoft.com, Herbert.he@gmail.com

Abstract

Large Language Models (LLMs) play powerful, black-box readers in the *retrieve-then-read* pipeline, making remarkable progress in knowledge-intensive tasks. This work introduces a new framework, *Rewrite-Retrieve-Read* instead of the previous *retrieve-then-read* for the retrieval-augmented LLMs from the perspective of the query rewriting. Unlike prior studies focusing on adapting either the retriever or the reader, our approach pays attention to the adaptation of the search query itself, for there is inevitably a gap between the input text and the needed knowledge in retrieval. We first prompt an LLM to generate the query, then use a web search engine to retrieve contexts. Furthermore, to better align the query to the frozen modules, we propose a trainable scheme for our pipeline. A small language model is adopted as a trainable rewriter to cater to the black-box LLM reader. The rewriter is trained using the feedback of the LLM reader by reinforcement learning. Evaluation is conducted on downstream tasks, open-domain QA and multiple-choice QA. Experiments results show consistent performance improvement, indicating that our framework is proven effective and scalable, and brings a new framework for retrieval-augmented LLM [1].

1 Introduction

Large Language Models (LLMs) have shown remarkable abilities for human language processing and extraordinary scalability and adaptability in few- or zero-shot settings.(Ouyang et al., 2022; Brown et al., 2020; Chowdhery et al., 2022). However, the training process depends on large-scale high-quality corpora but without the perception of the real world. Thus, LLMs still have to face the issue of hallucination (Yao et al., 2023; Bang et al., 2023) and temporal misalignment (Röttger and Pierrehumbert, 2021; Luu et al., 2022; Jang et al., 2022). This affects the reliability of LLMs and hinders wider practical application, because the consistency between the LLM responses with the real world needs further validation. Existing work has proved that incorporating external knowledge (i.e., non-parametric knowledge) with internal knowledge (i.e., parametric knowledge) can effectively alleviate hallucination, especially for knowledge-intensive tasks. In fact, retrieval-augmented LLMs have been shown so effective that they have been regarded as a standard solution to alleviate the factuality drawbacks in naive LLM generations. Retrieval augmentation is applied to select relative passages as external contexts for the language model, which is *retrieve-then-read* framework (Lewis et al., 2020b; Karpukhin et al., 2020; Izacard et al., 2022). Take the open-domain Question-Answering task (open-domain QA) as an example, a retriever first searches for related documents for a question. Then the LLM receives the question and the documents, then predicts an answer.

As most LLMs are only accessible through inference APIs, they play the part of black-box frozen readers in the pipeline. This makes previous retrieval augmentation methods that require complete access (Lewis et al., 2020b; Guu et al., 2020; Izacard et al., 2022) no longer feasible. Recent studies on retrieval-augmented language models lean more on the LLM-oriented adaptation. An idea is to train a dense retrieval model to cater to the frozen language model (Shi et al., 2023). By using feedback from the LLM as a training objective, the retrieval model is tuned for better LLM input contexts. Another research line focuses on the design of interactions between the retriever and the reader (Yao et al., 2023; Khattab et al., 2022), where both the

[*] Work done during an internship at [3]Microsoft Research Asia. # Equal contribution. †Corresponding author.

This paper was partially supported by Joint Research Project of Yangtze River Delta Science and Technology Innovation Community (No. 2022CSJGG1400).

[1] https://github.com/xbmxb/RAG-query-rewriting

retriever and the reader are usually frozen. The idea is to trigger the emergent ability through carefully crafted prompts or a sophisticated prompt pipeline. Multiple interactions with external knowledge allow the LLM to approach the correct answer step by step.

However, there are still problems remaining to be solved. Existing approaches overlook the adaptation of the query, i.e., the input of the *retrieve-then-read* pipeline. The retrieval query is either original from datasets or directly determined by the black-box generation, thus is always fixed. However, there is inevitably a gap between the input text and the knowledge that is really needed to query. This limits performance and places a burden on retrieval capability enhancement and prompt engineering.

In consideration of this issue, this paper proposes *Rewrite-Retrieve-Read*, a new framework for retrieval augmentation, which can be further tuned for adapting to LLMs. In front of the retriever, a step of *rewriting the input* is added, filling the gap between the given input and retrieval need, as is shown in Figure 1. We adopt the off-the-shelf tool, an internet search engine, as the retriever, which avoids the maintenance of the search index and can access up-to-date knowledge (Lazaridou et al., 2022). Different from previous studies (Khattab et al., 2022; Yao et al., 2023) that require the memory of multiple interaction rounds between the retriever and the LLM for each sample, the motivation of our rewriting step is to clarify the retrieval need from the input text.

We also propose a trainable scheme for our *rewrite-retrieve-read* framework (Figure 1 (c)). The black-box retriever and the reader form a frozen system. To further smooth the steps of our pipeline, we apply a small, trainable language model to perform the rewriting step, denoted as the *rewriter*. The rewriter is trained by reinforcement learning using the LLM performance as a reward, learning to adapt the retrieval query to improve the reader on downstream tasks.

Our proposed methods are evaluated on knowledge-intensive downstream tasks including open-domain QA (HotpoQA (Yang et al., 2018), AmbigNQ (Min et al., 2020), PopQA (Mallen et al., 2022)) and multiple choice QA (MMLU (Hendrycks et al., 2021)). The experiments are implemented on T5-large (Raffel et al., 2020) as the rewriter, ChatGPT (Ouyang et al., 2022) and Vicuna-13B (Chiang et al., 2023) as the LLM reader. The results show that query rewriting consistently improves the retrieve-augmented LLM performance. The results also indicate that the smaller language model can be competent for query rewriting.

To sum up, our proposed novel retrieval-augmentation method, *rewrite-retrieve-read* is the first framework where the input text is adapted for the frozen retriever and LLM reader. We introduce a tuneable scheme with a small, trainable model, achieving performance gains with less resource consumption.

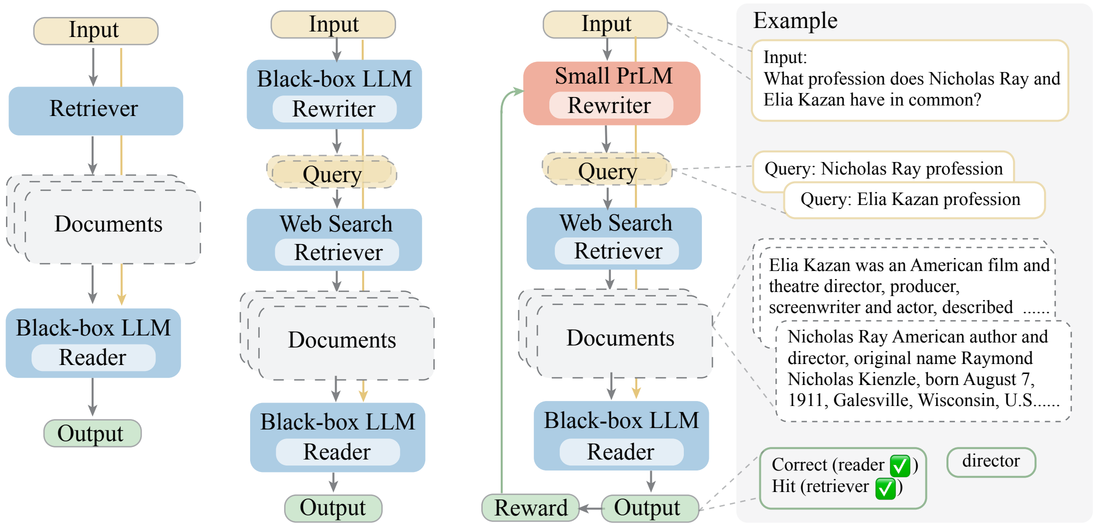
2 Related Work

2.1 Retrieval Augmentation

Language models require external knowledge to alleviate the factuality drawbacks. Retrieval augmentation has been regarded as the standard effective solution. With a retrieval module, related passages are provided to the language model as the context of the original input. Thus factual information like common sense or real-time news helps with output prediction through contextualized reading comprehension.

Earlier studies use sparse retriever (Chen et al., 2017) or dense retriever (Karpukhin et al., 2020) in front of a pre-trained language model (PrLM). The neural retriever and reader are both PrLMs of trainable size like BERT (Devlin et al., 2019) or BART (Lewis et al., 2020a). Hence, the whole *retrieve-then-reader* framework is a tuneable end-to-end system, where the retrieved contexts can be regarded as the intermediate results (Karpukhin et al., 2020; Lewis et al., 2020b). Approaches to smooth the two-step framework are proposed to optimize the retrieval and the reading comprehension (Sachan et al., 2021; Lee et al., 2022; Jiang et al., 2022). More recently, retrieval remains a powerful enhancement as the size of models and data scales rapidly (Mallen et al., 2022; Shi et al., 2023; Brown et al., 2020). On the other hand, retrieval enhancement can compensate for the shortfall in parameter size, compared to large-scale language models. For example, by jointly training the retriever and the reader, Atlas (Izacard et al., 2022) shows few-shot performance on par with 540B PalM (Chowdhery et al., 2022) but be of 50× smaller size.

The Internet as a knowledge base More related to our work, the search engine can assume the role of the retriever and use the Internet as the source of


Input
Retriever
Documents
Black-box LLM
Reader
Output
Input
Black-box LLM
Rewriter
Query
Web Search
Retriever
Documents
Black-box LLM
Reader
Output
Input
Small PrLM
Rewriter
Query
Web Search
Retriever
Documents
Black-box LLM
Reader
Reward
Output
Example
Input:
What profession does Nicholas Ray and Elia Kazan have in common?
Query: Nicholas Ray profession
Query: Elia Kazan profession
Elia Kazan was an American film and theatre director, producer, screenwriter and actor, described ......
Nicholas Ray American author and director, original name Raymond Nicholas Kienzle, born August 7, 1911, Galesville, Wisconsin, U.S......
Correct (reader ✓)
Hit (retriever ✓)
director

(a) Retrieve-then-read (b)Rewrite-retrieve-read (c) Trainable rewrite-retrieve-read

Figure 1: Overview of our proposed pipeline. From left to right, we show (a) standard *retrieve-then-read* method, (b) LLM as a query rewriter for our *rewrite-retrieve-read* pipeline, and (c) our pipeline with a trainable rewriter.

external knowledge. Komeili et al. (2022) use an internet search for relevant information based on the dialogue history to perform dialogue response generation. SeeKeR (Shuster et al., 2022) use a single Transformer to iteratively perform search query generation, then knowledge extraction for dialogue generation and sentence completion. For large-scale models, web search still shows effective for knowledge augmentation (Lazaridou et al., 2022), fact-checking (Menick et al., 2022), and LLM agent enhancement (Yao et al., 2023).

2.2 Cooperation with Black-box LLMs

Large Language Models, such as ChatGPT (Ouyang et al., 2022), Codex (Chen et al., 2021), PaLM (Chowdhery et al., 2022), emerge impressive natural language processing ability as well as remarkable scalability. This leads to a tendency to embrace LLMs on a wide range of NLP tasks. However, LLMs are only accessible as a black box in most cases, which is because (i) Some like ChatGPT are not open-source and kept private; (ii) The large parameter scale requires computational resources that are not always affordable to users. This constraint means nothing is available except input and output texts.

Existing studies have proved that LLMs' abilities can be better leveraged by carefully designed interaction methods. GenRead (Yu et al., 2023) prompts an LLM to generate context instead of deploying a retriever, showing that LLMs can retrieve internal knowledge by prompting. ReAct (Yao et al., 2023) and Self-Ask (Press et al., 2022) combines the Chain-of-Thought (CoT) (Wei et al., 2022; Wang et al., 2022) and inter-actions with web APIs. Only relying on prompt construction, ReAct provides novel baselines for interactive tasks. Demonstrate–Search–Predict (DSP) (Khattab et al., 2022) defines a sophisticated pipeline between an LLM and a retriever. Unlike ReAct, DSP integrates prompts for demonstration bootstrap besides multi-hop breakdown and retrieval.

Despite the promising performance in the zero or few-shot setting, the behavior of LLMs sometimes needs adjustments. A feasible approach is to append trainable small models in front of or after the LLM. The small models, as a part of the parameters of the system, can be fine-tuned for optimization. RePlug (Shi et al., 2023) is proposed to fine-tune a dense retriever for the frozen LLM in the *retrieve-then-read* pipeline. The retriever is trained under the LLM's supervision to retrieve documents that are suitable for the LLM. With the same purpose, Directional Stimulus Prompting (Li et al., 2023) deploys a small model to provide the LLM with stimulus (e.g., keywords for summarization, or dialogue actions for response generation), which is updated according to the LLM reward.

Different from the inspiring work mentioned above, our proposed pipeline contains a query rewriting step in front of the *retrieve-then-read* module. We further propose a trainable scheme with a small rewriting model, which is a novel enhancement for retrieval-augmented LLM by re-