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

검색 강화 대형 언어 모델을 위한 쿼리 다시 작성

북배 마[1,2,*], 예운 공[3, #, †], 팽성 하[4, #], 해 조[1,2,†], 난 단[3] [1]컴퓨터 과학 및 공학부, 상해 교통 대학교 [2]상해 교육 위원회 지능형 상호작용 및 인지 공학 핵심 실험실, 상해 교통 대학교 [3]마이크로소프트 리서치 아시아 [4]마이크로소프트 애저 AI sjt
umaxb@sjtu.edu.cn, zhaohai@cs.sjtu.edu.cn, {yegong, nanduan}@microsoft.com, Herbert.he@gmail.com

요약

대규모 언어 모델(LLMs)은 *retrieve-then-read* 파이프라인에서 강력한 블랙박스 리더로 작동하며, 지식 집약적 과제에서 눈에 띄는 진전을 이루고 있습니다. 이 연구는 쿼리 재작성 관점에서 이전 *retrieve-then-read* 대신 *Rewrite-Retrieve- Read* 새로운 프레임워크를 소개합니다. 이전 연구들이 리트리버 또는 리더 중 하나에 적응하는 데 초점을 맞춘 반면, 이 접근법은 리트리벌에서 입력 텍스트와 필요한 지식 사이의 불가피한 격차를 고려하여 검색 쿼리 자체의 적응에 주의를 기울입니다. 먼저 LLM을 사용하여 쿼리를 생성한 다음 웹 검색 엔진을 사용하여 컨텍스트를 검색합니다. 또한 쿼리를 동결 모듈에 더 잘 맞추기 위해 파이프라인에 대한 학습 가능한 계획을 제안합니다. 작은 언어 모델이 블랙박스 LLM 리더에 맞게 학습 가능한 재작성기로 채택됩니다. 재작성기는 강화 학습을 통해 LLM 리더의 피드백을 사용하여 학습됩니다. 다운스트림 작업인 오픈 도메인 QA 및 다중 선택 QA에서 평가를 수행합니다. 실험 결과는 일관된 성능 향상을 보여주며, 이는 프레임워크가 효과적이고 확장 가능하며 검색 강화 LLM [1]에 대한 새로운 프레임워크를 제공함을 나타냅니다.

1 소개

대규모 언어 모델(LLMs)은 인간 언어 처리 및 소규모 또는 제로 샷 설정에서 놀라운 확장성과 적응성을 보여왔습니다.(Ouyang 외, 2022; Brown 외, 2020; Chowdhery 외, 2022). 그러나 훈련 과정은 대규모 고품질 코퍼스에 의존하지만 인식 없이 실제 세계와 관련하여. 따라서 LLM은 여전히 환각(Yao et al., 2023; Bang et al., 2023)과 시간 오류(Röttger and Pierrehumbert, 2021; Luu et al., 2022; Jang et al., 2022)라는 문제를 직면해야 합니다. 이는 LLM의 신뢰성을 영향을 미치고 더 넓은 실용적 적용을 방해하며, LLM 응답과 실제 세계 간의 일관성이 추가 검증이 필요하기 때문입니다. 기존 연구는 외부 지식(즉, 비모수적 지식)을 내부 지식(즉, 모수적 지식)과 결합하는 것이 환각을 효과적으로 완화할 수 있음을 증명했습니다. 특히 지식 집약적 작업에서 그렇습니다. 사실, 검색 강화 LLM은 매우 효과적인 것으로 입증되어 나티브 LLM 생성의 사실성 결함을 완화하는 표준 솔루션으로 간주되어 왔습니다. 검색 증가는 언어 모델에 대한 외부 컨텍스트로 관련 문서를 선택하기 위해 적용됩니다. 이는 *retrieve-then-read* 프레임워크(Lewis et al., 2020b ; Karpukhin et al., 2020; Izacard et al., 2022)입니다. 예를 들어, 오픈 도메인 질의 응답 작업(오픈 도메인 QA)을 보면, 검색기가 먼저 질문에 대한 관련 문서를 검색합니다. 그런 다음 LLM은 질문과 문서를 받아 답변을 예측합니다.

대부분의 LLM은 추론 API를 통해 접근할 수 있기 때문에 파이프라인에서 블랙박스 동결 리더의 역할을 합니다. 이는 완전한 접근을 요구하는 이전 검색 증강 방법(Lewis 외, 2020b; Guu 외, 2020; Izac-ard 외, 2022)을 더 이상 실행할 수 없게 만듭니다. 검색 증강 언어 모델에 대한 최근 연구는 LLM 지향 적응에 더 중점을 둡니다. 하나의 아이디어는 동결 언어 모델을 수용하기 위해 밀도 높은 검색 모델을 훈련시키는 것입니다(Shi 외, 2023). LLM의 피드백을 훈련 목표로 사용하여 검색 모델은 더 나은 LLM 입력 컨텍스트에 맞게 조정됩니다. 다른 연구 방향은 검색기와 리더 간의 상호작용 설계에 초점을 맞춥니다(Yao 외, 2023; Khattab 외, 2022), 여기서 두 구성 요소는 모두 {v*}

∗ 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

리트리버와 리더는 보통 동결됩니다. 아이디어는 신중하게 작성된 프롬프트나 정교한 프롬프트 파이프라인을 통해 잠재적인 능력을 발현시키는 것입니다. 외부 지식과의 여러 상호작용을 통해 LLM은 올바른 답에 단계적으로 접근할 수 있습니다.

그러나 여전히 해결해야 할 문제가 남아 있습니다. 기존 접근 방식은 쿼리 적응, 즉 *retrieve-then-read* 파이프라인의 입력을 간과합니다. 검색 쿼리는 데이터셋에서 직접 가져오거나 블랙박스 생성에 의해 직접 결정되므로 항상 고정되어 있습니다. 그러나 입력 텍스트와 실제로 쿼리해야 하는 지식 사이에는 필연적으로 차이가 존재합니다. 이는 성능을 제한하고 검색 기능 향상 및 프롬프트 엔지니어링에 부담을 줍니다.

이 문제를 고려하여, 본 논문은 *Rewrite-Retrieve-Read*를 제안합니다. 이는 검색 증강을 위한 새로운 프레임워크로, LLM에 맞게 추가로 조정할 수 있습니다. 검색기 앞에 *rewriting the input* 단계가 추가되어, 주어진 입력과 검색 필요 간의 격차를 해소합니다. 이는 그림 1에 나와 있습니다. 우리는 검색기로 인터넷 검색 엔진과 같은 시판 도구를 채택하여, 검색 인덱스 유지 관리를 피하고 최신 지식을 접근할 수 있도록 합니다 (Lazaridou et al., 2022). 이전 연구 (Khattab et al., 2022; Yao et al., 2023)와 달리, 각 샘플에 대해 검색기와 LLM 간의 여러 상호 작용 라운드의 메모리를 요구하지 않고, 우리의 재작성 단계의 동기는 입력 텍스트에서 검색 필요를 명확히 하는 것입니다.

저희는 또한 *rewrite-retrieve-read* 프레임워크(그림 1 (c))를 위한 학습 가능한 계획을 제안합니다. 블랙박스 검색기와 리더는 고정된 시스템을 형성합니다. 파이프라인의 단계를 더욱 원활하게 하기 위해, 저희는 작은 학습 가능한 언어 모델을 적용하여 재작성 단계를 수행합니다. 이는 *rewriter*로 표시됩니다. 재작성기는 LLM의 성능을 보상으로 사용하여 강화 학습을 통해 학습되며, 다운스트림 작업에서 리더를 개선하기 위해 검색 쿼리를 적응시키는 방법을 학습합니다.

우리가 제안한 방법들은 오픈 도메인 QA(HotpoQA (양 외, 2018), AmbigNQ (민 외, 2020), PopQA (Mallen 외, 2022))와 다중 선택 QA(MMLU (Hendrycks 외, 2021))를 포함한 지식 집약적 다운스트림 작업에서 평가되었습니다. 실험은 T5-large (Raffel 외, 2020)를 리라이터로, ChatGPT (Ouyang 외, 2022)와 함께 구현되었습니다. 비쿠나-13B (Chiang et al., 2023)를 LLM 리더로 사용. 결과는 쿼리 다시 쓰기가 일관되게 검색 강화 LLM 성능을 향상시킨다는 것을 보여줍니다. 결과는 또한 더 작은 언어 모델이 쿼리 다시 쓰기에 적합할 수 있음을 나타냅니다.

요약하자면, 우리가 제안한 새로운 검색 증강 방법인 *rewrite-retrieve-read*은 입력 텍스트를 동결된 검색기와 LLM 리더에 맞게 적응시키는 최초의 프레임워크입니다. 우리는 적은 자원 소비로 성능 향상을 달성하는 작은 훈련 가능 모델을 소개합니다.

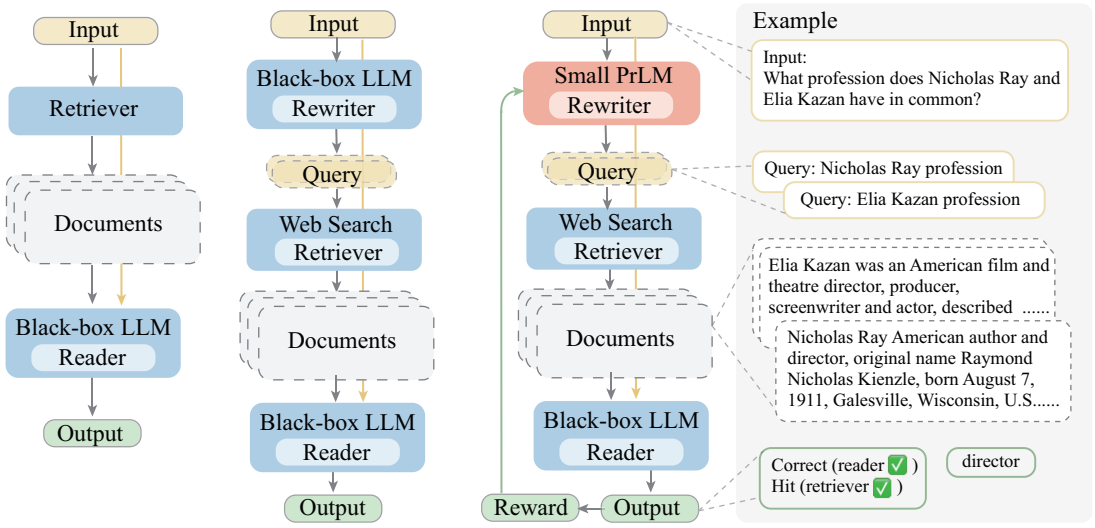
2 관련 연구

2.1 검색 강화

언어 모델은 사실적 한계를 보완하기 위해 외부 지식이 필요합니다. 검색 증강은 표준적이고 효과적인 해결책으로 간주되어 왔습니다. 검색 모듈을 통해 언어 모델은 원래 입력의 맥락으로 관련된 문장을 제공받습니다. 따라서 상식이나 실시간 뉴스 같은 사실적 정보는 맥락화된 독해 이해를 통해 출력 예측에 도움이 됩니다.

이전 연구에서는 사전 학습된 언어 모델(PrLM) 앞에 희소 검색기(Chen 외, 2017) 또는 밀집 검색기(Karpukhin 외, 2020)를 사용했습니다. 신경 검색기와 리더 모두 BERT(Devlin 외, 2019) 또는 BART(Lewis 외, 2020a)와 같은 학습 가능한 크기의 PrLM입니다. 따라서 전체 *retrieve-then-reader* 프레임워크는 조정 가능한 엔드투엔드 시스템이며, 검색된 컨텍스트는 중간 결과로 간주될 수 있습니다(Karpukhin 외, 2020; Lewis 외, 2020b). 두 단계 프레임워크를 원활하게 하는 접근 방식은 검색 및 읽기 이해를 최적화하기 위해 제안되었습니다(Sachan 외, 2021; Lee 외, 2022; Jiang 외, 2022). 최근에는 모델 및 데이터의 크기가 빠르게 증가함에 따라 검색이 강력한 향상 수단으로 유지되고 있습니다(Mallen 외, 2022; Shi 외, 2023; Brown 외, 2020). 다른 한편으로, 검색 강화는 대규모 언어 모델과 비교하여 매개변수 크기의 부족을 보상할 수 있습니다. 예를 들어, 검색기와 리더를 공동 학습함으로써 Atlas(Izacard 외, 2022)는 540B PalM(Chowdhery 외, 2022)과 유사한 몇 가지 성능을 보여주지만, 크기는 50× 더 작습니다.

인터넷을 지식 기반으로 활용하기 우리 업무와 더 관련성이 높은 검색 엔진은 검색 도구로서의 역할을 수행하고 인터넷을 정보원으로 활용할 수 있습니다.



(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).

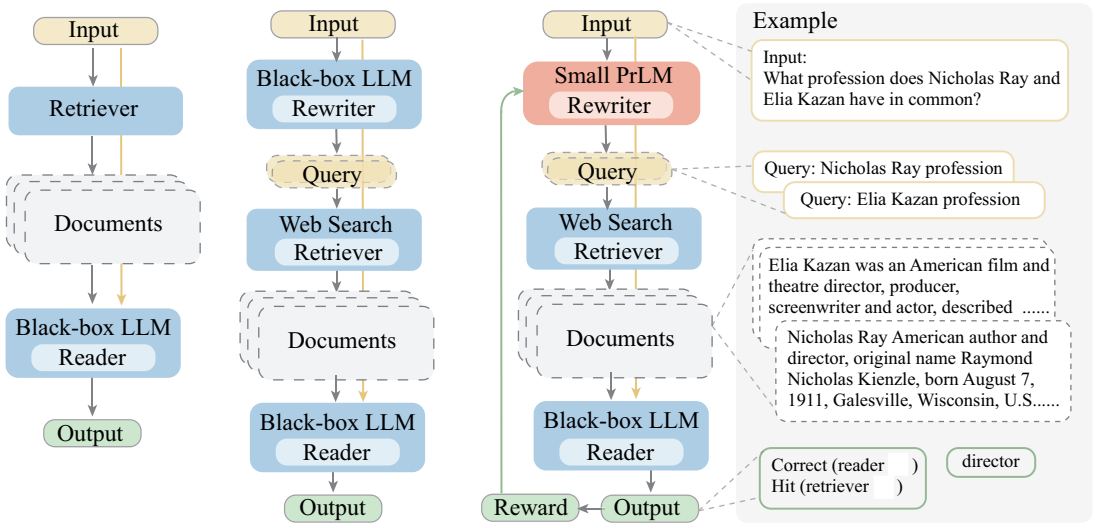
2.2 Cooperation with Black-box LLMs

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-



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

그림 1: 제안하는 파이프라인의 개요. 왼쪽에서 오른쪽으로, (a) 표준 *retrieve-then-read* 방법, (b) LLM을 쿼리 다시 작성기로 사용하는 *rewrite-retrieve-read* 파이프라인, (c) 학습 가능한 다시 작성기를 포함한 우리의 파이프라인을 보여줍니다.

외부 지식. 코메이리 외 (2022)는 대화 기록에 기반한 관련 정보 인터넷 검색을 사용하여 대화 응답 생성을 수행합니다. SeeKeR (셔스터 외, 2022)는 검색 쿼리 생성, 지식 추출을 위한 단일 트랜스포머를 반복적으로 사용하여 대화 생성 및 문장 완성에 사용합니다. 대규모 모델의 경우, 웹 검색은 여전히 지식 증강 (라자리두 외, 2022), 사실 확인 (메닉 외, 2022), LLM 에이전트 향상 (야오 외, 2023)에 효과적인 것으로 나타났습니다.

2.2 블랙박스 LLM과의 협력

대규모 언어 모델(Large Language Models), 예를 들어 챗GPT(Ouyang 외, 2022), 코드엑스(Chen 외, 2021), 팜(Chowdhery 외, 2022)은 인상적인 자연어 처리 능력뿐만 아니라 눈에 띄는 확장성도 보여줍니다. 이는 다양한 NLP 작업에 LLMs를 적용하려는 경향을 이끌어냅니다. 그러나 대부분의 경우 LLMs는 블랙박스로만 접근할 수 있는데, 이는 (i) 챗GPT와 같은 일부는 오픈소스가 아니고 비공개로 유지되고 있기 때문이며, (ii) 많은 매개변수 규모로 인해 사용자가 항상 감당할 수 있는 컴퓨팅 리소스가 필요하기 때문입니다. 이 제약 조건은 입력 및 출력 텍스트 외에는 아무것도 사용할 수 없다는 것을 의미합니다.

기존 연구들은 LLM의 능력을 신중하게 설계된 상호작용 방법을 통해 더 잘 활용할 수 있음을 증명했습니다. GenRead (Yu et al., 2023)는 리트리버를 배치하는 대신 LLM에 컨텍스트를 생성하도록 프롬프트하여 LLM이 프롬프트를 통해 내부 지식을 검색할 수 있음을 보였습니다. ReAct (Yao 외, 2023)과 Self-Ask (Press 외, 2022)는 체인-오브-생각(CoT) (Wei 외, 2022; Wang 외, 2022)과 웹 API와의 상호작용을 결합합니다. 프롬프트 구성에만 의존하여 ReAct는 상호작용 작업에 대한 새로운 기준선을 제공합니다. Demonstrate–Search–Predict (DSP) (Khattab 외, 2022)는 LM과 리트리버 사이의 정교한 파이프라인을 정의합니다. ReAct와 달리 DSP는 다중 홉 분해 및 검색 외에도 시연 부트스트랩을 위한 프롬프트를 통합합니다.

제로 샷 또는 몇 개 샷 설정에서 유망한 성능에도 불구하고, LLM의 행동은 때때로 조정이 필요합니다. 실현 가능한 방법은 LLM 앞뒤에 훈련 가능한 작은 모델을 추가하는 것입니다. 작은 모델은 시스템의 매개변수 일부로 최적화를 위해 미세 조정될 수 있습니다. RePlug (Shi 외, 2023)는 *retrieve- then-read* 파이프라인에서 동결된 LM을 위해 밀집 검색기를 미세 조정하기 위해 제안되었습니다. 검색기는 LLM의 감독 하에 LM에 적합한 문서를 검색하도록 훈련됩니다. 동일한 목적으로 Directional Stimulus Prompting (Li 외, 2023)은 LLM에 자극(예: 요약용 키워드 또는 응답 생성을 위한 대화 동작)을 제공하는 작은 모델을 배치하며, 이는 LLM 보상에 따라 업데이트됩니다.

위에서 언급한 영감을 주는 작업과 달리, 우리가 제안한 파이프라인은 *retrieve-then-read* 모듈 앞에 쿼리 다시 쓰기 단계를 포함합니다. 우리는 또한 작은 다시 쓰기 모델을 사용한 학습 가능한 구성을 제안하며, 이는 검색 강화 LLM에 대한 새로운 강화입니다.

constructing the search query.

3 Methodology

We present *Rewrite-Retrieve-Read*, a pipeline that improves the retrieval-augmented LLM from the perspective of query rewriting. Figure 1 shows an overview. This section first introduces the pipeline framework in section 3.1, then the trainable scheme in section 3.2.

3.1 *Rewrite-Retrieve-Read*

A task with retrieval augmentation can be denoted as follows. Given a dataset of a knowledge-intensive task (e.g., open-domain QA), $D = \{(x, y)_i\}, i = 0, 1, 2, \ldots, N$, x (e.g., a question) is the input to the pipeline, y is the expected output (e.g., the correct answer). Our pipeline consists of three steps. (i) Query rewrite: generate a query $\tilde{x}$ for required knowledge based on the original input x. (ii) Retrieve: search for related context, doc. (iii) Read: comprehend the input along with contexts $[doc, x]$ and predict the output $\hat{y}$.

A straightforward but effective method is to ask an LLM to rewrite queries to search for information that is potentially needed. We use a few-shot prompt to encourage the LLM to think, and the output can be none, one or more queries to search.

3.2 Trainable Scheme

Besides, total reliance on a frozen LLM has shown some drawbacks. Reasoning errors or invalid search hinders the performance (Yao et al., 2023; BehnamGhader et al., 2022). On the other hand, retrieved knowledge may sometimes mislead and compromise the language model (Mallen et al., 2022). To better align to the frozen modules, it is feasible to add a trainable model and adapt it by taking the LLM reader feedback as a reward.

Based on our framework, we further propose to utilize a trainable small language model to take over the rewriting step, as is shown in the right part of Figure 1. The trainable model is initialized with the pre-trained T5-large (770M) (Raffel et al., 2020), denoted as *trainable rewriter*, G_θ. The rewriter is first trained on pseudo data to warm up (§3.2.1), then continually trained by reinforcement learning (§3.2.2).

3.2.1 Rewriter Warm-up

The task, query rewriting, is quite different from the pre-training objective of sequence-to-sequence generative models like T5. First, we construct a pseudo dataset for the query rewriting task. Inspired by recent distillation methods (Hsieh et al., 2023; Ho et al., 2022), we prompt the LLM to rewrite the original questions x in the training set and collect the generated queries $\tilde{x}$ as pseudo labels. The collected samples are then filtered: Those that get correct predictions from the LLM reader are selected into the warm-up dataset, denoted as $D_{Train} = \{(x, \tilde{x}) | \hat{y} = y\}$. The rewriter G_θ is fine-tuned on D_{Train} with the standard log-likelihood as the training objective, denoted as

$$\mathcal{L}_{warm} = -\sum_t logp_\theta(\hat{\tilde{x}}_t \mid \tilde{x}_{<t}, x). \quad (1)$$

The rewriter model after warm-up shows modest performance, which depends on the pseudo data quality and rewriter capability. Highly relying on the human-written prompt line, $\tilde{x}$ can be sub-optimal. The relatively small scale of the rewriter size is also a limitation of the performance after the warm-up. Then we turn to reinforcement learning to align the rewriter to the following retriever and LLM reader.

3.2.2 Reinforcement Learning

To further fine-tune the rewriter to cater to the LLM reader, we adopt a policy gradient reinforcement learning framework.

Task Formulation In the context of reinforcement learning, the rewriter optimization is formulated as a Markov Decision Process 5-tuple $\langle \mathcal{S}, \mathcal{A}, P, R, \gamma \rangle$. (i) The state space $\mathcal{S}$ is a finite set limited by the vocabulary and the sequence length. (ii) The action space $\mathcal{A}$ is equals to the vocabulary. (iii) The transition probability P is determined by the policy network, which is the rewriter model G_θ. (iv) The reward function R gives a reward value that depends on the current state. The policy gradient is derived from rewards, used as the training objective. (v) γ denotes the discount factor. More specifically, the rewriter G_θ after the warm-up is the initial policy model π_0. At each step t, the action a_t is to generate the next token $\hat{\tilde{x}}_t$ based on the observation of the present state, $s_t = [x, \hat{\tilde{x}}_{<t}]$. When the generation is stopped by the End-Of-Sentence token, one episode is ended. After finishing the retrieval and reading, a reward is computed by evaluating the final output, i.e., a score for the LLM reader prediction.

Policy Optimization We adopt Proximal Policy Optimization (PPO) (Schulman et al., 2017), following (Ramamurthy et al., 2022). Maximization

constructing the search query.

3 방법론

저희는 *Rewrite-Retrieve-Read*을(를) 제시합니다. 이는 쿼리 재작성 관점에서 검색 강화 LLM을 개선하는 파이프라인입니다. 그림 1은 개요를 보여줍니다. 이 섹션에서는 먼저 섹션 3.1에서 파이프라인 프레임워크를 소개한 다음 섹션 3.2에서 학습 가능한 스키마를 소개합니다.

3.1 ***Rewrite-Retrieve-Read***

지식 기반 작업에 대한 검색 증강 작업을 다음과 같이 나타낼 수 있습니다. 예를 들어, 오픈 도메인 QA와 같은 지식 기반 작업의 데이터 세트가 주어지면, $D = \{(x, y)_i\}, i = 0, 1, 2, \ldots, N$, x (예컨대 질문)이 파이프라인에 입력으로 들어가고, y은 기대되는 출력(예: 정답)입니다. 저희 파이프라인은 세 단계로 구성됩니다. (i) 쿼리 재작성: 원본 입력 x을 기반으로 필요한 지식에 대한 쿼리 $\tilde{x}$를 생성합니다. (ii) 검색: 관련 컨텍스트를 검색합니다. doc (iii) 읽기: 입력과 컨텍스트 $[doc, x]$를 함께 이해하고 출력 $\hat{y}$을 예측합니다.

간단하지만 효과적인 방법은 LLM에게 잠재적으로 필요한 정보를 검색하기 위한 쿼리를 다시 작성하도록 요청하는 것입니다. 우리는 LLM이 생각하도록 유도하기 위해 몇 가지 예시를 포함한 프롬프트를 사용하고, 출력은 검색할 쿼리가 없을 수도, 하나일 수도, 또는 여러 개일 수도 있습니다.

3.2 학습 가능 스키마

게다가, 동결된 LLM에 대한 완전한 의존성은 일부 단점을 드러냈습니다. 추론 오류 또는 유효하지 않은 검색이 성능을 저해합니다 (Yao et al., 2023; BehnamGhader et al., 2022). 반면에, 검색된 지식이 때로는 언어 모델을 오도하고 손상시킬 수 있습니다 (Mallen et al., 2022). 동결된 모듈에 더 잘 맞추기 위해, 훈련 가능한 모델을 추가하고 LLM 리더의 피드백을 보상으로 삼아 이를 적응시키는 것이 가능합니다.

저희 프레임워크에 기반하여, 우리는 다시 작은 학습 가능한 언어 모델을 사용하여 다시 작성 단계를 수행하는 것을 제안합니다. 이는 그림 1의 오른쪽 부분에 표시되어 있습니다. 학습 가능한 모델은 미리 학습된 T5-large (770M) (Raffel et al., 2020)로 초기화되며, 이는 *trainable rewriter*, G_θ로 표시됩니다. 다시 작성기는 먼저 가짜 데이터로 학습되어 워밍업됩니다 (§3.2.1), 그 후 강화 학습으로 계속해서 학습됩니다 (§3.2.2).

3.2.1 리라이터 워밍업

작업인 쿼리 재작성은 T5와 같은 시퀀스-투-시퀀스 생성 모델의 사전 학습 목표와 상당히 다릅니다. 먼저, 우리는 {v*}를 구성합니다.

쿼리 재작성 작업용 가짜 데이터셋. 최근 증류 방법(Hsieh et al., 2023; Ho et al., 2022)에 영감을 받아, LLM에 원본 질문 x을 재작성하도록 프롬프트하고 생성된 쿼리 $\tilde{x}$를 가짜 레이블로 수집합니다. 수집된 샘플은 그 후 필터링됩니다: LLM 리더로부터 올바른 예측을 받는 것들은 워밍업 데이터셋 $D_{Train} = \{(x, \tilde{x}) | \hat{y} = y\}$에 선택됩니다. 리라이터 G_θ은 표준 로그 가능성을 훈련 목표로 D_{Train}에 미세 조정됩니다.

$$\mathcal{L}_{warm} = -\sum_t logp_\theta(\hat{\tilde{x}}_t \mid \tilde{x}_{<t}, x). \quad (1)$$

워밍업 후의 리라이터 모델은 가짜 데이터의 품질과 리라이터의 능력에 따라 겸손한 성능을 보입니다. 인간이 작성한 프롬프트 라인에 크게 의존하는 $\tilde{x}$는 최적이 아닐 수 있습니다. 상대적으로 작은 리라이터의 크기도 워밍업 후 성능의 한계입니다. 그러므로 강화 학습을 통해 리라이터를 다음과 같은 리트리버와 LLM 리더에 맞추도록 전환합니다.

3.2.2 강화 학습

LLM 독자를 위해 리라이터를 더욱 세밀하게 조정하기 위해, 우리는 정책 기울기 강화 학습 프레임워크를 채택합니다.

강화 학습의 맥락에서, 재작성 최적화는 다음과 같이 마르코프 결정 과정(Markov Decision Process)의 5-튜플 $\langle \mathcal{S}, \mathcal{A}, P, R, \gamma \rangle$로 공식화됩니다.(i) 상태 공간 $\mathcal{S}$은 어휘와 문장 길이에 제한을 받는 유한 집합입니다.(ii) 행동 공간 $\mathcal{A}$은 어휘와 같습니다.(iii) 전환 확률 P은 정책 네트워크, 즉 재작성 모델 G_θ에 의해 결정됩니다.(iv) 보상 함수 R는 현재 상태에 따라 달라지는 보상 값을 제공합니다. 정책 기울기는 보상에서 유도되어 학습 목표로 사용됩니다.(v) γ은 할인 계수를 나타냅니다. 구체적으로, 워밍업 후의 재작성기 G_θ는 초기 정책 모델 π_0입니다. 각 단계 t에서 행동 a_t은 현재 상태의 관찰 $s_t = [x, \hat{\tilde{x}}_{<t}]$을 기반으로 다음 토큰 $\hat{\tilde{x}}_t$을 생성하는 것입니다. 문장 종료 토큰에 의해 생성이 중단되면 한 에피소드가 끝납니다. 검색 및 읽기를 마친 후, 최종 출력에 대한 평가로 보상이 계산되며, 즉 LLM 리더 예측에 대한 점수입니다.

정책 최적화 우리는 (Schulman et al., 2017)의 근접 정책 최적화(PPO)를 채택하며, (Ramamurthy et al., 2022)를 따릅니다. 최대화

of the expectation of the reward R is formulated as

$$\begin{aligned}
&\max_{\theta} \mathbb{E}_{\hat{\tilde{x}} \sim p_\theta(\cdot|x)}[R(x, \hat{\tilde{x}})], \\
&\max_{\theta} \mathbb{E}_{(s_t,a_t)\sim\pi_{\theta'}}[min\{k_{t,\theta} A^{\theta'}(s_t, a_t); \\
&\qquad \text{clip}\left(k_{t,\theta}, 1-\varepsilon, 1+\varepsilon\right) A^{\theta'}(s_t, a_t)\}], \\
&k_{t,\theta} = \frac{p_\theta(a_t \mid s_t)}{p_{\theta'}(a_t \mid s_t)},
\end{aligned} \tag{2}$$

where θ' is the temporarily fixed policy for sampling and θ is updated. A denotes the advantage function, which is formulated based on the estimation of value network V_ϕ. The value network V_ϕ is initialized from the policy network π_0. The formulation follows Generalized Advantage Estimation (GAE) (Schulman et al., 2015).

$$\begin{aligned}
\delta_t &= R(s_t, a_t) + V_\phi(s_{t+1}) - V_\phi(s_t), \\
\hat{A}_t^\theta(s_t, a_t) &= \sum_{t'=0}^{\infty} \lambda^{t'} \delta_{t+t'},
\end{aligned} \tag{3}$$

where λ is the bias-variance trade-off parameter.

The reward function R reflects the quality of the generated queries, which needs to be consistent with the final evaluation of the task. $\hat{\tilde{x}}$ is fed to the retriever and the reader for a final prediction $\hat{y}$. A part of the reward function is the measures of $\hat{y}$ compared to the golden label y (e.g., exact match and F_1 of the predicted answers), denoted as R_{lm}. Besides, a KL-divergence regularization is added to prevent the model from deviating too far from the initialization (Ramamurthy et al., 2022; Ziegler et al., 2019).

$$R(s_t, a_t) = R_{lm}(\hat{\tilde{x}}, y) - \beta \text{KL}(\pi_\theta \| \pi_0). \tag{4}$$

The final loss function is composed of policy loss and value loss.

$$\begin{aligned}
\mathcal{L}_\theta &= -\frac{1}{|\mathcal{S}|\,T} \sum_{\tau\in\mathcal{S}} \sum_{t=0}^{T} \min(k_{t,\theta} A^{\theta'}, \text{clip}\, A^{\theta'}), \\
\mathcal{L}_\phi &= \frac{1}{|\mathcal{S}|\,T} \sum_{\tau\in\mathcal{S}} \sum_{t=0}^{T} \left(V_\phi(s_t) - R_t\right)^2, \\
\mathcal{L}_{ppo} &= \mathcal{L}_\theta + \lambda_v \mathcal{L}_\phi.
\end{aligned} \tag{5}$$

Here, $\mathcal{S}$ denotes the sampled set, and T is for step numbers.

4 Implementation

Rewriter For the frozen pipeline in §3.1, we prompt an LLM to rewrite the query with few-shot in-context learning (Brown et al., 2020; Min et al., 2022). Our prompt follows the formulation of *[instruction, demonstrations, input]*, where the input is x. The instruction is straightforward and demonstrations are 1-3 random examples from training sets and are kept constant across all runs, mainly for the task-specific output format illustration, i.e., a short phrase as an answer for HotpotQA, and an option as an answer for MMLU. For the training scheme in §3.2, we fine-tuning a T5 as the rewriter.

Retriever We use the Bing search engine as the retriever. It requires no candidate index construction like a dense retriever, nor candidates like a textbook. But it allows for a wide knowledge scope and up-to-time factuality. With Bing API, the retrieval is performed in two approaches. (i) For all retrieved web pages, we concatenate the snippets that are related sentences selected by Bing. This method is similar to using a search engine in a browser, input a query and press Enter, then collect the texts shown on the search result page. (ii) For retrieved web pages, we request the URLs and parser to get all the texts. This is similar to clicking on items on the search result page. Then we use BM25 to keep those with higher relevance scores with the query, reducing the document length.

Reader The reader is a frozen LLM, where we adopt ChatGPT (gpt-3.5-turbo) and Vicuna-13B. It performs reading comprehension and prediction with few-shot in-context learning. In our prompt, following the brief instruction and the demonstrations, the input is x or $[doc, \hat{\tilde{x}}]$ with retrieval augmentation.

It has been proved that both the phrasing of prompt lines (Zhang et al., 2023a) and the selection of demonstrations show effects on the in-context learning performance (Su et al., 2022; Zhang et al., 2023b). As it is not the focus of this work, we pay no more attention to prompt editing.

5 Experiments

5.1 Task Settings

5.1.1 Open-domain QA

Three open-domain QA datasets are used for evaluation. (i) HotPotQA (Yang et al., 2018) consists of complex questions that require multi-hop reasoning. We evaluate the full test set. (ii) AmbigNQ (Min et al., 2020) provides a disambiguated version of Natural Questions (NQ) (Kwiatkowski et al., 2019). For ambiguous questions in NQ, minimal constraints are added to break it into several similar

보상 R에 대한 기대는 다음과 같이 표현됩니다.

$$\begin{aligned}
&\max_{\theta} \mathbb{E}_{\hat{\hat{x}} \sim p_{\theta}(\cdot|x)}[R(x, \hat{\hat{x}})], \\
&\max_{\theta} \mathbb{E}_{(s_t, a_t) \sim \pi_{\theta'}}[min\{k_{t,\theta} A^{\theta'}(s_t, a_t); \\
&\qquad \text{clip}\left(k_{t,\theta}, 1-\varepsilon, 1+\varepsilon\right) A^{\theta'}(s_t, a_t)\}], \\
&k_{t,\theta} = \frac{p_{\theta}(a_t \mid s_t)}{p_{\theta'}(a_t \mid s_t)},
\end{aligned} \tag{2}$$

여기서 θ'는 샘플링을 위한 일시적으로 고정된 정책이고, θ는 업데이트됩니다. A는 가치 함수(value function)를 나타내며, 이는 가치 네트워크 V_{ϕ}의 추정치에 기반하여 공식화됩니다. 가치 네트워크 V_{ϕ}는 정책 네트워크 π_0에서 초기화됩니다. 이 공식은 일반화된 이점 추정(Generalized Advantage Estimation, GAE) (Schulman et al., 2015)을 따릅니다.

$$\begin{aligned}
&\delta_t = R(s_t, a_t) + V_{\phi}(s_{t+1}) - V_{\phi}(s_t), \\
&\hat{A}_t^{\theta}(s_t, a_t) = \sum_{t'=0}^{\infty} \lambda^{t'} \delta_{t+t'},
\end{aligned} \tag{3}$$

λ는 편향-분산 트레이드오프 파라미터입니다.

보상 함수 R는 생성된 쿼리의 품질을 반영하며, 이는 과제의 최종 평가들과 일치해야 한다. $\hat{\hat{x}}$은 최종 예측 $\hat{y}$를 위해 리트리버와 리더에 입력된다. 보상 함수의 일부는 $\hat{y}$을 황금 라벨 y (와 비교한 측정값이다. 예를 들어, 예측된 답변의 정확 일치 및 F_1), R_{lm}로 표기된다. 또한, KL-발산 정규화가 추가되어 모델이 초기화에서 너무 벗어나지 않도록 방지한다 (Ramamurthy 외, 2022; Ziegler 외, 2019).

$$R(s_t, a_t) = R_{lm}(\hat{\hat{x}}, y) - \beta \text{KL}\left(\pi_{\theta} \| \pi_0\right). \tag{4}$$

최종 손실 함수는 정책 손실과 값 손실로 구성됩니다.

$$\begin{aligned}
&\mathcal{L}_{\theta} = -\frac{1}{|\mathcal{S}|\, T} \sum_{\tau \in \mathcal{S}} \sum_{t=0}^{T} \min(k_{t,\theta} A^{\theta'}, \text{clip}\, A^{\theta'}), \\
&\mathcal{L}_{\phi} = \frac{1}{|\mathcal{S}|\, T} \sum_{\tau \in \mathcal{S}} \sum_{t=0}^{T} \left(V_{\phi}(s_t) - R_t\right)^2, \\
&\mathcal{L}_{ppo} = \mathcal{L}_{\theta} + \lambda_v \mathcal{L}_{\phi}.
\end{aligned} \tag{5}$$

여기서 $\mathcal{S}$은 샘플링된 집합을 나타내고, T은 단계 번호를 나타냅니다.

4 구현

§3.1의 동결 파이프라인에 대해, 우리는 소수의 샷으로 LLM을 유도하여 쿼리를 다시 작성하도록 합니다. 문맥 학습 (Brown 외, 2020; Min 외, 2022). 우리의 프롬프트는 *[instruction, demonstrations, input]*의 구성을 따르며, 입력은 x입니다. 지침은 간단하고 직관적이며, 데모는 훈련 세트에서 무작위로 선택된 1-3개의 예시이며, 모든 실행에서 일관되게 유지됩니다. 주로 작업별 출력 형식 설명, 즉 HotpotQA에 대한 답변으로 짧은 구문과 MMLU에 대한 답변으로 옵션을 제시하기 위함입니다. §3.2의 훈련 방식에 대해, 우리는 T5를 재작성기로 미세 조정합니다. 검색기 빙(Bing) 검색 엔진을 검색기로 사용합니다. 밀집 검색기와 같은 후보자 색인 구성이 필요하지 않으며, 교과서와 같은 후보자도 필요하지 않습니다. 하지만 광범위한 지식 범위와 최신 사실성을 제공합니다. 빙(Bing) API를 통해 두 가지 방식으로 검색이 수행됩니다. (i) 검색된 모든 웹 페이지에 대해, 빙에 의해 선택된 관련 문장을 연결합니다. 이 방법은 브라우저에서 검색 엔진을 사용하여 쿼리를 입력하고 Enter를 누른 다음 검색 결과 페이지에 표시되는 텍스트를 수집하는 것과 유사합니다. (ii) 검색된 웹 페이지에 대해 URL을 요청하고 파서(parser)를 사용하여 모든 텍스트를 가져옵니다. 이는 검색 결과 페이지의 항목을 클릭하는 것과 유사합니다. 그런 다음 BM25를 사용하여 쿼리와 더 높은 관련성 점수를 가진 것들만 유지하여 문서 길이를 줄입니다.

독자 독자는 동결된 LLM으로, ChatGPT(gpt-3.5-turbo)와 Vicuna-13B를 사용합니다. 이는 몇 번의 샷으로 컨텍스트 학습을 통해 독해와 예측을 수행합니다. 우리의 프롬프트에서, 간략한 지시사항과 데모에 따라 입력은 x 또는 $[doc, \hat{\hat{x}}]$이며, 검색 증강이 포함됩니다.

장 등(2023a)의 연구에서 밝혀진 바와 같이 프롬프트 문장의 구성과 시연의 선택 모두가 컨텍스트 내 학습 성능에 영향을 미친다(수 등, 2022; 장 등, 2023b). 본 연구의 초점이 아니기 때문에 프롬프트 편집에 더 이상 주의를 기울이지 않는다.

5개의 실험

5.1 작업 설정

5.1.1 개방형 도메인 질의응답

세 개의 오픈 도메인 QA 데이터셋이 평가에 사용됩니다. (i) HotPotQA (양 외, 2018)는 다중 홉 추론을 필요로 하는 복잡한 질문들로 구성되어 있습니다. 우리는 전체 테스트 세트를 평가합니다. (ii) AmbigNQ (민 외, 2020)는 Natural Questions (NQ) (Kwiatkowski 외, 2019)의 명확화된 버전을 제공합니다. NQ의 모호한 질문에 대해 최소한의 제약 조건이 추가되어 여러 유사한 질문으로 분할됩니다.

Direct prompt
Answer the question in the following format, end the answer with '**'. {demonstration} Question: {x} Answer:
Reader prompt in retrieval-augment pipelines
Answer the question in the following format, end the answer with '**'. {demonstration} Question: {doc} {x} Answer:
Prompts for LLM as a frozen rewriter
Open-domain QA: Think step by step to answer this question, and provide search engine queries for knowledge that you need. Split the queries with ';' and end the queries with '**'. {demonstration} Question: {x} Answer: *Multiple choice QA:* Provide a better search query for web search engine to answer the given question, end the queries with '**'. {demonstration} Question: {x} Answer:

Table 1: Prompt lines used for the LLMs.

but specific questions. The first 1000 samples are evaluated in the test set. (iii) PopQA (Mallen et al., 2022) includes long-tail distributions as it contains more low-popularity knowledge than other popular QA tasks. We split the dataset into 13k for training and 714 for testing.

Open-domain QA benchmarks are sets of question-answer pairs denoted as $\{(q, a)_i\}$. We use ChatGPT for both the reader and the frozen rewriter. The evaluation metrics are Exact Match (EM) and F_1 scores. For the reward function in RL, we use an indicator to reward if the retrieved content hits the answer and penalize if misses the answer, denoted as Hit. The total reward is a weighted sum of EM, F_1, and Hit.

$$Hit = \begin{cases} 1 & a \text{ in } doc, \\ -1 & else \end{cases} \tag{6}$$
$$R_{lm} = EM + \lambda_f F_1 + \lambda_h Hit.$$

5.1.2 Multiple-choice QA

For multiple-choice QA, our evaluation is conducted on Massive Multi-task Language Understanding (MMLU) (Hendrycks et al., 2021), an exam question dataset including 4 categories: Humanities, STEM, Social Sciences, and Other. Each category is split into 80% for the training set and 20% for the test set.

Multiple-choice QA can be formulated as $\{(q', a)_i\}$, where $q' = [q, c_0, c_1, c_2, c_3]$. c denotes the options, generally there are four for each question. The retrieved documents that are included in the officially provided contaminated lists are ignored. The questions with options are rewritten into search queries. The answer is one option. EM is reported as metrics and used for the reward.

$$R_{lm} = EM. \tag{7}$$

We use ChatGPT as a frozen rewriter and the reader. We also use Vicuna-13B as the reader for evaluation due to the rate limit issue of ChatGPT. More information on datasets and training setup are presented in the appendix.

5.2 Baselines

The following settings are implemented to evaluate and support our methods. (i) **Direct**: The standard in-context learning without any augmentations. (ii) **Retrieve-then-read**: The standard retrieval-augmented method. Retrieved documents are concatenated with the question. (iii) **LLM as a frozen rewriter**: As is introduced in §3.1, we prompt a frozen LLM to reason and generate queries by few-shot in-context learning. (iv) **Trainable rewriter**: Applying the fine-tuned rewriter, the output queries are used by the retriever and the reader. Table 1 presents prompt line forms. Please note that the prompts for prediction are kept the same for each task.

5.3 Results

Experimental results on open-domain QA are reported in Table 2. For the three datasets, query rewriting consistently brings performance gain with both a frozen rewriter and a trainable rewriter. On AmbigNQ and PopQA, the standard retrieval augments the reader, indicating useful external knowledge is retrieved. On HotpotQA, the standard retrieval hurts the reader. This shows that using complex questions as queries cannot compensate for the parametric knowledge, but bring noises instead (Mallen et al., 2022). This suggests that multi-hop questions are not suitable queries for the web search engine. The scores increase by adding the rewriting step. On PopQA, our trainable rewriter surpasses standard retrieval while being inferior to the LLM rewriter. This indicates that the

Direct prompt
Answer the question in the following format, end the answer with '**'. {demonstration} Question: {x} Answer:
Reader prompt in retrieval-augment pipelines
Answer the question in the following format, end the answer with '**'. {demonstration} Question: {doc} {x} Answer:
Prompts for LLM as a frozen rewriter
Open-domain QA: Think step by step to answer this question, and provide search engine queries for knowledge that you need. Split the queries with ';' and end the queries with '**'. {demonstration} Question: {x} Answer: *Multiple choice QA:* Provide a better search query for web search engine to answer the given question, end the queries with '**'. {demonstration} Question: {x} Answer:

표 1: LLMs에 사용된 프롬프트 줄.

하지만 구체적인 질문들입니다. 테스트 세트에서 처음 1,000개의 샘플이 평가됩니다. (iii) PopQA (Mallen et al., 2022)는 다른 인기 있는 QA 작업보다 저인기 지식이 더 많이 포함되어 있어 긴 꼬리 분포를 포함합니다. 우리는 데이터 세트를 13,000개의 훈련 세트와 714개의 테스트 세트로 분할했습니다.

오픈 도메인 QA 벤치마크는 질문-답변 쌍의 집합으로, $\{(q, a)_i\}$로 표시됩니다. 우리는 리더와 프리즈된 리라이터 모두에 ChatGPT를 사용합니다. 평가 지표는 정확 일치(EM)와 F_1 점수입니다. RL의 보상 함수에서는 답변을 맞춘 경우 보상하고, 틀린 경우 벌점하는 지표를 사용하며, Hit로 표시됩니다. 총 보상은 EM, F_1, 및 Hit의 가중 합계입니다.

$$Hit = \begin{cases} 1 & a \text{ in } doc, \\ -1 & else \end{cases} \tag{6}$$
$$R_{lm} = EM + \lambda_f F_1 + \lambda_h Hit.$$

5.1.2 다중 선택 질의응답

다중 선택 질의응답에 대한 평가는 대규모 다중 작업 언어 이해(MMLU) (Hendrycks 외, 2021)에서 수행됩니다. 이는 인문, STEM, 사회 과학 및 기타의 4개 카테고리를 포함하는 시험 문제 데이터셋입니다. 각 카테고리는 훈련 세트 80%와 테스트 세트 20%로 나뉩니다.

다중 선택 질의응답은 $\{(q', a)_i\}$로 공식화될 수 있으며, $q' = [q, c_0, c_1, c_2, c_3]$를 포함합니다. c은 일반적으로 각 질문에 네 개씩 있는 선택지를 나타냅니다. 공식적으로 제공된 오염 목록에 포함된 검색 결과 문서는 무시됩니다. 선택지를 포함한 질문은 검색 쿼리로 다시 작성됩니다. 답은 한 개의 선택지입니다. EM는 지표로 보고되고 보상으로 사용됩니다.

$$R_{lm} = EM. \tag{7}$$

우리는 ChatGPT를 얼린 다시 쓰기 도구와 독자로서 이용합니다.

챗GPT의 요금 제한 문제 때문에 평가용 리더로 비쿠나-13B를 사용합니다. 데이터셋과 훈련 설정 등에 대한 자세한 정보는 부록에 제시되어 있습니다.

5.2 기준선

다음 설정들은 우리 방법들을 평가하고 지원하기 위해 구현되었습니다. (i) 직접: 어떤 증강 없이 표준 인컨텍스트 러닝. (ii) 검색 후 읽기: 표준 검색 증강 방법. 검색된 문서들은 질문과 연결됩니다. (iii) 얼린 LLM으로서 재작성기: §3.1에서 소개된 것처럼, 우리는 몇 개의 샷 인컨텍스트 러닝을 통해 얼린 LLM을 프롬프트하여 추론하고 쿼리를 생성합니다. (iv) 학습 가능한 재작성기: 미세 조정된 재작성기를 적용하여, 출력 쿼리들은 검색기와 리더에 의해 사용됩니다. 표 1은 프롬프트 줄 형태를 제시합니다. 예측을 위한 프롬프트가 각 작업마다 동일하게 유지된다는 점에 유의하십시오.

5.3 결과

오픈 도메인 QA에 대한 실험 결과는 표 2에 보고되어 있습니다. 세 데이터셋에 대해 쿼리 다시 쓰기는 냉동 재작성기와 훈련 가능한 재작성기 모두에서 성능 향상을 일관되게 가져옵니다. AmbigNQ 및 PopQA에서 표준 검색은 리더를 보완하여 유용한 외부 지식이 검색되었음을 나타냅니다. HotpotQA에서 표준 검색은 리더에 해를 끼칩니다. 이는 복잡한 질문을 쿼리로 사용하는 것이 매개적 지식을 보상할 수 없지만 대신 노이즈를 유발한다는 것을 보여줍니다(Mallen et al., 2022). 이는 다중 홉 질문이 웹 검색 엔진에 적합한 쿼리가 아니라는 것을 암시합니다. 다시 쓰기 단계를 추가하면 점수가 증가합니다. PopQA에서 우리의 훈련 가능한 재작성기는 표준 검색보다 뛰어나지만 LLM 재작성기보다는 열등합니다. 이는 훈련 가능한 재작성기가 유망한 접근 방식임을 나타냅니다.

distillation of query rewriting is sub-optimal.

The scores on multiple-choice QA are presented in Table 3. With ChatGPT as a reader, it can be observed that query rewriting improves the scores in most of the settings, except for the social sciences category. With Vicuna as a reader, our method achieves more gains on the four categories compared to ChatGPT. This agrees with the intuition that a more powerful reader has more parametric memories, thus more difficult to compensate with external knowledge.

Model	**EM**	**F_1**
HotpotQA		
Direct	32.36	43.05
Retrieve-then-read	30.47	41.34
LLM rewriter	32.80	43.85
Trainable rewriter	34.38	45.97
AmbigNQ		
Direct	42.10	53.05
Retrieve-then-read	45.80	58.50
LLM rewriter	46.40	58.74
Trainable rewriter	47.80	60.71
PopQA		
Direct	41.94	44.61
Retrieve-then-read	43.20	47.53
LLM rewriter	46.00	49.74
Trainable rewriter	45.72	49.51

Table 2: Metrics of open-domain QA.

MMLU	**EM**			
	Human.	STEM	Other	Social
ChatGPT				
Direct	75.6	58.8	69.0	71.6
Retrieve-then-read	76.7	63.3	70.0	78.2
LLM rewriter	77.0	63.5	72.6	76.4
Vicuna-13B				
Direct	39.8	34.9	50.2	46.6
Retrieve-then-read	40.2	39.8	55.2	50.6
LLM rewriter	42.0	41.5	57.1	52.2
Trainable rewriter	43.2	40.9	59.3	51.2

Table 3: Metrics of multiple choice QA.

6 Analysis

6.1 Training Process

The training process includes two stages, warm-up and reinforcement learning. This section shows the validation scores of the three open-domain QA datasets for further analysis. Figure 2 presents the metric scores through training iterations in the process of reinforcement learning. As the rewriting models have been warmed up on the pseudo data before RL, scores at "0 iteration" denote the ability acquired from the warm-up training.

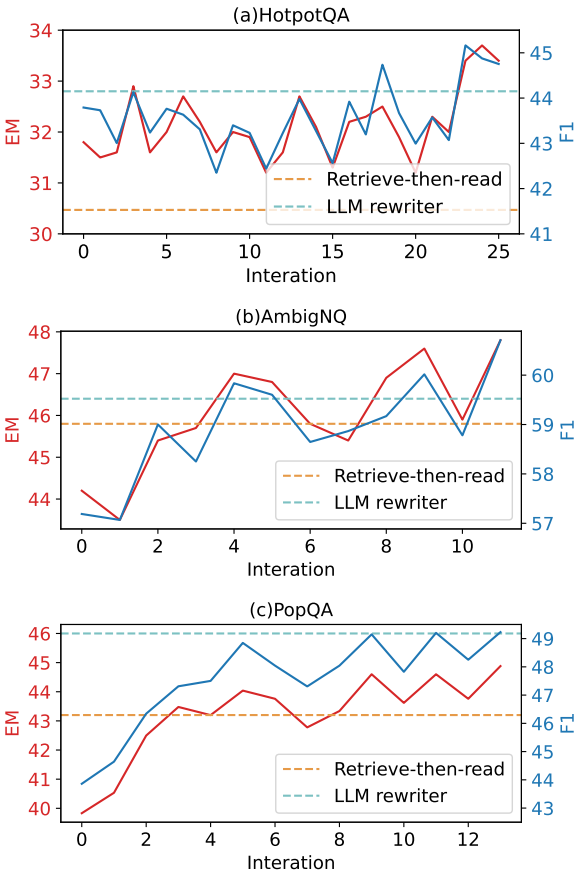


Figure 2: Reinforcement learning validation scores of (a)HotpotQA, (b)AmbigNQ, and (c)PopQA. The solid lines show EM (red) and F1 (blue) numbers through training iterations. The dashed lines are EM scores of the standard retrieve-then-read method (orange) and retrieval with an LLM as the rewriter (green).

It can be observed that the curves show upward trends with some fluctuations on all the datasets. (i) For multi-hop questions in HotpotQA, the standard retrieval is relatively weaker. Complex questions can be not specific search queries and show a larger gap from rewritten queries, i.e., the green and red lines. (ii) On AmbigNQ and PopQA, our method surpasses the baselines after several iterations (3 or 4). This indicates that the RL training stage can compensate for the insufficiency of the distillation on the pseudo data during warm-up training. (iii) In particular, on PopQA, the trainable rewriter remains inferior to the LLM rewriter. This can be explained as the dataset is constructed for adaptive retrieval (Mallen et al., 2022), which only uses retrieval where it helps to avoid harmful redundant retrieval. Thus, *"None"* is a possible query that means no retrieval. This causes more complexity and uncertainty. LLM rewriter knows better when the retrieval is needed for itself as a reader, although the rewriting step is not concatenated as

쿼리 재작성의 증류는 최적이 아닙니다.

다지선다형 QA의 점수는 표 3에 제시되어 있습니다. ChatGPT를 리더로 사용하면 쿼리 다시 쓰기가 사회과학 카테고리를 제외한 대부분의 설정에서 점수를 향상시키는 것을 관찰할 수 있습니다. Vicuna를 리더로 사용하면 ChatGPT에 비해 4개의 카테고리에서 더 많은 이득을 얻을 수 있습니다. 이는 더 강력한 리더가 더 많은 매개변수 기억을 가지고 있어 외부 지식으로 보상하기 더 어렵다는 직관에 부합합니다.

Model	**EM**	**F_1**
HotpotQA		
Direct	32.36	43.05
Retrieve-then-read	30.47	41.34
LLM rewriter	32.80	43.85
Trainable rewriter	34.38	45.97
AmbigNQ		
Direct	42.10	53.05
Retrieve-then-read	45.80	58.50
LLM rewriter	46.40	58.74
Trainable rewriter	47.80	60.71
PopQA		
Direct	41.94	44.61
Retrieve-then-read	43.20	47.53
LLM rewriter	46.00	49.74
Trainable rewriter	45.72	49.51

표 2: 오픈 도메인 QA의 지표.

MMLU	**EM**			
	Human.	STEM	Other	Social
ChatGPT				
Direct	75.6	58.8	69.0	71.6
Retrieve-then-read	76.7	63.3	70.0	78.2
LLM rewriter	77.0	63.5	72.6	76.4
Vicuna-13B				
Direct	39.8	34.9	50.2	46.6
Retrieve-then-read	40.2	39.8	55.2	50.6
LLM rewriter	42.0	41.5	57.1	52.2
Trainable rewriter	43.2	40.9	59.3	51.2

표 3: 다중 선택 QA의 지표.

6 분석

6.1 학습 과정

훈련 과정은 워밍업과 강화 학습 두 단계로 구성됩니다. 이 섹션에서는 세 개의 오픈 도메인 QA 데이터셋에 대한 추가 분석을 위해 검증 점수를 보여줍니다. 그림 2는 강화 학습 과정에서 훈련 반복을 통한 메트릭 점수를 제시합니다. 재작성 모델이 RL 전에 가짜 데이터에서 워밍업되었기 때문에 "0 반복"에서의 점수는 워밍업 훈련에서 습득한 능력을 나타냅니다.

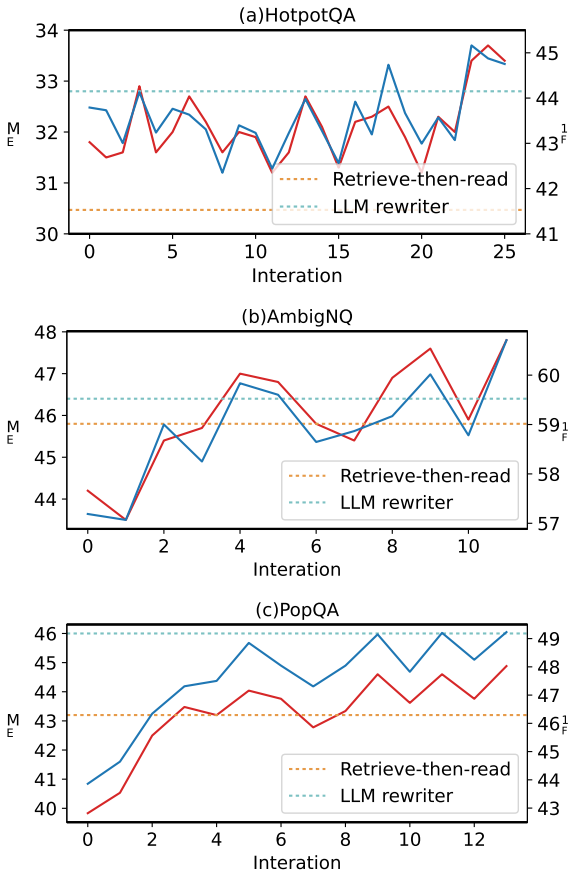


그림 2: (a)HotpotQA, (b)AmbigNQ, (c)PopQA의 강화 학습 검증 점수. 실선 그래프는 EM(빨간색)과 F1(파란색) 점수를 훈련 반복 횟수에 따라 나타냅니다. 점선 그래프는 표준 검색 후 읽기 방법(주황색)과 LLM을 리라이터로 사용한 검색 방법(초록색)의 EM 점수를 나타냅니다.

곡선들이 모든 데이터셋에서 일부 변동과 함께 상향 추세를 보이는 것을 관찰할 수 있습니다. (i) HotpotQA의 다중 점프 질문에 대해 표준 검색은 상대적으로 약합니다. 복잡한 질문은 특정 검색 쿼리가 아니며 다시 작성된 쿼리와의 격차가 더 큽니다. 즉, 녹색과 빨간 선입니다. (ii) AmbigNQ와 PopQA에서 우리 방법은 몇 번의 반복(3 또는 4) 후 기본선을 초과합니다. 이는 RL 훈련 단계가 워밍업 훈련 중 가짜 데이터에 대한 증류 부족을 보상할 수 있음을 나타냅니다. (iii) 특히 PopQA에서 훈련 가능한 다시 작성기가 LLM 다시 작성기보다 여전히 열등합니다. 이는 데이터 세트가 적응형 검색(Mallen et al., 2022)을 위해 구축되었으며, 도움이 되는 경우에만 검색을 사용하는 것으로, 해로운 중복 검색을 피할 수 있기 때문입니다. 따라서 *"None"*는 검색이 필요하지 않다는 의미를 갖는 가능한 쿼리입니다. 이는 더 많은 복잡성과 불확실성을 유발합니다. LLM 다시 작성기는 독자로서 언제 검색이 필요한지 더 잘 알고 있지만, 다시 작성 단계는 연결되지 않습니다.

the input context of the reader.

We calculate the performance of query *"None"*. The questions that can be correctly answered without retrieval (i.e., the "Direct" method) are those samples that need no more context. Comparing this retrieval-free set with those that are rewritten to be *"None"* query, the F_1 score of the LLM rewriter is 71.9% and the T5 rewriter score is 67.1%. If we consider the questions that can be correctly answered without retrieval but go wrong with retrieval as the retrieval-free set, the F_1 scores are 78.7% for LLM rewriter and 77.4% for T5.

Model	EM	F_1	Hit ratio
No retrieval	42.10	53.05	–
Upper bound	58.40	69.45	100
Retrieve-then-read			
w/ snippet	38.70	50.50	61.1
w/ BM25	45.80	58.50	76.4
LLM rewriter			
w/ snippet	39.80	52.64	63.5
w/ BM25	46.40	58.74	77.5
Trainable rewriter			
w/ BM25[2]	47.80	60.71	82.2

Table 4: Retrieval analysis on AmbigNQ.

6.2 Retrieval Result

Our proposed method is a pipeline framework, instead of an end-to-end system. The query rewriting first affects the retrieved context, then the context makes a difference to the output of the reader. Hence, QA metrics are indirect measurements. We take a closer look at the retrieved context and the reader capability through the retrieval metric, hit ratio. After text normalization, the hit rate is computed to measure whether the retrieved context contains the correct answers.

Table 4 shows the scores on AmbigNQ. The scores in the second line are computed on a selection of the samples whose retrieved contexts hit correct answers (under the standard retrieve-then-read setting). The scores show the approximate upper bound ability of the reader with retrieval augmentation, abbreviated as the "upper bound" score. The effectiveness of retrieval is proved compared to the no retrieval setting (the first line). For each retrieval method, two settings are presented: (i) collecting Bing snippets, (ii) selecting from URLs by BM25. The metrics show that content selection with BM25 recalls better documents than snippets,

[2]Our trainable rewriter is adapted to the retriever using BM25 during RL training. Using the output queries of the test set after training, the snippet hit rate is 73.4%.

Example 1: multi-hop question	Hit	Correct
Q0: The youngest daughter of Lady Mary-Gaye Curzon stars with Douglas Smith and Lucien Laviscount in what 2017 film?	✗	✗
Q1: the youngest daughter of Lady Mary-Gaye Curzon; 2017 film stars Douglas Smith and Lucien Laviscount	✓	✓
Q2: Lady Mary-Gaye Curzon youngest daughter 2017 film with Douglas Smith and Lucien Laviscount	✓	✓
Example 2:		
Q0: What 2000 movie does the song "All Star" appear in?	✗	✗
Q1: movie "All Star" 2000	✗	✗
Q2: 2000 movie "All Star" song	✓	✓
Example 3: multiple choice		
Q0: A car-manufacturing factory is considering a new site for its next plant. Which of the following would community planners be most concerned with before allowing the plant to be built? Options: A. The amount of materials stored in the plant B. The hours of operations of the new plant C. The effect the plant will have on the environment D. The work environment for the employees at the plant	✗	✗
Q1: What would community planners be most concerned with before allowing a car-manufacturing factory to be built?	✓	✓

Figure 3: Examples for intuitive illustration. Q0 denotes original input, Q1 is from the LLM rewriter, and Q2 is from the trained T5 rewriter. **Hit** means retriever recall the answer, while **Correct** is for the reader output.

while query rewriting makes progress on both settings. We also observed that the improvement in the hit rate of the retriever is more significant than the improvement in the reader. This is consistent with the findings in related search (Mallen et al., 2022; Liu et al., 2023).

6.3 Case Study

To intuitively show how the query rewriting makes a difference in the retrieved contexts and prediction performance, we present examples in Figure 3 to compare the original questions and the queries. In example 1, the original question asks for a film that *the youngest daughter of Lady Mary-Gaye Curzon* co-stars with two certain actors. Both query 1 and query 2 put the keyword *film* forward, closely following *the youngest daughter of Lady Mary-Gaye Curzon*. With both, the actress *Charlotte Calthorpe* and her movie information can be retrieved and the answer is included. The second is an example where the query from the LLM rewriter failed but

독자의 입력 맥락.

우리는 쿼리 *"None"*의 성능을 계산합니다. 검색 없이 올바르게 답변할 수 있는 질문(즉, "직접 " 방법)은 더 이상의 컨텍스트가 필요하지 않은 샘플들입니다. 검색 없이 답할 수 있는 이 집합을 검색하여 다시 작성된 쿼리인 *"None"*과 비교하면, LLM 재작성기의 F_1 점수는 71.9%이고 T5 재작성기의 점수는 67.1%입니다. 검색 없이 올바르게 답변할 수 있지만 검색으로 잘못된 질문들을 검색 없이 답변할 수 있는 집합으로 간주하면, LLM 재작성기의 F_1 점수는 78.7%이고 T5 재작성기의 점수는 77.4%입니다.

Model	**EM**	**F_1**	**Hit ratio**
No retrieval	42.10	53.05	–
Upper bound	58.40	69.45	100
Retrieve-then-read			
w/ snippet	38.70	50.50	61.1
w/ BM25	45.80	58.50	76.4
LLM rewriter			
w/ snippet	39.80	52.64	63.5
w/ BM25	46.40	58.74	77.5
Trainable rewriter			
w/ BM25[2]	47.80	60.71	82.2

표 4: AmbigNQ의 검색 분석.

6.2 검색 결과

우리가 제안하는 방법은 엔드투엔드 시스템이 아닌 파이프라인 프레임워크입니다. 쿼리 재작성은 먼저 검색된 컨텍스트에 영향을 미치고, 그 다음에 컨텍스트는 리더의 출력에 차이를 만듭니다. 따라서 QA 지표는 간접적인 측정값입니다. 우리는 검색 지표인 히트 비율을 통해 검색된 컨텍스트와 리더의 능력을 자세히 살펴봅니다. 텍스트 정규화 후, 히트 비율이 계산되어 검색된 컨텍스트가 올바른 답변을 포함하고 있는지 측정합니다.

표 4는 AmbigNQ의 점수를 보여줍니다. 두 번째 줄의 점수는 올바른 답변을 포함한 검색 맥락을 가진 샘플 집합에 대해 계산됩니다(표준 검색 후 읽기 설정 아래). 이 점수는 검색 보완을 통한 리더의 대략적인 상한 능력을 나타내며, "상한 점수"로 줄여 부릅니다. 검색 효과성은 검색이 없는 설정(첫 번째 줄)과 비교하여 입증되었습니다. 각 검색 방법에는 두 가지 설정이 제시됩니다: (i) 빙 스니펫 수집, (ii) BM25로 URL에서 선택. 이 지표는 BM25를 사용한 콘텐츠 선택이 스니펫보다 더 나은 문서를 회상한다는 것을 보여줍니다.

[2]Our trainable rewriter is adapted to the retriever using BM25 during RL training. Using the output queries of the test set after training, the snippet hit rate is 73.4%.

Example 1: multi-hop question	Hit	Correct
Q0: The youngest daughter of Lady Mary-Gaye Curzon stars with Douglas Smith and Lucien Laviscount in what 2017 film?	❌	❌
Q1: the youngest daughter of Lady Mary-Gaye Curzon; 2017 film stars Douglas Smith and Lucien Laviscount	✅	✅
Q2: Lady Mary-Gaye Curzon youngest daughter 2017 film with Douglas Smith and Lucien Laviscount	✅	✅
Example 2:		
Q0: What 2000 movie does the song "All Star" appear in?	❌	❌
Q1: movie "All Star" 2000	❌	❌
Q2: 2000 movie "All Star" song	✅	✅
Example 3: multiple choice		
Q0: A car-manufacturing factory is considering a new site for its next plant. Which of the following would community planners be most concerned with before allowing the plant to be built? Options: A. The amount of materials stored in the plant B. The hours of operations of the new plant C. The effect the plant will have on the environment D. The work environment for the employees at the plant	❌	❌
Q1: What would community planners be most concerned with before allowing a car-manufacturing factory to be built?	✅	✅

그림 3: 직관적인 설명을 위한 예시. Q0는 원본 입력을 나타내고, Q1은 LLM 재작성기에서 나온 것이며, Q2는 훈련된 T5 재작성기에서 나온 것입니다. Hit는 검색기가 답을 회상함을 의미하고, Correct는 리더의 출력을 의미합니다.

쿼리 재작성 작업이 두 설정 모두에서 진전을 보였습니다. 또한 리트리버의 히트율 개선 정도가 리더의 개선 정도보다 더 크다는 것을 관찰했습니다. 이는 관련 검색 분야의 연구 결과(Mallen 외, 2022; Liu 외, 2023)와 일치합니다.

6.3 사례 연구

쿼리 재작성이 검색된 컨텍스트와 예측 성능에 어떤 차이를 만드는지 직관적으로 보여주기 위해, 우리는 그림 3에서 원본 질문과 쿼리를 비교하는 예시를 제시합니다. 예시 1에서, 원본 질문은 두 명의 특정 배우와 함께 출연하는 영화를 요청합니다. 쿼리 1과 쿼리 2 모두 키워드 *film*를 앞세우고
the youngest daughter of Lady Mary-Gaye Curzon
를 바짝 따릅니다. 두 경우 모두 여배우 *Charlotte Calthorpe*와 그녀의 영화 정보를 검색할 수 있으며, 답변이 포함됩니다. 두 번째 예시는 LLM 재작성기로부터의 쿼리가 실패한 경우입니다.

the query from T5 gets the correct answer. The number *2000* is misunderstood in query 1, while query 2 keeps *200 movie* together, avoiding meaningless retrieval. Example 3 is for multiple choice. The query simplifies the background and enhances the keyword *community planner*. The retrieve contexts are mainly about *Introduction to Community Planning* where the answer *environment* appears several times.

7 Conclusion

This paper introduces the *Rewrite-Retrieve-Read* pipeline, where a query rewriting step is added for the retrieval-augmented LLM. This approach is applicable for adopting a frozen large language model as the reader and a real-time web search engine as the retriever. Further, we propose to apply a tuneable small language model the rewriter, which can be trained to cater to the frozen retriever and reader. The training implementation consists of two stages, warm-up and reinforcement learning. Evaluation and analyses on open-domain QA and multiple-choice QA show the effectiveness of query rewriting. Our work proposes a novel retrieval-augmented black-box LLM framework, proves that the retrieval augmentation can be enhanced from the aspect of query rewriting, and provides a new method for integrating trainable modules into black-box LLMs.

Limitations

We acknowledge the limitations of this work. (i) There is still a trade-off between generalization and specialization among downstream tasks. Adding a training process, the scalability to direct transfer is compromised, compared to few-shot in-context learning. (ii) The research line of *LLM agent* has shown impressive performance but relies on multiple calls to the LLM for each sample (Khattab et al., 2022; Yao et al., 2023), where the LLM plays as an agent to flexibly call the retriever multiple times, reads the context in earlier hops, and generates follow-up questions. Different from these studies, our motivation is to enhance the one-turn retriever-then-read framework with a trainable query rewriter. (iii) Using a web search engine as the retriever also leads to some limitations. Neural dense retrievers that are based on professional, filtered knowledge bases may potentially achieve better and controllable retrieval. More discussion is included in the appendix.

References

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*.

Parishad BehnamGhader, Santiago Miret, and Siva Reddy. 2022. Can retriever-augmented language models reason? the blame game between the retriever and the language model. *arXiv preprint arXiv:2212.09146*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Association for Computational Linguistics (ACL)*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harrison Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating large language models trained on code. *CoRR*, abs/2107.03374.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of

T5의 쿼리는 올바른 답변을 얻습니다. 쿼리 1에서는 *2000*이 잘못 이해되지만, 쿼리 2에서는 *200 movie*을 함께 유지하여 의미 없는 검색이 방지됩니다. 예시 3은 다중 선택을 위한 것입니다. 쿼리는 배경을 단순화하고 키워드 *community planner*를 강화합니다. 검색된 문맥은 주로 *Introduction to Community* 및 *Planning*에 관한 것으로, 답변 *environment*가 여러 번 나타납니다.

7 결론

이 논문에서는 검색 강화 대형 언어 모델(LLM)에 쿼리 재작성 단계를 추가한 *Rewrite-Retrieve-Read* 파이프라인을 소개합니다. 이 접근 방식은 얼린 대형 언어 모델을 리더로, 실시간 웹 검색 엔진을 리트리버로 사용하는 데 적용할 수 있습니다. 또한, 리트리버와 리더에 맞게 훈련될 수 있는 조정 가능한 소형 언어 모델을 리라이터로 적용할 것을 제안합니다. 훈련 구현은 워밍업과 강화 학습 두 단계로 구성됩니다. 개방형 QA 및 다중 선택 QA에 대한 평가 및 분석은 쿼리 재작성의 효과를 보여줍니다. 이 연구는 검색 강화 블랙박스 LLM 프레임워크를 제안하고, 검색 강화를 쿼리 재작성 측면에서 향상시킬 수 있음을 증명하며, 블랙박스 LLM에 훈련 가능한 모듈을 통합하는 새로운 방법을 제공합니다.

제한 사항

이 연구의 한계를 인정합니다. (i) 하류 작업 간 일반화와 전문화 사이의 여전히 타협이 존재합니다. 훈련 과정을 추가하면 직접 전이에 대한 확장성이 소수의 샷 인 컨텍스트 학습에 비해 손상됩니다. (ii) *LLM agent* 연구 라인은 인상적인 성능을 보여주었지만, 각 샘플에 대해 LLM에 여러 번 호출에 의존합니다 (Khattab 외, 2022; Yao 외, 2023), 여기서 LLM은 에이전트로 작용하여 리트리버를 여러 번 호출하고, 이전 홉에서 컨텍스트를 읽고, 후속 질문을 생성합니다. 이러한 연구와 달리, 저희의 동기는 훈련 가능한 쿼리 리라이터로 단일 턴 리트리버-그런 다음 읽기 프레임워크를 향상시키는 것입니다. (iii) 웹 검색 엔진을 리트리버로 사용하는 것도 일부 한계를 초래합니다. 전문적이고 필터링된 지식 기반을 기반으로 하는 신경 밀도 리트리버는 더 나은 및 제어 가능한 검색을 달성할 수 있습니다. 추가 논의는 부록에 포함되어 있습니다.

참조

방예진, 사무엘 카야와이와야, 이나연, 대문량, 서단, 브라이언 윌리, 홀리 로베니아, 기자위, 유철정, 정월리, 도결열, 서언, 펑파스칼. 2023. 챗GPT에 대한 다중 작업, 다국어, 다모달 평가: 추론, 환각 및 상호작용. *arXiv preprint arXiv:2302.04023*.

파리샤드 베흐 가넨티아고 미레트, 시바 레디. 2022. 검색기 증강 언어 모델은 추론할 수 있는가? 검색기와 언어 모델 사이의 책임 전가 게임. *arXiv preprint arXiv:2212.09146*.

톰 브라운, 벤자민 만, 닉 라이더, 멜라니 수브비아, 제러드 D 카플란, 프라풀라 다리왈, 아르빈드 네엘라칸탄, 프라나브 샤임, 기리쉬 사스트리, 아만다 아셀, 외. 2020. 언어 모델은 소량 학습자다. *Advances in neural information processing systems*, 33 :1877–1901.

전단기, 아담 피쉬, 제이슨 웨스턴, 그리고 앙트완 보르드. 2017. 위키피디어를 읽어 오픈 도메인 질문에 답하기. *Association for Computational Linguistics (ACL)*에서.

마크 첸, 제리 투워크, 허우준, 원치밍, 헨리케 폰데 드 올리베이라 핀토, 제라드 카플란, 해리슨 에드워즈, 유리 부르다, 니콜라스 조셉, 그렉 브록먼, 알렉스 레이, 라울 푸리, 그레첸 크루거, 마이클 페트로프, 하이디 클라프, 기리쉬 사스트리, 패멀라 미쉬킨, 브룩 챈, 스콧 그레이, 닉 라이더, 미하일 파블로프, 아레테아 파워, 루카스 카이저, 모하메드 바바리안, 클레멘스 윈터, 필리프 틸레, 펠리페 페트로스키 수크, 데이브 커밍스, 마티아스 플랩퍼트, 포티오스 찬츠리스, 엘리자베스 반스, 아리엘 허버트-보스, 윌리엄 헤브겐 거스, 알렉스 니콜, 알렉스 파노, 니코라스 테자크, 제리 탕, 이고르 바부슈킨, 수치르 발라지, 산탄누 자인, 윌리엄 사우더스, 크리스토퍼 헤세, 앤드류 N. 카, 잔 라이크, 조슈아 아치암, 베단트 미스라, 에반 모리카와, 알렉 라포드, 매튜 나이트, 마일스 브런데이지, 미라 무라티, 케이티 메이어, 피터 웰린더, 밥 맥그루, 다리오 아모데아, 샘 맥캔들리시, 일리야 수트스케버, 그리고 보이치에흐 자렘바. 2021. 코드에 훈련된 대규모 언어 모델 평가. *CoRR*, abs/2107.03374.

웨이린 장, 주한 리, 자 린, 잉 생, 장하오 우, 하오 장, 리안민 정, 시위안 주앙, 용하오 주앙, 조셉 E. 곤잘레스, 이온 스토이카, 에릭 P. 싱. 2023. 비쿠나: GPT-4를 감동시키는 90%* 챗GPT 품질의 오픈 소스 채팅봇.

아칸크샤 초우드리, 샤란 나랑, 제이콥 데블린, 마튼 보스마, 가우라브 미쉬라, 아담 로버츠, 폴 바햄, 정형원, 찰스 서튼, 세바스티안 게르만 등. 2022. 팜: 경로에 따른 언어 모델링 확장. *arXiv preprint arXiv:2204.02311*.

제이콥 데블린, 밍웨이 창, 켄턴 리, 그리고 크리스티나 투타나바. 2019. BERT: 사전 학습의

deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

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. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Namgyu Ho, Laura Schmid, and Se-Young Yun. 2022. Large language models are reasoning teachers. *arXiv preprint arXiv:2212.10071*.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander J. Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. *ArXiv*, abs/2305.02301.

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.

Joel Jang, Seonghyeon Ye, Changho Lee, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun Kim, and Minjoon Seo. 2022. Temporalwiki: A lifelong benchmark for training and evaluating ever-evolving language models.

Zhengbao Jiang, Luyu Gao, Jun Araki, Haibo Ding, Zhiruo Wang, Jamie Callan, and Graham Neubig. 2022. Retrieval as attention: End-to-end learning of retrieval and reading within a single transformer. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Abu Dhabi, UAE.

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, Online. Association for Computational Linguistics.

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*.

Mojtaba Komeili, Kurt Shuster, and Jason Weston. 2022. Internet-augmented dialogue generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8460–8478, Dublin, Ireland. Association for Computational Linguistics.

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*.

Angeliki Lazaridou, Elena Gribovskaya, Wojciech Stokowiec, and Nikolai Grigorev. 2022. Internet-augmented language models through few-shot prompting for open-domain question answering. *arXiv preprint arXiv:2203.05115*.

Haejun Lee, Akhil Kedia, Jongwon Lee, Ashwin Paranjape, Christopher Manning, and Kyoung-Gu Woo. 2022. You only need one model for open-domain question answering. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3047–3060, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics.

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

Zekun Li, Baolin Peng, Pengcheng He, Michel Galley, Jianfeng Gao, and Xifeng Yan. 2023. Guiding large language models via directional stimulus prompting. *arXiv preprint arXiv:2302.11520*.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*.

Kelvin Luu, Daniel Khashabi, Suchin Gururangan, Karishma Mandyam, and Noah A. Smith. 2022. Time waits for no one! analysis and challenges of temporal misalignment. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5944–5958, Seattle, United States. Association for Computational Linguistics.

깊은 양방향 트랜스포머를 이용한 언어 이해. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, 4171-4186쪽. 계산언어학회.

켈빈 구, 켄턴 리, 조라 둥, 파누퐁 파수-팟, 및 밍웨이 창. 2020. 검색 강화 언어 모델 사전 학습. *International confer- ence on machine learning*에서, 페이지 3929-3938. PMLR.

댄 헨드릭스의, 콜린 번스, 스티븐 바사르트, 앤디 주, 만타스 마제카, 던 송, 그리고 제이콥 스틴하르트. 2021. 대규모 다중 작업 언어 이해 측정. *Proceedings of the International Conference on Learning Representations (ICLR)*.

호남규, 라우라 슈미드, 윤세영. 2022. 대규모 언어 모델은 추론 교사이다. *arXiv preprint arXiv:2212.10071*.

성유 셰, 춘량 리, 지관 예, 후탄 나코스트, 야스히사 후지이, 알렉산더 J. 래트너, 란제이 크리슈나, 진유 리, 토마스 피스터. 2023. 단계별 증류! 더 적은 훈련 데이터와 더 작은 모델 크기로 더 큰 언어 모델을 능가하는 방법. *ArXiv*, abs/2305.02301.

고티에 이자카드, 패트릭 루이스, 마리아 로멜리, 루카스 호세이니, 파비오 페트로니, 티모 슈릭, 제인 드위베디-유, 아르망 주랭, 세바스찬 리델, 에드와르 그라브. 2022. 몇 번의 촬영으로 학습하는 검색 증강 언어 모델.

조엘 장, 성현 예, 창호 이, 소희 양, 중보 신, 장훈 한, 경훈 김, 그리고 민준 서. 2022. Temporalwiki: 지속적으로 진화하는 언어 모델을 훈련하고 평가하기 위한 평생 벤치마크.

정보 강, 루유 고, 준 아라키, 하보 딩, 지루오 왕, 제이미 콜란, 그리고 그레이엄 노비그. 2022. Retrieval as attention: 단일 트랜스포머 내에서의 검색 및 읽기 학습. *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 아부다비, UAE.

블라디미르 카르푸힌, 바르라스 오구즈, 세원 민, 패트릭 루이스, 레델 우, 세르게이 에두노프, 단치 첸, 그리고 웬타우 이. 2020. 개방형 도메인 질의 응답을 위한 밀집 문단 검색. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*에서, 6769–6781쪽, 온라인. 계산 언어학 협회.

오마르 카타브, 케샤브 산타나무, 리사 리 샹, 데이비드 홀, 퍼시 리앙, 크리스토퍼 포츠, 그리고 마테이 자하리아. 2022. 시연-검색-예측: 지식 집중적 NLP를 위한 검색 및 언어 모델의 구성. *arXiv preprint arXiv:2212.14024*.

모자타바 코메이리, 커트 슈스터, 제이슨 웨스턴. 2022. 인터넷 증강 대화 생성. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*에서, 페이지 8460-8478, 아일랜드 더블린. 계산 언어학 협회.

톰 콧카우스키, 제니마리아 팔로마키, 올리비아 레드필드, 마이클 콜린스, 안쿠르 파릭, 크리스 알베르티, 다니엘 에프스타인, 일리아 폴로수힌, 제이콥 데블린, 켄턴 리 등. 2019. 자연 질문: 질문 답변 연구의 벤치마크. *Transactions of the Association for Computational Linguistics*.

안젤리키 라자리두, 엘레나 그리보프스카야, 보이치에흐 스토코비에츠, 니콜라이 그리고레프. 2022. 인터넷 증강 언어 모델: 개방형 도메인 질의 응답을 위한 소규모 샘플 프롬프트. *arXiv preprint arXiv:2203.05115*.

이해준, 아킬 케디아, 이종원, 아슈윈 파라자페, 크리스토퍼 매닝, 그리고 우경구. 2022. 오픈 도메인 질의 응답에는 하나의 모델만 필요합니다. *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 3047-3060쪽, 아부다비, 아랍에미리트. 계산 언어학 협회.

마이크 루이스, 유인한 류, 나만 고얄, 마르잔 가즈비니자데, 압델라만 모하메드, 오머 레비, 베셀린 스토야노프, 루크 제틀모이어. 2020a. BART: 자연어 생성, 번역 및 이해를 위한 소음 제거 시퀀스-투-시퀀스 사전 훈련. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*에서, 페이지 7871-7880. 계산 언어학 협회.

패트릭 루이스, 에단 페레스, 알렉산드라 픽투스, 파비오 페트로니, 블라디미르 카르푸킨, 나만 고얄, 하인리히 퀴틀러, 마이크 루이스, 웬타우 이, 팀 록타셸 외. 2020b. 지식 집약적 NLP 과제를 위한 검색 강화 생성. *Advances in Neural Information Processing Systems*, 33:9459–9474.

리제쿤, 펑바올린, 허펑청, 미셸 갤리, 가오젠펑, 그리고 얀시펑. 2023. 방향성 자극 프롬프트를 통한 대형 언어 모델 가이드. *arXiv preprint arXiv:2302.11520*.

넬슨 F 리우, 케빈 린, 존 휴잇, 애슈윈 파라자페, 미켈레 베빌라쿠아, 파비오 페트로니, 퍼시 리앙. 2023. 중간에 길을 잃다: 언어 모델이 긴 맥락을 어떻게 사용하는가. *arXiv preprint arXiv:2307.03172*.

켈빈 루, 다니엘 카샤비, 수친 구루라잔, 카리스마 만디암, 그리고 노아 A. 스미스. 2022. 시간은 아무도 기다리지 않는다! 시간적 불일치의 분석과 도전. *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 5944-5958쪽, 미국 시애틀. 계산 언어학 협회.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Hannaneh Hajishirzi, and Daniel Khashabi. 2022. When not to trust language models: Investigating effectiveness and limitations of parametric and non-parametric memories. *arXiv preprint*.

Jacob Menick, Maja Trebacz, Vladimir Mikulik, John Aslanides, Francis Song, Martin Chadwick, Mia Glaese, Susannah Young, Lucy Campbell-Gillingham, Geoffrey Irving, et al. 2022. Teaching language models to support answers with verified quotes. *arXiv preprint arXiv:2203.11147*.

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? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering ambiguous open-domain questions. In *EMNLP*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

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 preprint arXiv:2210.03350*.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. 2023. Toolllm: Facilitating large language models to master 16000+ real-world apis. *ArXiv preprint*, abs/2307.16789.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Rajkumar Ramamurthy, Prithviraj Ammanabrolu, Kianté Brantley, Jack Hessel, Rafet Sifa, Christian Bauckhage, Hannaneh Hajishirzi, and Yejin Choi. 2022. Is reinforcement learning (not) for natural language processing?: Benchmarks, baselines, and building blocks for natural language policy optimization.

Paul Röttger and Janet Pierrehumbert. 2021. Temporal adaptation of BERT and performance on downstream document classification: Insights from social media. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2400–2412, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Devendra Singh Sachan, Siva Reddy, William L. Hamilton, Chris Dyer, and Dani Yogatama. 2021. End-to-end training of multi-document reader and retriever for open-domain question answering. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 25968–25981.

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*.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. 2015. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. Hugginggpt: Solving ai tasks with chatgpt and its friends in huggingface. *arXiv preprint arXiv:2303.17580*.

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*.

Kurt Shuster, Mojtaba Komeili, Leonard Adolphs, Stephen Roller, Arthur Szlam, and Jason Weston. 2022. Language models that seek for knowledge: Modular search & generation for dialogue and prompt completion. In *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 373–393. Association for Computational Linguistics.

Hongjin Su, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, et al. 2022. Selective annotation makes language models better few-shot learners. *arXiv preprint arXiv:2209.01975*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *CoRR*, abs/2203.11171.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for

알렉스 말렌, 아카리 아사아, 빅토르 중, 라자르시 다스, 하난네 하지시르지, 그리고 다니엘 카샤비. 2022. 언어 모델을 신뢰하지 말아야 할 때: 매개변수 및 비매개변수 메모리의 효과와 한계 조사. *arXiv preprint*.

제이콥 메닉, 마야 트레바츠, 블라디미르 미쿨릭, 존 아슬란디스, 프랜시스 송, 마틴 챈드윅, 미아 글레이즈, 수잔나 영, 루시 캠벨-길링엄, 제프리 어빙 외. 2022. 언어 모델에게 검증된 인용구로 답변을 지원하는 방법을 가르치기. *arXiv preprint arXiv:2203.11147*.

민세원, 류신시, 아리 홀츠만, 미켈 아르테체, 마이크 루이스, 한나하 지르지, 그리고 루크 제틀-모이어. 2022. 시연의 역할 재고: 컨텍스트 내 학습이 작동하게 하는 것은 무엇인가? *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*에서, 11048-11064쪽, 아부다비, 아랍에미리트. 계산 언어학 협회.

민세원, 줄리안 마이클, 한나네 하지시르지, 그리고 루크 제틀모이어. 2020. AmbigQA: 모호한 개방형 질문에 답하기. *EMNLP*에서.

오우양 롱, 제프리 우, 쉬 강, 디오고 알메이다, 캐롤 와인라이트, 패멀라 미쉬킨, 장 충, 산히니 아가왈, 카타리나 슬라마, 알렉스 레이 외. 2022. 인간 피드백을 사용하여 언어 모델을 지시사항에 따르도록 훈련시키기. *Advances in Neural Information Processing Systems*, 35:27730–27744.

오피르 프레스, 무루 장, 세원 민, 루도비크 슈미트, 노아 A 스미스, 마이크 루이스. 2022. 언어 모델의 구성성 격차를 측정하고 줄이기. *arXiv preprint arXiv:2210.03350*.

진유가, 양시호, 예이닝, 주쿤룬, 옌란, 루야시, 린얀 카이, 콩신, 당샹루, 천빌, 외. 2023. Toolllm: 대규모 언어 모델이 16000+ 실제 API를 마스터하도록 지원. abs/2307.16789.

콜린 라펠, 노암 샤저, 아담 로버츠, 캐서린 리, 샤란 나랑, 마이클 마테나, 잔치 주, 웨이 리, 그리고 피터 J. 류. 2020. 텍스트-투-텍스트 트랜스포머를 통한 전이 학습의 한계 탐구. *Journal of Machine Learning Research*, 21(140):1–67.

라자쿠마르 라마무르티, 프리트비라지 아만나브로루, 키안테 브랜틀리, 잭 헤셀, 라페트 시파, 크리스티안 바우크하게, 하난네 하지시르지, 그리고 예진 최. 2022. 강화 학습(이) 자연어 처리에 적합한가?: 자연어 정책 최적화를 위한 벤치마크, 기준선, 그리고 구성 요소.

폴 뢰트거와 자넷 피에르험버트. 2021. 소셜 미디어에서 BERT의 시간적 적응과 하류 문서 분류 성능: 도미니카 공화국 푸에르토 플라타에서 열린 *Findings of the Association for Computational Linguistics: EMNLP 2021*에서 2400-2412 페이지. 계산 언어학 협회.

데벤드라 싱 사찬, 시바 레디, 윌리엄 L. 해밀턴, 크리스 다이어, 그리고 다니 요가타마. 2021. 개방형 도메인 질의 응답을 위한 다중 문서 리더 및 리트리버의 엔드-투-엔드 훈련. *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, 25968-25981쪽.

티모 슈릭, 제인 드위베디-유, 로베르토 데시, 로베르타 라일레누, 마리아 로멜리, 루크 제틀레모어, 니콜라 칸체다, 토마스 시알롬. 2023. Toolformer: 언어 모델은 도구를 사용하는 방법을 스스로 배울 수 있다. *arXiv preprint arXiv:2302.04761*.

존 슐만, 필립 모리츠, 세르게이 레빈, 마이클 조던, 그리고 피터 아벨. 2015. 일반화된 이점 추정치를 사용한 고차원 연속 제어. *arXiv preprint arXiv:1506.02438*.

존 슐만, 필립 볼스키, 프라풀라 다리왈, 알렉 래드포드, 그리고 올레그 클리모프. 2017. 근사 정책 최적화 알고리즘. *arXiv preprint arXiv:1707.06347*.

신용량, 송개도, 탄허, 이동생, 루위명, 및 장월정. 2023. Hugging-gpt: 챗GPT와 허그페이스 친구들을 사용하여 AI 작업 해결. *arXiv preprint arXiv:2303.17580*.

시위지아, 민세원, 야스나가 미치히로, 서민준, 리치 제임스, 마이크 루이스, 루크 제틀모어, 그리고 이웬 타오 이. 2023. Replug: 검색을 보강한 블랙박스 언어 모델. *arXiv preprint arXiv:2301.12652*.

커트 슈스터, 모자타 코메이리, 레너드 아돌프스, 스티븐 롤러, 아서 슬램, 제이슨 웨스트온. 2022. 지식을 찾는 언어 모델: 대화 및 프롬프트 완료를 위한 모듈식 검색 및 생성. *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, 373-393쪽. 계산 언어학 협회.

홍진 수, 정고 카사이, 첸 헨리 우, 위지아 시, 천루 왕, 지아신 신, 루이 장, 마리 오스텐도르프, 루크 제틀레모어, 노아 A 스미스 외. 2022. 선택적 주석은 언어 모델을 더 나은 소샘 학습자로 만듭니다. *arXiv preprint arXiv:2209.01975*.

왕설지, 제이슨 웨이, 데일 슈어먼스, 꾸옥 V. 레, 에드 H. 치, 그리고 데니 주. 2022. 자기 일관성은 언어 모델의 사고 사슬 추론에 개선된다. *CoRR*, abs/2203.11171.

제이슨 웨이, 설지 왕, 데일 슈어먼스, 마튼 보스마, 브라이언 이처, 소화 페리, 에드 H. 치, 콴크 V. 레, 그리고 데니 주. 2022. 사고 과정을 유도하는 체인-오브-생각 프롬프트는 대규모 언어 모델에서 추론을 유도한다. *NeurIPS*에서.

양지린, 치펑, 장사이정, 조슈아 벤지오, 윌리엄 코헨, 루슬란 살라후트디노프, 크리스토퍼 D. 매닝. 2018. HotpotQA: 데이터셋으로

diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2023. Generate rather than retrieve: Large language models are strong context generators. In *International Conference for Learning Representation (ICLR)*.

Tianjun Zhang, Xuezhi Wang, Denny Zhou, Dale Schuurmans, and Joseph E Gonzalez. 2023a. Tempera: Test-time prompt editing via reinforcement learning. In *The Eleventh International Conference on Learning Representations*.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2023b. Automatic chain of thought prompting in large language models. In *The Eleventh International Conference on Learning Representations (ICLR 2023)*.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

A Warm-up Dataset

For the warm-up training of the tuneable rewriter, we construct a pseudo dataset for the query rewriting task. For benchmarks that provide official training and test splits (HotpotQA and AmbigNQ), we use the whole training set. For those that have no official splits (PopQA and MMLU), we randomly split the full dataset. In detail, PopQA contains 16 types of questions, thus split into 13k for training and 714 for testing following stratified sampling. For MMLU, each of the 4 categories is randomly split into 80% for the training set and 20% for the test set. Then the training sets of each benchmark are used to derive the pseudo dataset for the query rewriting, i.e., $D_{Train} = \{(x, \tilde{x}) | \hat{y} = y\}$. We present the statistics of the splits and warm-up dataset in Table 5.

B Setup Details

For warm-up, we train the T5-large with 3e-5 learning rate, {16, 20} batch size, for {6,8,12} epochs. For reinforcement learning, we set the sampling steps to 5120, 10 threads, 512 steps for each. After sampling, the policy network is trained for {2,3,4} epochs, with learning rate as 2e-6 and batch size as {8,16}. λ_f and λ_h are 1.0. β in Eq. 4 is dynamically adapted according to Ramamurthy et al. (2022); Ziegler et al. (2019),

$$e_t = \text{clip}\left(\frac{\text{KL}\left(\pi \| \pi_0\right) - \text{KL}_{\text{target}}}{\text{KL}_{\text{target}}}, -0.2, 0.2\right),$$
$$\beta_{t+1} = \beta_t \left(1 + \text{K}_\beta e_t\right),$$

where KL_{target} is set to 0.2, K_β is set to 0.1. β_0 is initialized to be 0.001. The generation strategy follows the 4-beam search and returns the one sequence. In the implementation of the BM25-based retriever, the textboxes from searched URLs are parsed from HTML code. We compute BM25 scores between the paragraph from each textbox and the query following the scikit-learn package, then keep those with higher scores until the reserved context reaches a max length. In reinforcement learning, the results of AmbigNQ are with the BM25 method, while others use snippets as context.

Task	Training Set	Warm-up	Test Set
HotpotQA	90.4k	37.5k	7.4k
AmbigNQ	19.4k	8.6k	1k
PopQA	13.0k	6.0k	0.7k
Humanities	3.8k	1.5k	0.9k
STEM	2.4k	0.9k	0.6k
Other	2.6k	1.3k	0.6k
Social Science	2.4k	1.3k	0.6k

Table 5: Metrics of multiple choice QA.

C Web Search: Tool Use

Our proposed pipeline integrates an externally built web search engine as the retriever module. We present more discussion on the advantages and disadvantages here.

The usage of external tools expands the ability boundary of language models, compensating for the parametric knowledge, and grounding the capabilities of language models to interact with environments (Qin et al., 2023; Schick et al., 2023). Recent studies show a trend to leverage plug-and-play tools like search engines to enhance language agents (Lazaridou et al., 2022; Menick et al., 2022; Shuster et al., 2022; Shen et al., 2023). Search engine APIs are well-developed retrievers, saving efforts to build and maintain another retriever, like a Contriever. Accessible to the whole Internet, the web search retrieves from a wide-range, up-to-date

다양하고 설명 가능한 다중 점프 질의 응답. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* 2369-2380쪽, 브뤼셀, 벨기에. 계산 언어학 협회.

요순유, 제프리 조, 유전, 두난, 이작 샤프란, 카르틱 나스림한, 그리고 조원. 2023. ReAct: 언어 모델에서 추론과 행동의 시너지 효과. *International Conference on Learning Representations (ICLR)*에서.

유문호, 단 이터, 왕서항, 허이치옹, 주명언, 소움야 산얄, 주성광, 마이클 젠그, 그리고 강멍. 2023. 검색보다 생성: 대규모 언어 모델은 강력한 맥락 생성기이다. *International Conference for Learning Representation (ICLR)*에서.

장천준, 왕설지, 주든니, 슈어만스 데일, 그리고 조셉 E 곤잘레스. 2023a. Tempera: 강화 학습을 통한 테스트 시간 프롬프트 편집. *The Eleventh International Conference on Learning Representations*에서.

장조생, 장스톤, 리무, 그리고 알렉스 스몰라. 2023b. 대규모 언어 모델에서 자동 사고 사슬 프롬프트. *The Eleventh International Conference on Learning Representations (ICLR 2023)*에서.

다니엘 M 지글러, 니산 스텐넌, 제프리 우, 톰 B 브라운, 앨릭 래드포드, 다리오 아모데이, 폴 크리스티아노, 그리고 제프리 어빙. 2019. 인간 선호도에 따른 언어 모델 미세 조정. *arXiv preprint arXiv:1909.08593*.

워밍업 데이터셋

튜닝 가능한 리라이터의 워밍업 훈련을 위해, 우리는 쿼리 리라이팅 작업을 위한 가짜 데이터셋을 구성합니다. 공식 훈련 및 테스트 분할을 제공하는 벤치마크(HotpotQA 및 AmbigNQ)의 경우, 전체 훈련 세트를 사용합니다. 공식 분할이 없는 벤치마크(PopQA 및 MMLU)의 경우, 전체 데이터셋을 무작위로 분할합니다. 자세히 말하자면, PopQA는 16가지 유형의 질문을 포함하고 있으므로, 층화 표본 추출을 따라 13k를 훈련용, 714를 테스트용으로 분할합니다. MMLU의 경우, 4개의 각 카테고리를 무작위로 80%는 훈련 세트, 20%는 테스트 세트로 분할합니다. 그런 다음 각 벤치마크의 훈련 세트는 쿼리 리라이팅을 위한 가짜 데이터셋, 즉 $D_{Train} = \{(x, \tilde{x}) | \hat{y} = y\}$을 유도하는 데 사용됩니다. 분할 및 워밍업 데이터셋의 통계를 표 5에 제시합니다.

B 설정 세부 정보

워밍업을 위해, 우리는 3e-5 학습률로 T5-large를 훈련시키고, {16, 20} 배치 크기로 {6,8,12} 에포크 동안 훈련시킵니다. 강화 학습을 위해, 우리는 샘플링을 설정합니다.

Task	Training Set	Warm-up	Test Set
HotpotQA	90.4k	37.5k	7.4k
AmbigNQ	19.4k	8.6k	1k
PopQA	13.0k	6.0k	0.7k
Humanities	3.8k	1.5k	0.9k
STEM	2.4k	0.9k	0.6k
Other	2.6k	1.3k	0.6k
Social Science	2.4k	1.3k	0.6k

표 5: 다중 선택 QA의 지표.

5120걸음, 10스레드, 각각 512걸음. 샘플링 후, 정책 네트워크는 {2,3,4} 에포크 동안 학습률 2e-6과 배치 크기 {8,16}으로 학습됩니다. λ_f와 λ_h은 1.0입니다. 식 4의 β은 Ramamurthy 외 (2022); Ziegler 외 (2019)에 따라 동적으로 조정됩니다.

$$e_t = \text{clip}\left(\frac{\text{KL}\left(\pi \| \pi_0\right) - \text{KL}_{\text{target}}}{\text{KL}_{\text{target}}}, -0.2, 0.2\right),$$
$$\beta_{t+1} = \beta_t \left(1 + \text{K}_\beta e_t\right),$$

KL_{target}는 0.2로 설정되고, K_β는 0.1로 설정됩니다. β_0은 0.001로 초기화됩니다. 생성 전략은 4-빔 검색(4-beam search)을 따르고 하나의 시퀀스를 반환합니다. BM25 기반 리트리버(retriever)의 구현에서, 검색된 URL의 텍스트 박스들은 HTML 코드에서 파싱됩니다. 우리는 scikit-learn 패키지를 따라 각 텍스트 박스의 단락과 쿼리 사이의 BM25 점수를 계산하고, 예약된 컨텍스트가 최대 길이에 도달할 때까지 더 높은 점수를 가진 것들을 유지합니다. 강화 학습에서 AmbigNQ의 결과는 BM25 방법과 함께 사용되며, 다른 것들은 스니펫을 컨텍스트로 사용합니다.

C 웹 검색: 도구 사용

저희가 제안하는 파이프라인은 외부에서도 구축된 웹 검색 엔진을 리트리버 모듈로 통합합니다. 여기에서 장점과 단점에 대한 더 많은 토론을 제시합니다.

외부 도구의 사용은 언어 모델의 능력 범위를 확장하고, 매개변수 지식을 보완하며, 언어 모델이 환경과 상호작용할 수 있는 능력을 구체화합니다 (Qin et al., 2023; Schick et al., 2023). 최근 연구들은 검색 엔진과 같은 플러그앤플레이 도구를 활용하여 언어 에이전트를 향상시키는 추세를 보입니다 (Lazaridou et al., 2022; Menick et al., 2022; Shuster et al., 2022; Shen et al., 2023). 검색 엔진 API는 잘 개발된 검색기로, Contriever와 같은 별도의 검색기를 구축하고 유지 관리하는 노력을 절약할 수 있습니다. 인터넷 전체에 접근할 수 있는 웹 검색은 광범위하고 최신 정보로 검색합니다.

knowledge base. The temporal misalignment problem on a fixed candidate database can be alleviated.

On the other hand, web search APIs are commercial products requiring subscriptions. Also, the vast amount of knowledge on the web can be difficult to control. The retrieved context from the Internet can be occasionally inconsistent, redundant, and toxic, which hinders the LLM reader.

Beyond retrieval augmentation, in a general scope, other tools called by LLMs, like code interpreters, online models, and expert applications, are all similar to search engines, without trainable parameters to optimize. There could be a gap between the LM and these tools. This paper proposes an idea to align them through a trainable small model.

지식 베이스. 고정된 후보 데이터베이스에서 시간 불일치 문제를 완화할 수 있습니다.

반면, 웹 검색 API는 구독이 필요한 상업적 제품입니다. 또한 웹상의 방대한 지식을 통제하는 것은 어려울 수 있습니다. 인터넷에서 검색한 문맥은 가끔 일관성이 없고, 중복되며, 유해할 수 있어 LLM 리더의 성능을 저해할 수 있습니다.

검색 강화 이상의 범위에서, LLM이 호출하는 코드 해석기, 온라인 모델 및 전문가 애플리케이션과 같은 다른 도구는 모두 훈련 가능한 최적화 매개변수가 없는 검색 엔진과 유사합니다. LM과 이러한 도구 사이에 격차가 있을 수 있습니다. 이 논문은 훈련 가능한 소형 모델을 통해 그들을 정렬하는 아이디어를 제안합니다.