Universiteit Leiden
The Netherlands

Generate-then-ground in retrieval-augmented generation for multi-hop question answering

Shi, Z.; Zhang, S.; Sun, W.; Gao, S.; Ren, P.; Chen, Z.; ... ; Srikumar, V.

Citation

Shi, Z., Zhang, S., Sun, W., Gao, S., Ren, P., Chen, Z., & Ren, Z. (2024). Generate-then-ground in retrieval-augmented generation for multi-hop question answering. *Proceedings Of The 62Nd Annual Meeting Of The Association For Computational Linguistics*, 7339–7353. doi:10.18653/v1/2024.acl-long.397

Version: Publisher's Version

License: Creative Commons CC BY 4.0 license

Downloaded from: https://hdl.handle.net/1887/4176754

Note: To cite this publication please use the final published version (if applicable).

Generate-then-Ground in Retrieval-Augmented Generation for Multi-hop Question Answering

Zhengliang Shi[1] **Shuo Zhang**[2] **Weiwei Sun**[1] **Shen Gao**[3]
Pengjie Ren[1] **Zhumin Chen**[1] **Zhaochun Ren**[4*]
[1]Shandong University, Qingdao, China [2]Bloomberg, London, United Kingdom
[3]University of Electronic Science and Technology of China, Chengdu, China
[4]Leiden University, Leiden, The Netherlands
shizhl@mail.sdu.edu.cn szhang611@bloomberg.net
sunnweiwei@gmail.com z.ren@liacs.leidenuniv.nl

Abstract

Multi-Hop Question Answering (MHQA) tasks present a significant challenge for large language models (LLMs) due to the intensive knowledge required. Current solutions, like Retrieval-Augmented Generation, typically *retrieve* potential documents from an external corpus to *read* an answer. However, the performance of this *retrieve-then-read* paradigm is constrained by the retriever and the inevitable noise in the retrieved documents. To mitigate these challenges, we introduce a novel *generate-then-ground* (GenGround) framework, synergizing the parametric knowledge of LLMs and external documents to solve a multi-hop question. GenGround empowers LLMs to alternate two phases until the final answer is derived: (1) formulate a simpler, single-hop question and directly generate the answer; (2) ground the question-answer pair in retrieved documents, amending any wrong predictions in the answer. We also propose an instructional grounding distillation method to generalize our method into smaller models. Extensive experiments conducted on four datasets illustrate the superiority of our method.

1 Introduction

Multi-Hop Question Answering (MHQA) tasks (Yang et al., 2018) require multi-hop reasoning using intensive knowledge to derive the answer (Xu et al., 2024). It has been widely employed in various practical scenarios and domains (Mavi et al., 2022). To answer a multi-hop question, most prior work integrates large language models (LLMs) with information retrieval techniques, following a *retrieve-then-read* paradigm (Shao et al., 2023). As illustrated in Figure 1, the initial step employs the LLMs to break down the complex question and formulate a series of simpler, single questions in a step-by-step manner. For each step, the LLMs are guided to derive an answer from the relevant documents, which are retrieved using the formulated question (Yao et al., 2022).

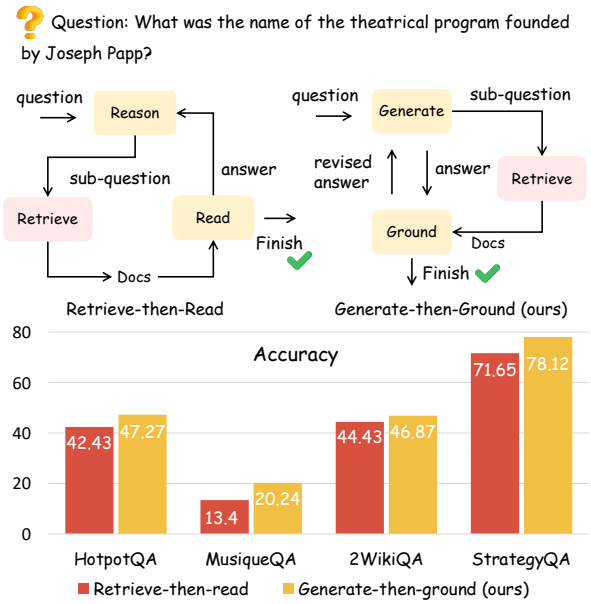

Figure 1: The top block depicts the comparison with the commonly-used *retrieve-then-read* paradigm in MHQA task. The bottom block provides the performance of our method and baselines in four MHQA benchmarks.

Despite the progress of the *retrieve-then-read paradigm*, it faces *two challenges* in practice. First, its effectiveness is constrained by the performance of the retriever (Yu et al., 2022; Xu et al., 2024). Given that the answer can only be derived from the retrieved documents, the inherent world knowledge of LLMs is overlooked. This limitation is particularly magnified in multi-hop QA tasks that frequently require complex logical reasoning (Mavi et al., 2022). The retrievers may struggle to retrieve all necessary documents to answer the question, leading to a performance decline using this paradigm (Yu et al., 2022; Abdallah and Jatowt, 2023). Second, the retrieved documents

* Corresponding author.

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 7339–7353
August 11-16, 2024 ©2024 Association for Computational Linguistics

inevitably contain irrelevant or plausible-looking statements (Gao et al., 2023b; Jiang et al., 2023b). Directly incorporating them into the chain-of-reasoning of LLMs may mislead the LLMs to generate incorrect or irrelevant responses (Adlakha et al., 2024; Thakur et al., 2023b). Therefore, developing an adaptive method for utilizing the retrieved documents remains an active research area.

Inspired by the extensive knowledge and powerful deduction capability of LLMs, *e.g.,* ChatGPT or GPT-4, we propose to address these challenges with a novel *generate-then-ground* (GenGround) method, as shown in Figure 1. This approach diverges from the retrieve-then-read paradigm by first allowing the LLMs to generate an immediate answer, and then grounding this answer in evidence to revise it. In this work, we focus on the following research questions: (1) **RQ1**: How does our method synergize the world knowledge of LLMs and retrieved documents to answer a multi-hop question? (2) **RQ2**: For LLMs with different scales of parameters, how do we generalize our method?

To address **RQ1**, we enable LLMs to alternate between *answer deduction* and *instructional knowledge grounding*. In the deduction phase, LLMs form sub-questions from the input question and context. The LLMs then produce an immediate answer for each sub-question. To prevent non-factual hallucination by LLMs (Zhang et al., 2023; Gao et al., 2023a), we guide LLMs to revise the answer in the grounding phase, using documents retrieved from an external corpus like Wikipedia. The LLMs ground the question-answer pair in evidence by citing relevant content and correcting errors. This revised answer is used for the next iteration's sub-question, continuing till the final hop. We also introduce a batch grounding strategy for efficient document use.

To address **RQ2**, we propose an instructional grounding distillation method. Despite LLMs like ChatGPT performing well with our method, smaller models may struggle with instruction-following in the grounding phase. Thus, we use 50k single-hop questions from the Natural Questions (NQ) dataset (Kwiatkowski et al., 2019), each with ground-truth and noise documents. We guide ChatGPT to generate and adjust an answer for each question, then distill ChatGPT's process into a student model using instruction tuning.

Extensive experiments on four commonly-used MHQA datasets demonstrate superior performance over strong baselines (*e.g.,* ReAct and DSPy), achieving the best performance overall. We also observe that our instructional grounding distillation empowers the smaller model with strong performance.

To sum up, our main contributions are as follows: (1) We propose a novel generate-then-ground framework for retrieval-argument generation technique in multi-hop question tasks, which effectively synergizes the knowledge of LLMs and retrieved documents. (2) We introduce an instructional grounding distillation method, enabling a smaller model with the generate-then-ground framework. (3) Experiments on four datasets are conducted to demonstrate the superiority of our method.

2 Related work

2.1 Multi-hop Question Answering

Multi-Hop Question Answering (MHQA) tasks focus on answering questions that require gathering information from multiple sources and conducting multi-step reasoning to arrive at a comprehensive answer (Zhang et al., 2024; Li and Du, 2023). Some works utilize the knowledge deduction capability (Wei et al., 2022) of LLMs to decompose the input question into single-hop questions and then solve them step-by-step (Wang et al., 2022a, 2023). And many techniques (Yao et al., 2023; Besta et al., 2023) are proposed to improve the reasoning ability of LLMs. However, the LLMs suffer from generating non-factual statements. As an intuitive solution, many recent works integrate retrieval into the chain of thought reasoning process (Yao et al., 2022; Schick et al., 2023), prompting the LLMs to generate the answer using retrieved documents. Although promising, the inevitable noise in retrieved documents could mislead the LLMs to a wrong reasoning direction and derive a wrong answer (Xu et al., 2024). In this work, we propose an instructional knowledge grounding method that enables LLMs to find the most relevant evidence from the document list, thereby reducing the effect of the noise.

2.2 Retrieval-Augmented Generation

Retrieval-augmented generation (RAG) has been proven a promising technique to improve the performance of LLMs in knowledge-intensive NLP tasks (Zhu et al., 2023; Yu et al., 2023), which

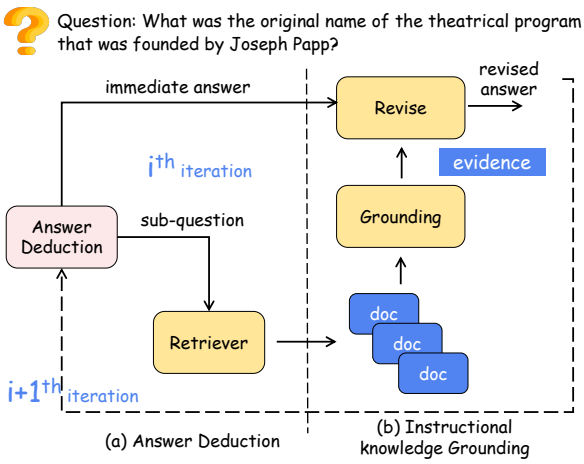

Figure 2: The architecture of the proposed generate-then-ground framework.

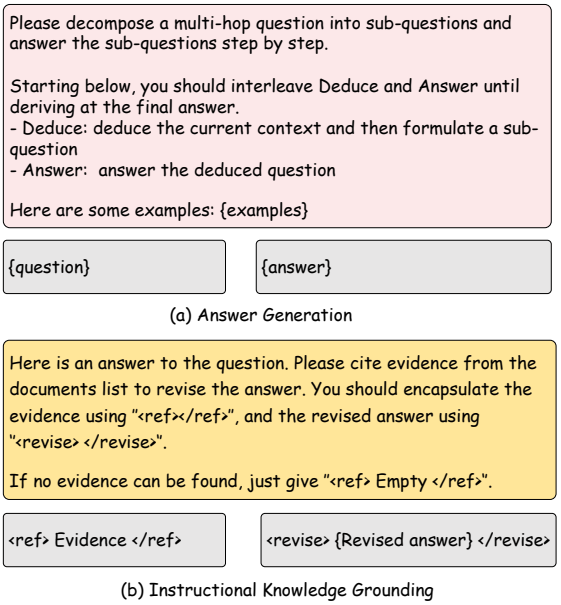

Figure 3: The instruction for the *answer deduction* (a) and *instructional knowledge grounding*(b) phases in our framework. The pink and yellow blacks indicate the input while the gray blocks indicate the output.

enhances LLMs with retrievers to access external knowledge. Existing RAG methods typically follow a retrieval-then-read pipeline (Ma et al., 2023; Feng et al., 2023; Gao et al., 2023b). Given a query, a retriever is first employed to retrieve the relevant document and a reader is then used to predict the answer on the condition of retrieved documents (Khattab et al., 2023). Despite the advancement of this paradigm, it is limited by the accuracy of retrievers (Yu et al., 2022; Zhang et al., 2024).

Recent works mitigate this problem by incorporating the world knowledge of LLMs (Gao et al., 2023a; Chen et al., 2023), where they utilize the LLM as a knowledge base to generate contextual documents and then read the answer from both retrieved and generated documents (Abdallah and Jatowt, 2023). However, the potential knowledge conflict between the two knowledge sources is ignored, which may hallucinate the LLMs (Xie et al., 2023; Thakur et al., 2023a; Mallen et al., 2023). In our work, we propose a generate-then-ground framework, more effectively incorporating the parametric knowledge of LLMs and external documents.

3 Generate-then-Ground with LLMs

This section provides a detailed explanation of GenGround. As depicted in Figure 2, GenGround empowers the LLMs to alternate between two phases over multiple iterations until the final answer a, is derived. These phases include *answer deduction* (Section 3.1) and *instructional knowledge grounding* (Section 3.2). During each iteration, the former guides the LLMs to generate a simpler, single-hop answer. The latter phase addresses LLMs' non-factual hallucination (Zhang et al., 2023) by prompting them to ground the question-answer pair in evidence and correct wrong predictions. The revised answer and sub-question are then integrated into the LLMs' context for the following iteration's prediction.

3.1 Answer Deduction

The answer deduction phase aims to utilize the world knowledge of LLMs stored within their parameters θ. Given the complex reasoning involved in multi-hop questions (Tang et al., 2021; Li and Du, 2023), it can be challenging to generate an accurate answer directly. As a result, we guide the LLM, denoted as $\mathcal{M}_\theta$, to break down a complex question $\mathcal{Q}$ into single-hop questions with a fine granularity. Formally, for i-th iteration, we denote the current context as $\mathcal{H} = \{(q_j, \tilde{j}_i) | j < i\}$, which comprises the accumulation of previous deduced sub-questions, $q_{<i}$, and revised answers, $\tilde{a}_{<i}$. The context $\mathcal{H}$, along with the input question $\mathcal{Q}$, is then fed into the LLMs to generate a sub-question, q_i, that defines the specific information to be retrieved. Subsequently, we prompt the LLMs, $\mathcal{M}_\theta$, to directly generate an answer, a_i, for the formulated question, q_i. This can be formulated as follows:

$$q_i, a_i = \mathcal{M}_\theta(\mathcal{I}_\mathcal{A}, \mathcal{Q}, \mathcal{H}_i). \quad (1)$$

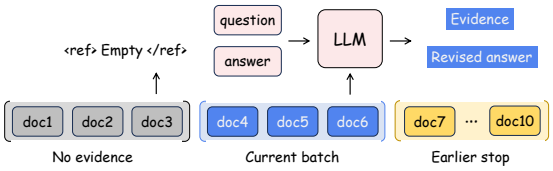

Figure 4: Demonstration of our batch grounding strategy with the batch size of 3 and retrieved documents amount of 10, where the LLMs ground the input question-answer pair into the second batch.

Here, the $\mathcal{I}_{\mathcal{A}}$ represents the instruction shown in Figure 3, which includes the task demonstration and in-context learning examples.

3.2 Instructional Knowledge Grounding

Given that LLMs may generate non-factual statements or "hallucinations", we further guide the LLMs to revise the generated answer, a_i, with the support of retrieved documents. Specifically, in the i-th iteration, we initially utilize a retriever (*e.g.,* Google, BM25, or dense retrieval model) to retrieve relevant documents, $\mathcal{D}_i$, using the deduced sub-question q_i (Equation 1):

$$\mathcal{D} = \text{Retrieval}(q_i). \tag{2}$$

We then guide the LLMs, $\mathcal{M}_\theta$, to ground the question-answer pair, (q_i, a_i), in established evidence by citing the most relevant content, $\tilde{d}$, from the retrieved documents, $\mathcal{D}_i$. Subsequently, we revise the answer a_i:

$$\tilde{a}_i = \mathcal{M}_\theta(\mathcal{I}_{\mathcal{G}}, Q, q_i, a_i). \tag{3}$$

As depicted in Figure 3(b), $\mathcal{I}_{\mathcal{G}}$ represents our grounding instruction in a zero-shot setting, while $\tilde{a}_i$ is the revision trajectory, which includes the cited evidence and the revised answer. The revision trajectory $\tilde{a}_i$, along with the question q_i, are then combined to build the context, $\mathcal{H}_{i+1} = \mathcal{H}_i \cup \{(q_i, \tilde{a}_i)\}$, for the LLMs in the subsequent iteration. If no relevant content can be cited (*e.g.,* the citation is `Empty`), we keep the generated answer as the revised answer without any changes.

3.3 Batch Knowledge Ground

Since retrieved documents are typically lengthy and contain inevitable noise (Xu et al., 2024), the LLMs are susceptible to being misled by plausible-looking statements during the grounding phase (Sun et al., 2023a; Thakur et al., 2023a). Therefore, we propose a simple yet efficient batch grounding strategy. Suppose the batch size is b. We first utilize the LLMs to revise the generated answer a_i using the $(1, b)$-th documents. If relevant evidence can be cited to revise the answer, we end our grounding phase for the current iteration and move to the next iteration. Otherwise, we prompt the LLMs to generate an "Empty" signal and then access the $(b+1, 2b)$-th documents sequentially. This process continues until the relevant evidence can be found to support our grounding phase. Figure 4 shows a concrete example with ten retrieved documents. If no relevant document can be found, we directly output the generated answer as a backup.

4 Generalization with Grounding Distillation

While LLMs like ChatGPT are skilled and adept at following instructions, they are often considered black boxes (Qin et al., 2023; Gao et al., 2024) and their extensive parameters can increase latency and inference cost in real-world applications (Sun et al., 2023b). Thus, we aim to adapt our framework to smaller, open source models with fewer parameters. Initial experiments show these smaller models struggle to cite relevant evidence during the knowledge grounding phase. To overcome this, we introduce **I**nstructional **G**rounding **Di**stillation (IDG), which distills the output trajectory of ChatGPT into a smaller student model.

4.1 Synthesize the Training Dataset

The instructional grounding distillation collects the trajectory of LLMs, *i.e.,* ChatGPT, during the instructional knowledge grounding (Section 3.2). This trajectory is then used as the training dataset to distill the grounding capability into a student model. To achieve this, we first sample 50k questions from the Natural Questions (NQ) dataset (Kwiatkowski et al., 2019). Each question q is paired with a corresponding ground-truth document $\tilde{d}$ and the noise documents $\mathcal{D}$. The questions in the NQ dataset are of high quality and single-hop, making them inherently similar to the setting of our instruction knowledge grounding (Section 3.2). Next, we supplement each question with an immediate answer a and a detailed revision trajectory $\tilde{a}$. Specifically, the immediate answer a is generated directly by feeding the question q into a smaller model, such as Mistral-7B (Jiang et al., 2023a). The revision trajectory $\tilde{a}$ is generated by ChatGPT with the assistance of the ground truth

Statistic	
# The data scale	45,710
# The average length of input instruction	70.87
# The average length of output	683.21
# The average number of ground truth documents	1.00
# The average length of ground truth documents	117.57

Table 1: The statistics of our synthetic dataset in the instructional grounding distillation method.

document. Various heuristic methods are also used to filter low-quality output (see Appendix III for more details). The statistics of out synthetic dataset is provided in Table 1.

4.2 Training Objective

Formally, for each question q in our synthetic dataset, we train the model to cite the relevant content from a document list and revise any incorrect predictions in the immediate answer a following the instruction $\mathcal{I}_{\mathcal{G}}$. Using the collected revision trajectory $\tilde{a}$, we apply the standard language modeling loss to optimize the student model:

$$\begin{aligned}\mathcal{L}_{\mathcal{G}} &= -\log P_\theta(\tilde{a}|\mathcal{I}_{\mathcal{G}}) \\ &= -\sum_{t=1}^{|\tilde{a}|} \log P_\theta(\tilde{a}_t|\tilde{a}_{(<t)}, \mathcal{I}_{\mathcal{G}}, \{\tilde{d}\} \cup \mathcal{D}).\end{aligned} \tag{4}$$

Here, the $\tilde{d}$ indicates the ground-truth document, and $\mathcal{D}$ indicates the noise documents.

5 Experimental Setup

5.1 Datasets

In line with previous research (Lewis et al., 2020; Xu et al., 2024), we conduct experiments on four common-used MHQA benchmarks, namely HotpotQA (HQA) (Yang et al., 2018), MuSiQue (MQA) (Trivedi et al., 2022), 2Wikimultihopqa (WQA) (Ho et al., 2020), and StrategyQA (SQA) (Geva et al., 2021). StrategyQA is derived from BIG-bench[1]. For the remaining benchmarks, we randomly sample 1.4k questions adhering to RetGen (Shao et al., 2023) and Verify-and-Edit (Zhao et al., 2023).

5.2 Baselines

We compare our method with *Generation w/ Retrieval* and *Generation w/o Retrieval* methods, respectively. The *w/* indicates *with* while the *w/* indicates *without*.

The *Generation w/o Retrieval* methods utilize the parametric knowledge of LLMs to answer questions. This includes (1) *CoT* (Wei et al., 2022), which prompts the LLMs to a series of intermediate reasoning steps when answering a question; (2) *CoT-SC* (Wang et al., 2022b), which samples a diverse set of reasoning paths and then selects the most consistent answer; and (3) *GenRead* (Yu et al., 2022), which generates the answer by reading from the documents generated by LLMs. The *Generation w/ Retrieval* methods augment LLMs with retrievers to access external knowledge when answering questions, including: (1) ReAct (Yao et al., 2022), which interleaves question generation, document retrieval, and knowledge incorporation to answer a question; (2) GRG (Abdallah and Jatowt, 2023), reading an answer from both retrieved documents and contextual documents generated by LLMs; (3) RetGen (Shao et al., 2023), which enhances LLMs with an iterative retrieval-generation synergy strategy to answer a multi-hop question. (4) DSP (Khattab et al., 2022), a programming framework empowered by LLMs; and (5) SearChain (Xu et al., 2024), which dynamically interacts with the retriever to verify and correct the generated answers.

Considering the complexity of multi-hop questions, we enhance GenRead and GRG with the chain-of-thought technique (*w/ decomposition*), dividing the input question into sub-questions and using GenRead (or GRG) for each sub-question.

5.3 Evaluation Metrics

Following previous studies (Xu et al., 2024; Ren et al., 2023), we use *Accuracy* (Acc) and *F1* metrics for evaluation. The *accuracy* metric checks if the ground truth answer is in the generated answer, which is also named cover-EM. The *F1* score is used to measure the overlap between the generated answer and the ground truth answer, which represents the harmonic mean of precision and recall. Recall is determined by considering the number of overlaps with the correct answer tokens, while precision is determined by considering the number of overlaps with all generated tokens. We also assess semantic accuracy (Acc†) using gpt-3.5-turbo-instruct[2] for a more thorough evaluation, which prompts the

[1] https://github.com/google/BIG-bench

[2] https://openai.com/

Methods	HotpotQA			MuSiQue			2Wikimultihopqa			StrategyQA
	F1	Acc	Acc†	F1	Acc	Acc†	F1	Acc	Acc†	Acc
Generate w/o Retrieval										
CoT (Wei et al., 2022)	35.28	30.79	37.07	23.35	13.21	17.85	35.41	32.46	34.52	67.83
CoT-SC (Wang et al., 2022b)	42.25	38.68	39.07	15.61	10.02	12.42	40.37	36.57	38.59	70.84
GenRead (Yu et al., 2022)	35.21	36.81	37.54	9.77	9.29	10.32	23.13	20.62	28.31	67.13
GenRead *w/ decomposition*	42.28	43.32	45.31	20.13	17.58	20.62	41.19	41.63	43.24	68.13
Generate w/ Retrieval										
VE (Zhao et al., 2023)	29.64	22.64	24.64	6.5	11.14	15.57	13.76	31.57	32.64	63.07
ReAct (Yao et al., 2022)	40.70	33.10	37.12	15.34	17.32	19.32	35.50	30.10	33.41	68.37
GRG *w/ decomposition*	50.21	45.18	50.80	24.87	17.91	22.33	40.42	40.48	43.05	75.21
RetGen (Shao et al., 2023)	28.30	41.04	44.10	21.04	17.69	20.19	36.00	42.17	45.21	73.42
SearChain (Xu et al., 2024)	-	46.76	48.12	-	17.07	20.45	-	42.14	46.27	76.95
DSPy (Khattab et al., 2023)	47.80	42.43	50.07	20.11	13.40	17.40	44.77	43.43	45.43	71.78
GenGround (Ours)	**52.26**	**47.27**	**55.73**	**27.36**	**20.24**	**24.77**	**50.21**	**45.61**	**48.58**	**77.12**

Table 2: Evaluation results on multi-hop question answering datasets. Acc† indicates the semantic accuracy of model outputs evaluated with `gpt-3.5-turbo-instruct` with the same prompt. Since the *SearChain* prompts the LLM to generate a long-form answer while the ground truth answer in our dataset is short-form, we only evaluate it with the Acc and Acc† metrics.

Method	HQA	MQA	WQA	Average Δ
Retrieval → ColBERTv2				
Ours (IDG)	42.08	14.37	32.69	-
Ours (Vanilla)	38.31	11.34	29.45	3.35↓
DSPy	36.41	7.42	28.31	5.67↓
GRG *w/ dp*	32.56	9.34	25.63	7.20↓
RetGen	26.52	10.12	24.13	9.45↓
SearChain	24.62	9.46	26.53	9.51↓

Table 3: Accuracy (Acc) on three datasets with Mistral-7B as backbone. The *w/ dq* indicates *decomposition*. The *Vanilla* and *IDG* indicate enable our framework by prompting and our grounding distillation, respectively.

Ablation	HQA			SQA
	F1	Acc	Acc†	Acc
w/o deduction	42.65 $\downarrow_{9}$	41.08$\downarrow_{6}$	43.14 $\downarrow_{12}$	66.51 $\downarrow_{10}$
w/o grounding	45.14$\downarrow_{7}$	41.35$\downarrow_{4}$	43.23$\downarrow_{5}$	72.34$\downarrow_{5}$
w/o batch	47.27$\downarrow_{5}$	45.03$\downarrow_{2}$	51.19$\downarrow_{4}$	71.72$\downarrow_{5}$

Table 4: Evaluation results of our ablation study on two MHQA benchmarks.

LLMs to evaluate the correctness of the generated answer taking the ground truth answer as reference. In this work, we implement the Acc† using gpt-3.5-turbo-instruct and the full prompt for our evaluation can be found in Appendix I.

To counter the potential bias of automatic metrics (Shi et al., 2023), we conduct a human evaluation, with three educated individuals assessing the *correctness* of 120 randomly sampled cases from four benchmarks on a three-scale rating.

5.4 Implementation Details

We utilize OpenAI's *gpt-3.5-turbo* as the backbone for our framework and all baselines, with the decoding temperature set to 0 for deterministic generation, and batch size set to 3 in our batch grounding strategy. The open source model, Mistral-7B (Jiang et al., 2023a), is also employed for comparison. We mainly use ColBERTv2 (Santhanam et al., 2021) as the retriever, retrieving top-10 documents for each question. Alternately, BM25 (Robertson et al., 2009), Google Search[3] serve as retrievers in our analysis experiment. Following previous work (Xu et al., 2024), we use Wikipedia 2017 for HotpotQA, and a large-scale passage collection built on Wikipedia 2018 for other open-domain QA benchmarks.

Our instructional grounding distillation trains Mistral-7B with 50k synthetic examples. We optimize the model using the deepspeed ZeRO strategy (Rasley et al., 2020) with the learning rate of $5e^{-5}$ and the weight decay coefficient of 0.01. The training of our model can be done within 18 hours with 3 NVIDIA A100-PCIE-80GB GPUs.

6 Experimental Results

6.1 Experimental Results

Overall performance. Table 2 presents the experiment results. Our framework surpasses

[3] https://serper.dev/

Method	HQA	MQA	WQA	Average Δ ↓
Retriever → BM25				
Ours	42.21	18.32	40.32	-
DSPy	40.86	15.32	30.85	5.27↓
GRG *w/ dq*	41.31	15.62	38.84	2.36↓
RetGen	39.12	8.41	35.83	6.50↓
SearChain	39.57	14.93	37.41	3.65↓
Retriever → Google Search				
Ours	48.95	21.54	46.87	-
DSPy	46.86	20.71	39.92	3.29↓
GRG *w/ dq*	42.57	18.41	43.21	4.39↓
RetGen	42.82	14.27	44.31	5.32↓
SearChain	44.35	19.76	44.39	2.95↓

Table 5: Accuracy (Acc) on three datasets using BM25 and Google Search as retrievers, respectively. The *w/ dq* is short for *without decomposition*.

others on all four datasets and all metrics. Specifically, on the HotpotQA dataset, our GenGround achieves Acc=47.27, F1=52.26, and Acc†=55.73, considerably improving over the *Generate w/ Retriever* baselines. It also significantly outperforms retrieval-then-read baselines like DSPy and SearChain, with a 4-6 point increase in accuracy metrics across all datasets. The similar improvement is observed in our human evaluation results (see Appendix IV for more detials). These results indicate that our method effectively utilizes world knowledge and LLMs' deductive abilities to answer questions.

Results with the smaller model. We further evaluate our method by swapping the backbone LLMs with the open source model, *i.e.,* Mistral-7B, and repeating the experiment under the same conditions. As shown in Table 3, we implement our methods in two ways with the Mistral-7B: (1) directly prompting (vanilla); (2) tuning with our proposed instructional grounding distillation (IGD). We obverse that directly prompting Mistral-7B with our method yields better performance compared with baselines. The instructional grounding distillation further improves overall performance significantly, *e.g.,* pushing the Acc to 42.08 in the HotpotQA dataset (9.84% relative improvement) and 14.37 in the MusiqueQA dataset (26.4% relative improvement).

Results with different retrievers. We further evaluate the performance of our framework by using different retrievers in various retrieval scenarios. As shown in Table 5, we replace ColBERTv2 with BM25 and Google Search, using ChatGPT as a backbone LLM in all instances. Our method demonstrates the best performance regardless of the retriever used, indicating its adaptability in both low recall (BM25) and high recall (Google Search) scenarios. This could be due to our answer deduction phase, which uses LLMs' parametric knowledge to supplement the retrieved knowledge. Moreover, our instructional knowledge grounding phase effectively incorporates the retrieved document by citing the most relevant evidence, mitigating the negative impact of noisy documents.

6.2 Ablation Study

We employ the following modifications and repeat the experiment in the same setting as Table 2.

w/o deduction. We remove the *answer deduction* phase mentioned in Section 3.1, prompting the LLMs to directly generate an answer for a multi-hop question and revise it. As illustrated in Table 4, we observe 6 and 10 absolute decreases in HQA and SQA datasets in terms of the Acc metric, respectively. These results demonstrate the necessity of deducing the intricate knowledge in our answer deduction phase.

w/o grounding. We replace the instructional knowledge grounding phase mentioned in Section 3.2) with directly generating an answer using retrieved documents As shown in Table 4, the F1 and Acc metrics have a significant decline. A potential reason is that the LLMs may hallucinate the plausible-looking statement in the retrieved documents. Our instructional knowledge grounding method further instructs the LLMs to find the most relevant evidence.

w/o batch. We remove the *batch grounding* strategy in Section 2. As shown in Table 4, the F1 decreases from 52.26 to 47.27 and the Acc decreases from 47.27 to 45.03. These comparisons indicate that the LLMs struggle to generate correct answers when the reference document list is lengthy with irrelevant information.

6.3 Case Study

We conduct several case studies and find that GenGround is more effective at generating high quality answers to a question. Details can be found in Appendix II.

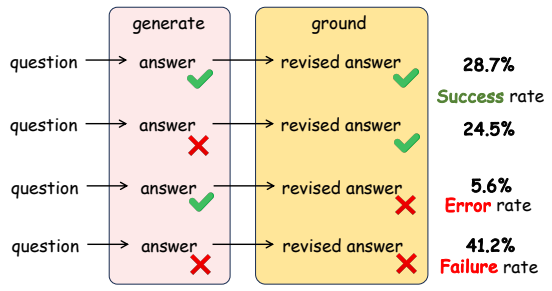

Figure 5: The fine-granularity *correctness* analysis of our answer deduction and knowledge grounding phases.

7 Analysis and Discussion

7.1 Result Consistency and Stability

We further explore the consistency and stability of our framework. Specifically, we repeat the experiment with the same setting as Table 2 in the HotpotQA dataset. The statistical significance of differences observed between the performance of two runs is tested using a two-tailed paired t-test. We find no significant difference between the results of two randomly conducted experiments (significance level $\alpha = 0.05$). We further explore the *fine-granularity stability* for our *answer deduction* and *instruction knowledge grounding* phases. Specifically, we compute the Rough score for the trajectory of our method in two repeated runs. The Rough-1, 2, and L are 81.33, 53.7, and 79.7, which shows the high lexicon similarity for the output, indicating the stability of our method.

7.2 Knowledge Incorporation

Our method combines the knowledge in LLMs' parameters and external documents to answer a question. We explore the synergistic integration of these two distinct knowledge sources. Specifically, we calculate the following three metrics: (1) *Success rate*: the rate at which LLMs either directly generate a correct answer or accurately revise an incorrectly generated answer; (2) *Failure rate*: LLMs generate a wrong answer and fail to correct it; (3) *Error rate*: LLMs generate a correct answer but incorrectly revise it.

Our method addresses multi-hop questions step-by-step. As existing datasets lack the ground truth answer for immediate answering trajectory, we invite three annotators to evaluate 100 randomly sampled cases from the Hotpot QA dataset in Table 2.

As illustrated in Figure 5, the overall *success rate* is 53.2%, with LLMs directly answering 28.7% of questions correctly. For 24.5% of the questions though, LLMs initially generate non-factual statements, and then use external documents for revision. These results underscore the importance of incorporating both knowledge sources. During our grounding phase, LLMs may be misled by plausible-looking statements in the retrieved documents. Therefore, we further calculate the **error rate**, which assesses how often LLMs are incorrect after revisions. We find that the error rate is only 5.6%, indicating that LLMs usually use the retrieved documents effectively.

7.3 Qualitative Analysis for Efficiency

The intensive parameters of LLMs typically raise concern about inference cost. Thus, we compare **token consumption** with *GRG w/ decomposition* and *RetGen*, using the HotpotQA dataset in Table 2. We show the frequency histogram for the number of consumed tokens in different methods in Figure 6. Though our framework achieves better performance, we observe that our method spends fewer tokens compared with strong baselines *RetGen* and *GRG w/ decomposition*. The potential reason is that our framework benefits from the deduction capability of LLMs to decompose a multi-hop question into simpler sub-questions and generate an answer directly, leading to a shorter reasoning trajectory.

We also train Mistral-7B using different amounts of randomly sampled examples to investigate the impact of **data scale** on the effectiveness of our instructional grounding distillation (IGD). We notice a slight decrease in performance as the amount of data reduces. For instance, when training with 45k examples, our method achieves Acc=42.08; but with 20k examples, it achieves Acc=40.12. All results are averaged over three runs. We also observe that our IGD allows Mistral-7B to achieve performance comparable to, and sometimes even better than, strong baselines using ChatGPT as the backbone, such as ReAct.

8 Conclusion

We present a generate-then-ground (GenGround) framework for multi-hop question answering tasks, synergizing the parametric knowledge of LLMs and external documents to solve a multi-hop question. Given a multi-hop question, our GenGround enable LLMs to alternate two phases until the final answer is derived: (1) formulate a

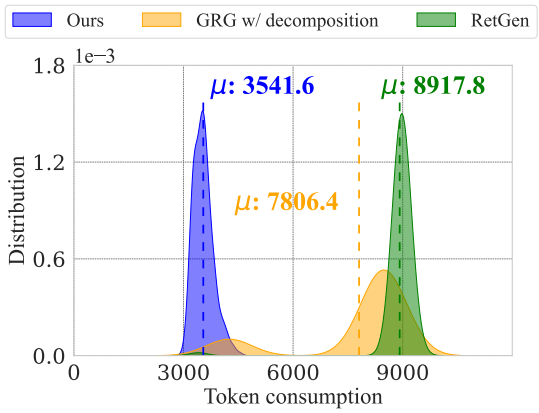

Figure 6: The efficiency analysis for different methods, where we count the number of consumed tokens and compute the average consumption μ.

simpler, single-hop question and directly generate the answer; (2) ground the question-answer pair in retrieved documents, amending any wrong predictions in the answer. To generalize our framework into smaller models, we also propose an instructional grounding distillation method. Extensive experiments conducted on four datasets illustrate the superiority of our framework.

Limitations

Despite the promising results demonstrated in this paper, there are several limitations to our framework.

- The first step of our framework involves generating an initial answer, which may be highly dependent on the task at hand. For different tasks, the model may struggle to generate a meaningful or useful initial answer, limiting its applicability across diverse problem domains.
- Our approach assumes that complex questions can be broken down into simpler ones. However, the task of decomposing complex questions is itself a challenging problem and has not been fully explored in our current framework.
- Our approach assumes that external documents can be used to correct initially non-factual statements generated by the model. However, if these sources do not contain the necessary information for correction, or if they contain misinformation, our framework's effectiveness could be compromised.

Ethics Statement

The research conducted in this paper centers around the development of a generate-then-ground framework for multi-hop question answering. Our framework enables Language Learning Models (LLMs) to alternately deduce answers and utilize established evidence to revise those answers across several iterations to solve multi-hop questions.

In the process of conducting this research, we have adhered to ethical standards to ensure the integrity and validity of our work. All the questions used in this study were obtained from existing benchmarks, thus ensuring a high level of transparency and reproducibility in our experimental procedure.

Furthermore, to support our retrieval system, we used an open source corpus, specifically, Wikipedia. This ensures that our research utilizes publicly accessible and freely available data, minimizing potential bias and promoting fairness.

We have made every effort to ensure that our research does not harm individuals or groups, nor does it involve any form of deception or potential misuse of information.

References

Abdelrahman Abdallah and Adam Jatowt. 2023. Generator-retriever-generator: A novel approach to open-domain question answering. *arXiv preprint arXiv:2307.11278*.

Vaibhav Adlakha, Parishad BehnamGhader, Xing Han Lu, Nicholas Meade, and Siva Reddy. 2024. Evaluating correctness and faithfulness of instruction-following models for question answering. In *Transactions of the Association for Computational Linguistics: TACL*.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Michal Podstawski, Hubert Niewiadomski, Piotr Nyczyk, et al. 2023. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence: AAAI*.

Liang Chen, Yang Deng, Yatao Bian, Zeyu Qin, Bingzhe Wu, Tat-Seng Chua, and Kam-Fai Wong. 2023. Beyond factuality: A comprehensive evaluation of large language models as knowledge generators. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: EMNLP*.

Zhangyin Feng, Xiaocheng Feng, Dezhi Zhao, Maojin Yang, and Bing Qin. 2023. Retrieval-generation synergy augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: EMNLP*.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, et al. 2023a. Rarr: Researching and revising what language models say, using language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics: ACL*.

Shen Gao, Zhengliang Shi, Minghang Zhu, Bowen Fang, Xin Xin, Pengjie Ren, Zhumin Chen, and Jun Ma. 2024. Confucius: Iterative tool learning from introspection feedback by easy-to-difficult curriculum. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023b. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did Aristotle Use a Laptop? A Question Answering Benchmark with Implicit Reasoning Strategies. *Transactions of the Association for Computational Linguistics (TACL)*.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics.

Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L'elio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023a. Mistral 7b. *ArXiv*, abs/2310.06825.

Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023b. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: EMNLP*.

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*.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. 2023. Dspy: Compiling declarative language model calls into self-improving pipelines. *arXiv preprint arXiv:2310.03714*.

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

Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Kuttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*.

Ruosen Li and Xinya Du. 2023. Leveraging structured information for explainable multi-hop question answering and reasoning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: EMNLP*.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting for retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: EMNLP*.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics: ACL*, pages 9802–9822.

Vaibhav Mavi, Anubhav Jangra, and Adam Jatowt. 2022. A survey on multi-hop question answering and generation. *arXiv preprint arXiv:2204.09140*.

Yujia Qin, Zihan Cai, Dian Jin, Lan Yan, Shihao Liang, Kunlun Zhu, Yankai Lin, Xu Han, Ning Ding, Huadong Wang, et al. 2023. Webcpm: Interactive web search for chinese long-form question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics: ACL*.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *SIGKDD*, page 3505–3506.

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

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Keshav Santhanam, O. Khattab, Jon Saad-Falcon, Christopher Potts, and Matei A. Zaharia. 2021. Colbertv2: Effective and efficient retrieval via lightweight late interaction. In *North American Chapter of the Association for Computational Linguistics*.

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*.

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. In *Findings of the Association for Computational Linguistics: EMNLP 2023*.

Zhengliang Shi, Weiwei Sun, Shuo Zhang, Zhen Zhang, Pengjie Ren, and Zhaochun Ren. 2023. Rade: Reference-assisted dialogue evaluation for open-domain dialogue. *ArXiv*, abs/2309.08156.

Weiwei Sun, Zhengliang Shi, Shen Gao, Pengjie Ren, Maarten de Rijke, and Zhaochun Ren. 2023a. Contrastive learning reduces hallucination in conversations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 13618–13626.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Pengjie Ren, Dawei Yin, and Zhaochun Ren. 2023b. Is ChatGPT Good at Search? Investigating Large Language Models as Re-Ranking Agent. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

Yixuan Tang, Hwee Tou Ng, and Anthony Tung. 2021. Do multi-hop question answering systems know how to answer the single-hop sub-questions? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: ACL*, pages 3244–3249. Association for Computational Linguistics.

Nandan Thakur, Luiz Bonifacio, Xinyu Zhang, Odunayo Ogundepo, Ehsan Kamalloo, David Alfonso-Hermelo, Xiaoguang Li, Qun Liu, Boxing Chen, Mehdi Rezagholizadeh, et al. 2023a. Nomiracl: Knowing when you don't know for robust multilingual retrieval-augmented generation. *arXiv preprint arXiv:2312.11361*.

Nandan Thakur, Luiz Bonifacio, Xinyu Crystina Zhang, Odunayo Ogundepo, Ehsan Kamalloo, David Alfonso-Hermelo, Xiaoguang Li, Qun Liu, Boxing Chen, Mehdi Rezagholizadeh, and Jimmy J. Lin. 2023b. Nomiracl: Knowing when you don't know for robust multilingual retrieval-augmented generation. *ArXiv*, abs/2312.11361.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. MuSiQue: Multihop questions via single-hop question composition. In *Transactions of the Association for Computational Linguistics: TACL*.

Liang Wang, Nan Yang, and Furu Wei. 2023. Query2doc: Query expansion with large language models. corr abs/2303.07678 (2023). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022a. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

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

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

Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. 2023. Adaptive chameleon or stubborn sloth: Unraveling the behavior of large language

models in knowledge conflicts. *arXiv preprint arXiv:2305.13300*.

Shicheng Xu, Liang Pang, Huawei Shen, Xueqi Cheng, and Tat-Seng Chua. 2024. Search-in-the-chain: Towards accurate, credible and traceable large language models for knowledge-intensive tasks. In *WWW*.

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

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601*.

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

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2022. Generate rather than retrieve: Large language models are strong context generators. *arXiv preprint arXiv:2209.10063*.

Wenhao Yu, Hongming Zhang, Xiaoman Pan, Kaixin Ma, Hongwei Wang, and Dong Yu. 2023. Chain-of-note: Enhancing robustness in retrieval-augmented language models. *arXiv preprint arXiv:2311.09210*.

Jiahao Zhang, Haiyang Zhang, Dongmei Zhang, Yong Liu, and Shen Huang. 2024. End-to-end beam retrieval for multi-hop question answering. In *2024 Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. 2023. Siren's song in the ai ocean: a survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*.

Ruochen Zhao, Xingxuan Li, Shafiq Joty, Chengwei Qin, and Lidong Bing. 2023. Verify-and-edit: A knowledge-enhanced chain-of-thought framework. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Toronto, Canada. Association for Computational Linguistics.

Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Zhicheng Dou, and Ji-Rong Wen. 2023. Large language models for information retrieval: A survey. *arXiv preprint arXiv:2308.07107*.

Appendices

Appendix I. Evaluation Metrics Details

Following prior research (Xu et al., 2024; Shao et al., 2023; Ren et al., 2023), we evaluate our method and baselines with three metrics: (1) *Accuracy* (Acc), which evaluates whether the ground truth answer is contained within the generated answer, (2) *F1 Score* (F1), which computes the lexical similarity between the generated answer and the ground truth answer. As the harmonic mean of precision and recall, it is calculated using term-level exact match other than ROUGE-L, with precision being the ratio of shared terms to predicted terms, and recall being the ratio of shared terms to actual terms, and (3) *Semantic Accuracy* (Acc†), which prompts the LLMs to evaluate the correctness of the generated answer taking the ground truth answer as reference. In this work, we implement the Acc† using gpt-3.5-turbo-instruct[4]. The prompt is as follows, where `{question}`, `{model output}`, and `{answer}` are placeholders. The results are averaged over three runs.

Prompt for Evaluating the Correctness of a Model Output

In the following task, you are given a Question, a model Prediction for the Question, and a Ground-truth Answer to the Question. You should decide whether the model Prediction implies the Ground-truth Answer.

Question
{question}

Prediction
{model output}

Ground-truth Answer
{answer}

Does the Prediction imply the Ground-truth Answer? Output Yes or No:

Appendix II. Case Study

We conduct several case studies and find that our method is more effective at generating high-quality answers to a question. A concrete example is shown in Table 7. We find that our GenGround can derive the correct answer successfully in two hops while the other baselines fail. In the first hop, the GenGround enables the LLMs to formulate a simpler, single-hop question and directly generate a correct answer, which intuitively demonstrates the world knowledge of LLMs. We also observe that although a wrong prediction is generated in the second hop initially, our GenGround can instruct LLMs to revise it with the assistance of an external document. This phenomena further illustrates the necessity to incorporate both the parametric knowledge of LLMs and external documents to answer a complex, multi-hop question.

Appendix III. Training Dataset for Grounding Distillation

Synthesize the dataset Our instructional grounding distillation collects the trajectory of LLMs, *i.e.,* ChatGPT, during the instructional knowledge grounding (Section 3.2). To achieve this, we first sample 50k questions from the Natural Questions (NQ) dataset (Kwiatkowski et al., 2019). Each question is paired with a corresponding ground-truth document and the noise documents. Next, we supplement each question with an immediate answer and a detailed revision trajectory. The immediate answer is generated directly by feeding the question into a smaller model, such as Mistral-7B. The revision trajectory is generated by ChatGPT with the assistance of the ground truth document.

For practical consideration, we prompt the LLMs to encapsulate the cited evidence and revised answer with special tokens, *i.e.,* "<ref> </ref>" and "<revise> </revise>" in the output. We employ the following heuristic methods to filter the low-quality generated output:

- The output contain no evidence, *e.g.,* "<ref> Empty </ref>".
- The output contain no revised answer, *e.g.,* no "<revise>" or "</revise>" can be found.
- The revised answer in the output is misaligned with the ground truth evidence.

We demonstrate diversity in the length of the input instruction, and output trajectory in Table 1.

[4] https://openai.com/

	Ours	GRG	DSPy	SearChain
Correctness	52.75	49.71	51.24	46.30

Table 6: The result of our human evaluation.

Appendix IV. Human Evaluation

we conduct a human evaluation, with three educated individuals assessing the *correctness* of 120 randomly sampled cases from four benchmarks on a three-scale rating. Specifically, the annotators are invited to manually evaluate the question-answering trajectory step-by-step using the ground truth document provided by existing datasets. To guarantee annotation quality, we ask at least two annotators to evaluate the same questions repeatedly. If there is a discrepancy between the two annotators (*i.e.,* two annotators give a different correctness score), we ask a third annotator to recheck it. As the results are shown in Table 6, our method achieves the best performance with a 3.66 point improvement on average. This result demonstrates the superiority of our method. The overall Kappa value of our human evaluation is 0.73, illustrating agreement among the annotators.

Query In what month is the annual documentary film festival, that is presented by the fortnightly published British journal of literary essays, held? (HotpotQA dataset)

GenGround

Question 1: What is the name of the annual documentary film festival presented by the fortnightly published British journal of literary essays?
Answer 1: The Fortnightly Review Documentary Film Festival.
Grounding 1: The document demonstrate <ref> The annual documentary film festival presented by the fortnightly published British journal of literary essays is called the London International Documentary Festival (LIDF) </ref>. <revise>the London International Documentary Festival (LIDF) </revise>.
Question 2: The annual documentary film festival presented by the fortnightly published British journal of literary essays is called the London International Documentary Festival (LIDF). In what month is LIDF held?
Answer 2: LIDF is held in the months of March and April* every year.
Grounding 2: The document demonstrate <ref> The London International Documentary Festival (or LIDF) is an annual documentary film festival that takes place in the months of March and April every year </ref>. The revised answer is <revise> LIDF is held in the months of March and April every year </revise>.
###Finish[March and April]

Chain of thought

Step 1: The annual documentary film festival is presented by the fortnightly published British journal of literary essays.
Step 2: The entity presenting the festival is the British journal of literary essays.
Step 3: "Fortnightly" means the journal is published every two weeks.
Step 4: Since the journal is published every two weeks, we need to find out which months fall around the time when the festival is typically held. If the journal is published fortnightly, it means there are approximately 26 issues per year. Given that the festival is annual, it should occur once within those 26 fortnightly periods.
Final Answer: This film festival, presented by the British journal of literary essays, is held in the month of May.

GRG w/ decomposition

Question 1: I need to get for Name of the British journal of literary essays
Document 1: [1] The London Review of Books (LRB) is a British journal of literary essays. It is published fortnightly... ||| [2] Essays in London and Elsewhere is a book of literary criticism by Henry James published in 1893...
Question 2: When is the annual documentary film festival held? Document 2: The annual documentary film festival, also known as DOC NYC, is held every November in New York City. It was founded in 2010 and has since become one of the largest documentary film festivals in the United States. The festival showcases over 300 films and hosts panels, workshops. ||| The London International Documentary Festival (or LIDF) is an annual documentary film festival that takes place in the months of March and April every year. The event...
###Finish[November]

SearChain

Question 1: [Unsolved Query]: What is the name of the annual documentary film festival presented by the fortnightly published British journal of literary essays? [Unsolved Query]: What is the schedule or dates for the annual documentary film festival presented by the fortnightly published British journal of literary essays? [Unsolved Query]: When does the annual documentary film festival presented by the fortnightly published British journal of literary essays usually take place? [Unsolved Query]: In what month is the annual documentary film festival usually held?
Document 1: [1] Essays in London and Elsewhere is a book of literary criticism by Henry James published in 1893...
Answer 1: The annual documentary film festival presented by the fortnightly published British journal of literary essays is called the London International Documentary Festival (LIDF). Question 2: [Query]: When does the London International Documentary Festival usually take place? Answer 5: The London International Documentary Festival usually takes place in the months of October every year.

Table 7: A concrete example on the HotpotQA dataset. We highlight the right and wrong prediction used to answer the input question. We find that our GenGround can derive the correct answer successfully in two hops while the other baselines fail. In the first hop, the GenGround enables the LLMs to formulate a simpler, single-hop question and directly generate a correct answer. We also observe that although a wrong prediction is generated in the second hop initially, our GenGround can instruct LLMs to revise it with the assistance of an external document.