EfficientRAG: Efficient Retriever for Multi-Hop Question Answering

Ziyuan Zhuang[1*], **Zhiyang Zhang**[1*], **Sitao Cheng**[1], **Fangkai Yang**[2‡], **Jia Liu**[1],
Shujian Huang[1], **Qingwei Lin**[2], **Saravan Rajmohan**[2], **Dongmei Zhang**[2], **Qi Zhang**[2]
[1] State Key Laboratory for Novel Software Technology, Nanjing University [2] Microsoft
ziyuan.zhuang@smail.nju.edu.cn

Abstract

Retrieval-augmented generation (RAG) methods encounter difficulties when addressing complex questions like multi-hop queries. While iterative retrieval methods improve performance by gathering additional information, current approaches often rely on multiple calls of large language models (LLMs). In this paper, we introduce EfficientRAG, an efficient retriever for multi-hop question answering. EfficientRAG iteratively generates new queries without the need for LLM calls at each iteration and filters out irrelevant information. Experimental results demonstrate that EfficientRAG surpasses existing RAG methods on three open-domain multi-hop question-answering datasets. The code is available in aka.ms/efficientrag.

1 Introduction

Large-language models (LLMs) have shown remarkable performance in numerous applications and tasks (OpenAI, 2023; Jiang et al., 2023a; Touvron et al., 2023b). However, LLMs lack knowledge underrepresented in their training data, especially in domain-specific settings, and still face the issues of hallucinations (Zhang et al., 2023; Huang et al., 2023; Yang et al., 2023). Retrieval-augmented generation (RAG) techniques (Lewis et al., 2020; Gao et al., 2023) have been widely adapted to retrieve knowledge from external resources to ground the generated responses. Previous RAG methods often adapt one-round retrieval, *e.g.*, only use the user query or question as the input to retrieve knowledge (Guu et al., 2020; Borgeaud et al., 2022; Izacard et al., 2023; Shi et al., 2023). Such one-round RAG is capable of answering questions which clearly state all the needed information in the input query (Thorne et al., 2018; Trischler et al., 2017; Rajpurkar et al., 2016), such as one-hop question, *e.g.*, *"what is Newton's third law of motion?"*. However, one-round RAG methods could fail in complex questions where more information is required beyond the first-round retrieved information, *e.g.*, multi-hop questions (Yang et al., 2018a; Trivedi et al., 2022a; Ho et al., 2020b). In order to deal with complex multi-hop questions, recent works propose to obtain required information through multi-round retrievals or reasonings, such as rewriting or generating queries for the following multi-round retrievals (Khattab et al., 2022; Ma et al., 2023; Shao et al., 2023; Jiang et al., 2023b), discriminating and correcting internal reasoning procedures (Gao et al., 2024), interleaving multiple retrieval and reasoning steps (Trivedi et al., 2023), multi-rounds of self-asking (Press et al., 2023). However, such iterative retrieval approaches have the following limitations: (1) they require multiple LLM calls concerning rewriting or generating new queries for the next round of retrieval, thus increasing the latency and cost. (2) they require dedicated prompting and few-shot examples that might need updating across different scenarios.

In this paper, we are inspired by the intuition that the types of relations in multi-hop questions are limited, or significantly fewer compared to the number of entities. As proved in Zhu et al. (2023) that small models have a certain ability of reasoning, we propose that identifying relations and their associated entities from retrieved information can be effectively managed by small models instead of LLMs. Thus, we propose EfficientRAG consists of a Labeler and a Filter to iteratively generate new queries for retrieval and in the meanwhile keep the most relevant retrieved information, enhancing efficiency compared to other RAG methods.

2 Empirical Study

2.1 Capability of LLM generator

In this section, we conducted an empirical study to assess how well an LLM-based generator per-

[*] Equal Contribution
[†] Work is done during an internship at Microsoft
[‡] Corresponding author

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 3392–3411
November 12-16, 2024 ©2024 Association for Computational Linguistics

forms with different levels of retrieved information. We test on three settings: direct prompt (no retrieved chunks), oracle chunks (oracle chunks as the context), and mixed chunks (both oracle and irrelevant chunks as the context) on three datasets, *i.e.*, HotpotQA (Yang et al., 2018b), 2Wiki-multihop (2WikiMQA) (Ho et al., 2020a) and MuSiQue (Trivedi et al., 2022b). The generator model includes GPT-3.5 (OpenAI, 2022), GPT-4 (OpenAI, 2023) with 1106-preview version, and Llama-3-8B[1] (Touvron et al., 2023a). We evaluate the model answer with accuracy metric by GPT-3.5, the prompt can be found in Appendix B.1. As illustrated in Figure 1, retrieval proves beneficial, with both oracle and mixture settings outperforming the direct answering approach. Nonetheless, the presence of irrelevant chunks continues to challenge the LLM generator, underscoring the need for more precise information retrieval.

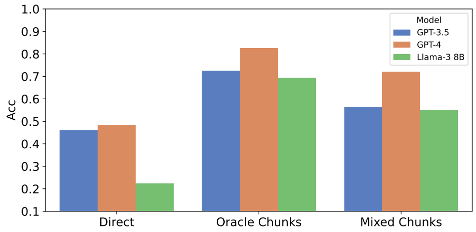


Figure 1: The performance with varying chunks settings over 2WikiMQA dataset with GPT-3.5/GPT-4/Llama3-8B as the generator.

2.2 Retrieve with Query Decomposition

It is a common practice to use LLMs for query decomposition when facing complex multi-hop questions (Gao et al., 2023). We conduct another empirical study to check how query decomposition approaches impact the retrieval stage. As shown in Figure 2, the number of oracle chunks retrieved by one-time decomposition (LLM Decompose, detailed in Table 11) outperforms the Direct retrieval for the original query. At a similar number of chunks, iterative decomposition (EfficientRAG Decompose) achieves higher recall. When retrieving approximately 20 chunks, the Recall achieved by EfficientRAG Decompose has comparable performance with the LLM Decompose when retrieving around 200 chunks, thus demonstrating the efficiency of EfficientRAG Decompose. All retrievers used the contriever-msmarco (Izacard et al., 2022) setup, with chunk retrievals configured as 1/3/5/10/20/30/50/100, and the LLM endpoint is gpt35-turbo-1106.

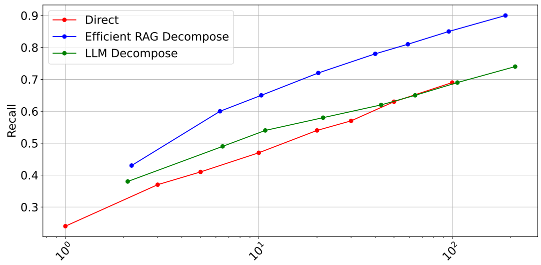


Figure 2: Recall of retrieve efficiency over three retrieval strategies on MuSiQue dataset. The x-axis is log-scaled. Each point on different lines represents the same number of retrieved chunks.

3 Methodology

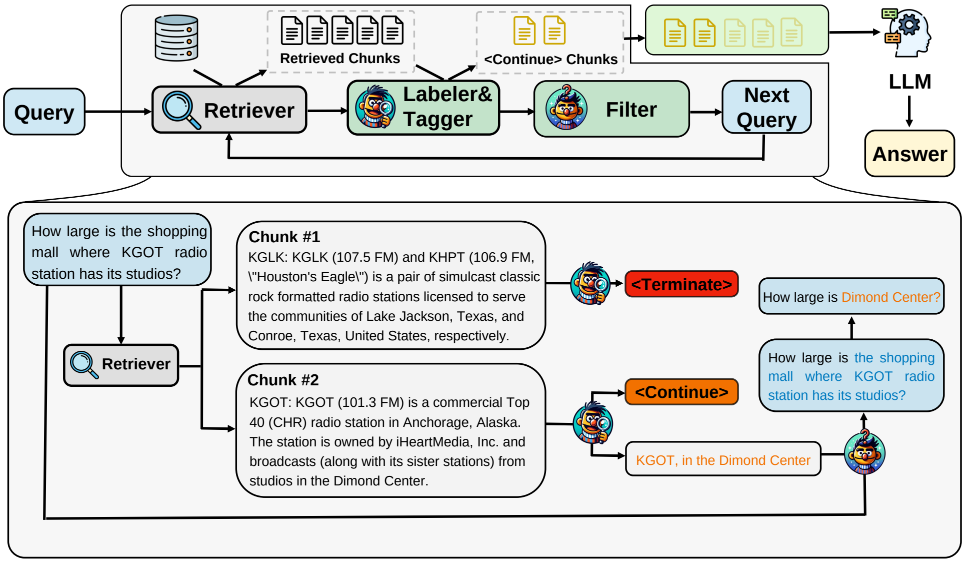
3.1 EfficientRAG Framework

In this section, we introduce EfficientRAG , a plug-and-play approach designed to efficiently retrieve relevant information with multiple retrieval rounds to enrich the retrieved information and reduce irrelevant information, then help improve the quality and accuracy of answers.

EfficientRAG consists of two lightweight components: the Labeler & Tagger and the Filter. These components share the same model structure, with the Labeler & Tagger[2] producing outputs from separate heads within the same model and the filter's output comes from another model. Both the Labeler and the Filter function as token-level classifiers, classifying tokens as either true or false. Figure 3 shows that how EfficientRAG fits into traditional RAG systems. Given a query, the retriever obtains relevant chunks from the database. Then the labeler module annotates a sequence of tokens in this document representing the useful information that could (partially) answer the query. The tagger module then tags the chunk, indicating whether the retrieved chunk is helpful or irrelevant. If the tag indicates there needs more information to answer the query, *i.e.*, tagged as <Continue>, we will add this chunk to a candidate pool, which will be fed to the LLM-based generator to have the final answer. Otherwise, if the document is labeled useless or irrelevant, we stop searching for the successor branches from this query. The filter module takes both the labeled tokens and the current query to construct a new query for the next round of retrieval. It is done by replacing the unknown part of

[1] https://llama.meta.com/llama3/

[2] We use the term Labeler as the representation.


Retrieved Chunks
<Continue> Chunks
Query
Retriever
Labeler&
Tagger
Filter
Next
Query
LLM
Answer
How large is the shopping mall where KGOT radio station has its studios?
Retriever
<Terminate>
<Continue>
How large is Dimond Center?
How large is the shopping mall where KGOT radio station has its studios?
KGOT, in the Dimond Center

Chunk #1
KGLK: KGLK (107.5 FM) and KHPT (106.9 FM, \"Houston's Eagle\") is a pair of simulcast classic rock formatted radio stations licensed to serve the communities of Lake Jackson, Texas, and Conroe, Texas, United States, respectively.

Chunk #2
KGOT: KGOT (101.3 FM) is a commercial Top 40 (CHR) radio station in Anchorage, Alaska. The station is owned by iHeartMedia, Inc. and broadcasts (along with its sister stations) from studios in the Dimond Center.

Figure 3: EfficientRAG framework operates within the iterative RAG system. Initially, EfficientRAG retrieves relevant chunks from the knowledge base, tagging each as either <Terminate> or <Continue>, and annotates preserved tokens *"KGOT in the Dimond Center"* from the <Continue> chunks. The Filter then processes the concatenation of the original question and the previously annotated tokens, *"Q: How large is the shopping mall where KGOT radio station has its studios? Info: KGOT, in the Dimond Center"*, and annotates the next-hop query tokens *"How large is Dimond Center?"*. This iterative process continues until all chunks are tagged <Terminate> or the maximum number of iterations is reached.

the query with labeled tokens (useful information).

Our approach efficiently generates new queries for subsequent retrieval rounds, aiming to retrieve information beyond the scope of the initial query. Once our approach gets enough information to answer the initial question, it stops and passes all this information to the final generator to get the final response. Leveraging our efficient RAG approach eliminates the need for multiple LLM calls for query generation, resulting in improved performance when tackling complex queries.

3.2 Synthetic Data Construction

We utilize LLM to synthesize training data for the Labeler and Filter. The process consists of multi-hop question decomposition, token labeling, next-hop question filtering, and negative sampling. Synthetic data is detailed in Table 1.

Multi-hop question decomposition. Given a multi-hop question and relevant chunks, we first prompt the LLM to decompose the original question into several single-hop questions. Each single-hop question corresponds to a chunk. Then, we ask the LLM to parse the dependency for the sub-questions.

Token Labeling. For each sub-question and corresponding chunk, we prompt the LLM to label important words in the chunk pertinent to the sub-question answering. We annotate each word in the chunk with a binary label to determine if it is important and should be preserved by EfficientRAG Labeler. We use the SpaCy toolkit[3] following Pan et al. (2024).

Next-hop question filtering. Given a single-hop question and the labeled tokens from its dependent questions, we prompt the LLM to generate a next-hop question, which is ideally the next query for retrieval. We extract the next-hop question tokens same as the Token Labeling procedure.

Negative Sampling. With each filtered next-hop question, we retrieve the most similar but not relevant chunk as the hard negative chunk. These negative chunks will be tagged <Terminate> while other relevant chunks are tagged <Continue>.

Dataset	HotpotQA	MuSiQue	2WikiMQA
Labeler	357k	93k	70k
Filter	73k	25k	13k

Table 1: Amount of synthesized training data for different datasets.

[3] https://spacy.io/

Method/Dataset	HotpotQA		MuSiQue		2WikiMQA	
	Recall@K	K	Recall@K	K	Recall@K	K
Direct-R@10	70.52	10.00	50.86	10.00	61.58	10.00
Direct-R@20	74.87	20.00	57.12	20.00	65.24	20.00
Direct-R@30	77.05	30.00	60.67	30.00	66.91	30.00
Decompose-R	74.38	21.31	**67.23**	19.74	71.02	21.24
Iter-RetGen iter2	81.69	14.44	57.74	14.82	73.80	14.95
Iter-RetGen iter3	**83.05**	16.42	58.26	17.19	74.29	17.32
SelfASK	73.42	35.27	60.43	32.36	**88.90**	33.68
EfficientRAG	81.84	**6.41**	49.51	**6.09**	84.08	**3.69**

Table 2: Results on retrieval performance. Baselines are implemented from the source code. Bold and underlined fonts denote the best and second-best results respectively. EfficientRAG demonstrates comparable recall while retrieving the fewest number of chunks.

Method/Dataset	HotpotQA			MuSiQue			2WikiMQA		
	EM	F1	Acc	EM	F1	Acc	EM	F1	Acc
Direct	22.87	26.94	25.79	5.59	8.76	5.51	27.33	31.11	28.67
CoT	27.99	34.05	30.53	10.16	13.85	9.21	29.25	35.14	31.71
Direct-R@10	38.24	44.55	44.56	13.39	18.14	17.12	26.18	31.88	32.70
Decompose-R	36.15	42.68	46.31	12.59	19.25	19.53	25.22	31.44	32.53
Iter-RetGen iter2	55.45	59.47	**59.29**	26.95	29.15	26.28	43.85	49.96	49.22
Iter-RetGen iter3	56.76	60.89	57.56	28.20	30.31	25.31	43.07	49.83	46.59
SelfASK	33.58	39.10	42.36	24.56	29.22	**26.97**	47.56	54.84	**55.16**
EfficientRAG	50.59	57.93	57.86	16.44	21.18	20.00	44.18	51.64	53.41

Table 3: Results on end-to-end question answering performance across three datasets. The highest accuracy (Acc) values are highlighted in bold, while the second-highest are underlined. EfficientRAG exhibits promising high accuracy, comparable to that of the LLM-based baselines.

3.3 Training

We train EfficientRAG Labeler for two tasks, token labeling and chunk filtering, as they both take in the same input. We use an auto-encoder language model as an encoder to derive embeddings for each token of concatenated sequence query, chunk. Subsequently, we use one fully connected layer to project the token embedding into a 2-dimensional space, indicating "useful token" and "useless token". Another fully connected layer is adapted to project the average pooling of the sequence embedding into a 2-dimensional space, representing the chunk tag <Continue> and <Terminate>. We train EfficientRAGFilter similarly, while its input sequence is the concatenation of query and labeled tokens. The Filter extracts words and concatenates them to formulate the next-hop query.

4 Experiments

4.1 End2end QA performance

We conduct evaluations of our EfficientRAG and multiple baselines on three multi-hop question-answering datasets same as §2.1. We select the following models as our baselines. First is direct answering without retrieval, including LMs with proprietary data. We include direct prompting and Chain-of-Thought prompting (Touvron et al., 2023a) and question decomposition prompting in this setting. Secondly, we include baselines with naive RAG with top-10 retrieve chunks as its knowledge. Third, we include advanced iterative RAG methods like Iter-RetGen (Shao et al., 2023) and SelfAsk (Press et al., 2023). The implementation prompts are in Appendix B.3.

Implementation Details. EfficientRAG Labeler and Filter are fine-tuned based on DeBERTa-v3-large (He et al., 2021) with 24 layers and 304M parameters. We adopt Llama-3-8B-Instruct for the question-answering stage and all other baselines. We utilize Contriever-MSMARCO (Izacard et al., 2022) as the retriever for both data synthesis and inference stages.

We constructed the training data following Section 3.2 with Llama-3-70B-Instruct (Prompts are detailed in Appendix B.2). We trained our model on 4× Nvidia A100 GPUs for about 10 GPU-hours separately, with AdamW (Loshchilov and Hutter, 2019) optimizer and a learning rate of 5e-6.

5 Results and Analysis

5.1 Retrieval Performance

The model's performance was assessed using the Recall@K metric across three distinct datasets. As presented in Table 2, EfficientRAG achieves notably high recall scores on HotpotQA and 2WikiMQA datasets, with recall values of 81.84 and 84.08, respectively. These results are impressive considering the minimal number of chunks retrieved 6.41 for HotpotQA and 3.69 for 2WikiMQA. However, the performance of EfficientRAG on the MuSiQue dataset was less satisfactory. This suboptimal result may be attributed to the smaller number of chunks retrieved and the increased complexity of the dataset.

We further evaluate the QnA performance on the three datasets. As is illustrated in Table 3, our EfficientRAG framework achieves the second-highest accuracy on both HotpotQA and 2WikiMQA, and it also performs well on MuSiQue even with low recall.

Those LLM-based systems perform unsatisfying since they require LLMs to generate partial answers with noisy knowledge inputs, but they always fail in the intermediate steps. We posit that more helpful knowledge and fewer irrelevant chunks are the key points to the RAG system, even a simple model can beat LLMs with the correct RAG paradigm.

5.2 Inference Efficiency

We randomly selected 200 samples from the MusiQue dataset for empirical research and calculated four indicators: LLM calls, iterations, latency, and GPU utilization. As shown in table 4 our method requires fewer iterations and achieves a 60%-80% improvement in time efficiency compared to other iterative methods while maintaining similar GPU utilization.

Method	# LLM calls	# Iteration	Latency (s)	GPU utils (%)
Direct	1.00	-	2.16	15.55
Direct-R	1.00	-	2.47	35.05
Iter-RetGen iter3	3.00	3.00	9.68	66.37
SelfASK	7.18	3.59	27.47	65.02
EfficientRAG	1.00	2.73	3.62	65.55

Table 4: Efficiency evaluation on different RAG paradigms. EfficientRAG exhibits a speed equivalent to direct retrieval methods and is three times faster than LLM-based baselines while maintaining a similar number of iterations.

5.3 Performance with Various Generators

EfficientRAG can benefit from more powerful generators. As is shown in Table 5, the use of GPT-3.5 as a generator enhances the end-to-end performance of both the baselines and our method. Notably, EfficientRAG continues to deliver exceptional results.

Method	EM	F1	Acc
Direct	27.85	33.22	33.79
CoT	38.56	47.28	46.62
Direct-R	34.09	39.46	41.07
iter-RetGen iter2	47.51	58.56	58.34
Iter-RetGen iter3	49.41	60.60	60.60
EfficientRAG	49.00	56.93	**61.88**

Table 5: End-to-end QA performance on the 2WikiMQA dataset using GPT-3.5-turbo-1106 generator. EfficientRAG achieves state-of-the-art accuracy.

5.4 Transferability

EfficientRAG demonstrates the flexibility to adapt to a variety of task scenarios without the need for additional downstream training. To evaluate its transferability, we conduct an experiment across the HotpotQA and 2WikiMQA datasets, training the model on one dataset and testing it on the other. As shown in Table 6, our model successfully generalizes across datasets and, in some instances, even outperforms models trained on the original dataset. These results highlight that EfficientRAG does not depend heavily on domain-specific knowledge, exhibiting robust adaptability across diverse tasks.

Test Set	Training Set	EM	F1	Acc
HotpotQA	HotpotQA	50.59	57.93	57.86
HotpotQA	2WikiMQA	43.38	49.70	53.38
2WikiMQA	2WikiMQA	44.18	51.64	53.41
2WikiMQA	HotpotQA	44.54	51.98	56.59

Table 6: Transferability experiments on 2WikiMQA and HotpotQA dataset. EfficientRAG demonstrates remarkable transferability across diverse datasets.

6 Conclusion

In this study, we introduce the EfficientRAG retriever, a novel approach for multi-hop question retrieval that iteratively generates new queries while circumventing the need for large language models. Evaluations across three benchmark datasets demonstrate that EfficientRAG not only achieves high recall with a minimal number of retrieved chunks but also delivers promising outcomes in subsequent question-answering tasks. These findings indicate that EfficientRAG outperforms traditional retrieval-augmented generation methods, particularly in the context of complex, multi-hop question-answering scenarios.

Limitations

The EfficientRAG framework can theoretically adapt to other models, but we opt not to implement a larger LLM as the final QnA reasoner due to time and resource limits. We analyze our method mainly on open-domain datasets, as it is hard to identify multi-hop question-answering datasets in in-domain settings.

Ethics Statement

The authors declare no competing interests. The datasets used in the training and evaluation come from publicly available sources and do not contain sensitive content such as personal information.

References

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. 2022. Improving language models by retrieving from trillions of tokens. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 2206–2240. PMLR.

Yuan Gao, Yiheng Zhu, Yuanbin Cao, Yinzhi Zhou, Zhen Wu, Yujie Chen, Shenglan Wu, Haoyuan Hu, and Xinyu Dai. 2024. Dr3: Ask large language models not to give off-topic answers in open domain multi-hop question answering. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*, pages 5350–5364. ELRA and ICCL.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *CoRR*, abs/2312.10997.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Retrieval augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 3929–3938. PMLR.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: decoding-enhanced bert with disentangled attention. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020a. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020b. Constructing A multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 6609–6625. International Committee on Computational Linguistics.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *CoRR*, abs/2311.05232.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Trans. Mach. Learn. Res.*, 2022.

Gautier Izacard, Patrick S. H. Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: Few-shot learning with retrieval augmented language models. *J. Mach. Learn. Res.*, 24:251:1–251:43.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023a. Mistral 7b. *CoRR*, 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 2023, Singapore, December 6-10, 2023*, pages 7969–7992. 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. *CoRR*, abs/2212.14024.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting for retrieval-augmented large language models. *CoRR*, abs/2305.14283.

OpenAI. 2022. Introducing chatgpt.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Dongmei Zhang. 2024. Llmlingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. *CoRR*, abs/2403.12968.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 5687–5711. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2383–2392. The Association for Computational Linguistics.

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, Singapore, December 6-10, 2023*, pages 9248–9274. Association for Computational Linguistics.

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. *CoRR*, abs/2301.12652.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and verification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 809–819. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. Newsqa: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP, Rep4NLP@ACL 2017, Vancouver, Canada, August 3, 2017*, pages 191–200. Association for Computational Linguistics.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022a. Musique: Multi-hop questions via single-hop question composition. *Trans. Assoc. Comput. Linguistics*, 10:539–554.

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

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, *ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 10014–10037. Association for Computational Linguistics.

Fangkai Yang, Pu Zhao, Zezhong Wang, Lu Wang, Bo Qiao, Jue Zhang, Mohit Garg, Qingwei Lin, Saravan Rajmohan, and Dongmei Zhang. 2023. Empower large language model to perform better on industrial domain-specific question answering. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: EMNLP 2023 - Industry Track, Singapore, December 6-10, 2023*, pages 294–312. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018a. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2369–2380. Association for Computational Linguistics.

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

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2023. Siren's song in the AI ocean: A survey on hallucination in large language models. *CoRR*, abs/2309.01219.

Tong Zhu, Junfei Ren, Zijian Yu, Mengsong Wu, Guoliang Zhang, Xiaoye Qu, Wenliang Chen, Zhefeng Wang, Baoxing Huai, and Min Zhang. 2023. Mirror: A universal framework for various information extraction tasks. ArXiv:2311.05419 [cs].

A Capability of LLM generator

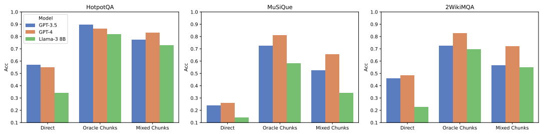


Figure 4: The performance with varying chunks settings over HotpotQA, 2Wiki-Multihop and MuSiQue dataset with GPT-3.5/GPT-4/Llama-3 8B as the generator.

B Prompt List

B.1 Accuracy Evaluation Prompt

You are an experienced linguist who is responsible for evaluating the correctness of the generated responses. You are provided with question, the generated responses and the corresponding ground truth answer. Your task is to compare the generated responses with the ground truth responses and evaluate the correctness of the generated responses. Response in JSON format with key "response" and value "yes" or "no".
Question: {question}
Prediction: {prediction}
Ground-truth Answer: {answer}
Your response:

Table 7: Detailed prompts for GPT-3.5 evaluation.

B.2 Data Synthesize Prompt

Question Decomposition Prompt You are assigned a multi-hop question decomposition task. Your mission is to decompose the original multi-hop question into a list of single-hop sub_questions, based on supporting documents for each sub_question, and such that you can answer each sub_question independently from each document. Each document infers a sub_question id which starts with ‘#‘. The evidence in the document indicates the relation of two entities, in the form of ‘entity1 - relation - entity2‘. The JSON output must contain the following keys: - "question": a string, the original multi-hop question. - "decomposed_questions": a dict of sub_questions and answers. The key should be the sub_question number(string format), and each value should be a dict containing: - "sub_question": a string, the decomposed single-hop sub_question. It MUST NOT contain information more than the original question and its dependencies. NEVER introduce information from documents. - "answer": a string, the answer of the sub_question. - "dependency": a list of sub_question number(string format). If the sub_question relies on the answer of other sub_questions, you should list the sub_question number here. Leave it empty for now because the questions now are all comparison type. - "document": a string, the document id that supports the sub_question. Notice that you don’t need to come out the compare question, just the sub_questions and answers.
Token Labeling Prompt You have been assigned an information extraction task. Your mission is to extract the words from a given paragraph so that others can answer a question using only your extracted words. Your extracted words should cover information from both the question and the answer, including entities (e.g. people, location, film) and core relations. Your response should be in JSON format and include the following key: - "extracted_words": a string composed of a list of words extracted from the paragraph, separated by a space. Please adhere to the following guidelines: - Do not reorder, change, miss, or add words. Keep it the same as the original paragraph. - Identify and extract ONLY the words explicitly mentioned in either the question or its answer, and strongly related to the question or its answer. - NEVER label any words that do not contribute meaningful information to the question or answer. - Only extract words that occurred in the paragraph.
Query Filtering Prompt You are assigned a multi-hop question refactoring task. Given a complex question along with a set of related known information, you are required to refactor the question by applying the principle of retraining difference and removing redundancies. Specifically, you should eliminate the content that is duplicated between the question and the known information, leaving only the parts of the question that have not been answered, and the new knowledge points in the known information. The ultimate goal is to reorganize these retrained parts to form a new question. You can only generate the question by picking words from the question and known information. You should first pick up words from the question, and then from each known info, and concatenate them finally. You are not allowed to add, change, or reorder words. The given known information starts with the word "Info: ". You response should be in JSON format and include the following key: - "filtered_query": a string representing the concatenation of the words from both the question and newly added information, separated by a space. Please adhere to the following guidelines: - Do not reorder, change, or add words. Keep it the same as the original question. - Identify and remove ONLY the words that are already known, keep the unknown information from both the question and information.

Table 8: Detailed prompts for training data construction with Llama-3 70B

B.3 Prompts List

All prompts can be found in this section, and are given in the order of Direct, CoT, Decompose, Direct-R, Iter-RetGen, and Self-ask, as shown in Tables 9 to 22.

Direct Prompting for HotpotQA

~~~~
As an assistant, your task is to answer the question directly after <Question>. Your answer should be after <Answer> in JSON format with key "answer" and its value should be string.
There are some examples for you to refer to:
<Question>: What is the name of this American musician, singer, actor, comedian, and songwriter, who worked with Modern Records and born in December 5, 1932?
<Answer>:
``` json
{{"answer": "Little Richard"}}
```
<Question>: Between Chinua Achebe and Rachel Carson, who had more diverse jobs?
<Answer>:
``` json
{{"answer": "Chinua Achebe"}}
```
<Question>: Remember Me Ballin' is a CD single by Indo G that features an American rapper born in what year?
<Answer>:
``` json
{{"answer": "1979"}}
```
Now your Question is
<Question>: {question}
<Answer>:
~~~~

Direct Prompting for MuSiQue

~~~~
As an assistant, your task is to answer the question directly after <Question>. Your answer should be after <Answer> in JSON format with key "answer" and its value should be string.
There are some examples for you to refer to:
<Question>: In which year did the publisher of In Cold Blood form?
<Answer>:
```json
{{"answer": "2001"}}
```
<Question>: Who was in charge of the city where The Killing of a Sacred Deer was filmed?
<Answer>:
``` json
{{"answer": "John Cranley"}}
```
<Question>: Where on the Avalon Peninsula is the city that Signal Hill overlooks?
<Answer>:
```json
{{"answer": "eastern tip"}}
```
Now your Question is
<Question>: {question}
<Answer>:
~~~~

Direct Prompting for 2Wiki-MultihopQA

~~~~
As an assistant, your task is to answer the question directly after <Question>. Your answer should be after <Answer> in JSON format with key "answer" and its value should be string.
There are some examples for you to refer to:
<Question>: Which film came out first, Blind Shaft or The Mask Of Fu Manchu?
<Answer>:
```json
{{"answer": "The Mask Of Fu Manchu"}}
```
<Question>: When did John V, Prince Of Anhalt-Zerbst's father die?
<Answer>:
```json
{{"answer": "12 June 1516"}}
```
<Question>: Which film has the director who was born later, El Extrano Viaje or Love In Pawn?
<Answer>:
```json
{{"answer": "El Extrano Viaje"}}
```
Now your Question is
<Question>: {question}
<Answer>:
~~~~

Table 9: Detailed prompts for Direct Question Answering with Llama-3 8B

CoT Prompting for HotpotQA

~~~
As an assistant, your task is to answer the question after <Question>. You should first think step by step about the question and give your thought and then answer the <Question>. Your answer should be after <Answer> in JSON format with key "thought" and "answer" and their values should be string.
There are some examples for you to refer to:
<Question>: What is the name of this American musician, singer, actor, comedian, and songwriter, who worked with Modern Records and born in December 5, 1932?
<Answer>:
``` json
{{"thought":"Modern Record is a big R&B label with artists including Etta James, Joe Houston, Little Richard, Ike, Tina Turner and John Lee Hooker in the 1950s and 1960s. Little Richard is an American musician, signer actor and songwriter, born in December 5 1932. So the answer is Little Richard.","answer": "Little Richard"}}
```
<Question>: Between Chinua Achebe and Rachel Carson, who had more diverse jobs?
<Answer>:
``` json
{{"thought":"Chinua Achebe was a Nigerian novelist, poet, professor, and critic. Rachel Carson was an American marine biologist, author, and conservationist. Chinua Achebe has 4 jobs while Rachel Carson has 3 jobs. So the answer is Chinua Achebe.","answer": "Chinua Achebe"}}
```
<Question>: Remember Me Ballin' is a CD single by Indo G that features an American rapper born in what year?
<Answer>:
``` json
{{"thought":"Remember Me Ballin' is the CD singer by Indo G that features Gangsta Boo, who is named Lola Mitchell, an American rapper born in 1979. So the answer is 1979.","answer": "1979"}}
```
Now your Question is
<Question>: {question}
<Answer>:
~~~

Table 10: Detailed prompts for Chain-of-Thought Question Answering with Llama-3 8B On hotpotQA

Question Decomposition Prompt You are assigned a multi-hop question decomposition task.

~~~
You should decompose the given multi-hop question into multiple single-hop questions, and such that you can answer each single-hop question independently.
Your response must be wrapped with ```json and ```.
You should answer in JSON format, your answer must contain the following keys:
- "decomposed_questions": a list of strings, each string is a single-hop question.
Here are some examples for your reference:
## Examples
<Multi-hop question>: Which film came out first, The Love Route or Engal Aasan?
Your response:
```json
{{ "decomposed_questions": [ "When does the film The Love Route come out?", "When does the film Engal Aasan come out?" ] }}
```
<Multi-hop question>: Where did the spouse of Moderen's composer die?
Your response:
```json
{{ "decomposed_questions": [ "Who is Modern's composer?", "Who is the spouse of Carl Nielsen?", "In what place did Anne Marie Carl-Nielsen die?" ] }}
```
<Multi-hop question>: Where was the director of film The Fascist born?
Your response:
```json
{{ "decomposed_questions": [ "Who is the director of film The Fascist?", "Where was Luciano Salce born?" ] }}
```

## Now it's your turn:
<Multi-hop question>: {question}
Your response:
~~~

Table 11: Detailed prompts for multi-hop question decomposition, applicable to all datasets.

CoT Prompting for MuSiQue

~~~~
As an assistant, your task is to answer the question after <Question>. You should first think step by step about the
question and give your thought and then answer the <Question>. Your answer should be after <Answer> in JSON format
with key "thought" and "answer" and their values should be string.
There are some examples for you to refer to:
<Question>: In which year did the publisher of In Cold Blood form?
<Answer>:
```json
{{"thought": "The publisher of In Cold Blood is Random house, which was formed in 2001. So the answer is 2001.",
"answer": "2001"}}
```
<Question>: Who was in charge of the city where The Killing of a Sacred Deer was filmed?
<Answer>:
```json
{{"thought": "The killing of a Scared Deer was filmed in Cincinnati, Ohio, where John Cranley is the mayor. So the
answer is John Cranley.", "answer": "John Cranley"}}
```
<Question>: Where on the Avalon Peninsula is the city that Signal Hill overlooks?
<Answer>:
```json
{{"thought": "Signal Hill overlooks the city St. John's, which is located on the eastern tip of the Avalon Peninsula. So
the answer is eastern tip.", "answer": "eastern tip"}}
```
Now your Question is
<Question>: {question}
<Answer>:
~~~~

Table 12: Detailed prompts for Chain-of-Thought Question Answering with Llama-3 8B on MuSiQue

CoT Prompting for 2Wiki-MultihopQA

~~~~
As an assistant, your task is to answer the question after <Question>. You should first think step by step about the
question and give your thought and then answer the <Question>. Your answer should be after <Answer> in JSON format
with key "thought" and "answer" and their values should be string.
There are some examples for you to refer to:
<Question>: Which film came out first, Blind Shaft or The Mask Of Fu Manchu?
<Answer>:
```json
{{"thought": "Blind Shaft is a 2003 Chinese film, and The Mask Of Fu Manchu is a 1932 American pre-Code adventure
film. The Mask Of Fu Manchu came out first. So the answer is The Mask Of Fu Manchu.", "answer": "The Mask Of Fu
Manchu"}}
```
<Question>: When did John V, Prince Of Anhalt-Zerbst's father die?
<Answer>:
```json
{{"thought": "The father of John V, Prince Of Anhalt-Zerbst is Ernest I, Prince of Anhalt-Dessau. He died on 12 June
1516. So the answer is 12 June 1516.", "answer": "12 June 1516"}}
```
<Question>: Which film has the director who was born later, El Extrano Viaje or Love In Pawn?
<Answer>:
```json
{{"thought": "The director of El Extrano Viaje is Fernando Fernan Gomez, he was born on 29 August 1921. The director
of Love In Pawn is Charles Saunders, he was born on 8 April 1904. Fernando Fernan Gomez was born later, so film El
Extrano Viaje has the director who was born later. So the answer is El Extrano Viaje.", "answer": "El Extrano Viaje"}}
```
Now your Question is
<Question>: {question}
<Answer>:
~~~~

Table 13: Detailed prompts for Chain-of-Thought Question Answering with Llama-3 8B on 2Wiki-MultihopQA

Retrieval Prompting for HotpotQA

~~~
Answer the given question in JSON format, you can refer to the document provided.
As an assistant, your task is to answer the question based on the given knowledge. Your answer should be after <Answer>
in JSON format with key "answer" and its value should be string.
The given knowledge will be embraced by <doc> and </doc> tags. You can refer to the knowledge to answer the question.
If the knowledge does not contain the answer, answer the question directly.
There are some examples for you to refer to:
<doc>
{{KNOWLEDGE FOR YOUR REFERENCE}}
</doc>
<Question>: What is the name of this American musician, singer, actor, comedian, and songwriter, who worked with
Modern Records and born in December 5, 1932?
<Answer>:
```json
{{"answer": "Little Richard"}}
```
<doc>
{{KNOWLEDGE FOR YOUR REFERENCE}}
</doc>
<Question>: Between Chinua Achebe and Rachel Carson, who had more diverse jobs?
<Answer>:
```json
{{"answer": "Chinua Achebe"}}
```
<doc>
{{KNOWLEDGE FOR YOUR REFERENCE}}
</doc>
<Question>: Remember Me Ballin' is a CD single by Indo G that features an American rapper born in what year?
<Answer>:
```json
{{"answer": "1979"}}
```
Now your question and reference knowledge are as follows.
<doc>
{knowledge}
</doc>
<Question>: {question}
<Answer>:
~~~

Table 14: Detailed prompt for retrieval on HotpotQA

Retrieval Prompting for MuSiQue

~~~
As an assistant, your task is to answer the question based on the given knowledge. Your answer should be after <Answer>
in JSON format with key "answer" and its value should be string.
The given knowledge will be embraced by <doc> and </doc> tags. You can refer to the knowledge to answer the question.
If the knowledge does not contain the answer, answer the question directly.
There are some examples for you to refer to:
<doc>
{{KNOWLEDGE FOR YOUR REFERENCE}}
</doc>
<Question>: In which year did the publisher of In Cold Blood form?
<Answer>:
``` json
{{"answer": "2001"}}
```
<doc>
{{KNOWLEDGE FOR YOUR REFERENCE}}
</doc>
<Question>: Who was in charge of the city where The Killing of a Sacred Deer was filmed?
<Answer>:
``` json
{{"answer": "John Cranley"}}
```
<doc>
{{KNOWLEDGE FOR YOUR REFERENCE}}
</doc>
<Question>: Where on the Avalon Peninsula is the city that Signal Hill overlooks?
<Answer>:
``` json
{{"answer": "eastern tip"}}
```
Now your question and reference knowledge are as follows.
<doc>
{knowledge}
</doc>
<Question>: {question}
<Answer>:
~~~

Table 15: Detailed prompt for retrieval on MuSiQue

Retrieval Prompting for 2Wiki-MultihopQA

~~~
As an assistant, your task is to answer the question based on the given knowledge. Your answer should be after <Answer>
in JSON format with key "answer" and its value should be string.
The given knowledge will be embraced by <doc> and </doc> tags. You can refer to the knowledge to answer the question.
If the knowledge does not contain the answer, answer the question directly.
There are some examples for you to refer to:
<doc>
{{KNOWLEDGE FOR YOUR REFERENCE}}
</doc>
<Question>: Which film came out first, Blind Shaft or The Mask Of Fu Manchu?
<Answer>:
```json
{{"answer": "The Mask Of Fu Manchu"}}
```
<doc>
{{KNOWLEDGE FOR YOUR REFERENCE}}
</doc>
<Question>: When did John V, Prince Of Anhalt-Zerbst's father die?
<Answer>:
```json
{{"answer": "12 June 1516"}}
```
<doc>
{{KNOWLEDGE FOR YOUR REFERENCE}}
</doc>
<Question>: Which film has the director who was born later, El Extrano Viaje or Love In Pawn?
<Answer>:
```json
{{"answer": "El Extrano Viaje"}}
```
Now your question and reference knowledge are as follows.
<doc>
{knowledge}
</doc>
<Question>: {question}
<Answer>:
~~~

Table 16: Detailed prompt for retrieval on 2Wiki-MultihopQA

Iter-RetGen Prompting for HotpotQA

~~~~
You should think step by step and answer the question after <Question> based on given knowledge embraced with <doc> and </doc>. Your answer should be after <Answer> in JSON format with key "thought" and "answer", their value should be string.
Here are some examples for you to refer to:
<doc>
{{KNOWLEDGE FOR THE QUESTION}}
</doc>
<Question>: What is the name of this American musician, singer, actor, comedian, and songwriter, who worked with Modern Records and born in December 5, 1932?
Let's think step by step.
<Answer>:
```json
{{ "thought": "Artists who worked with Modern Records include Etta James, Joe Houston, Little Richard, Ike and Tina Turner and John Lee Hooker in the 1950s and 1960s. Of these Little Richard, born in December 5, 1932, was an American musician, singer, actor, comedian, and songwriter.", "answer": "Little Richard" }}
```
<doc>
{{KNOWLEDGE FOR THE QUESTION}}
</doc>
<Question>: Between Chinua Achebe and Rachel Carson, who had more diverse jobs?
<Answer>:
```json
{{ "thought": "Chinua Achebe was a Nigerian novelist, poet, professor, and critic. Rachel Carson was an American marine biologist, author, and conservationist. So Chinua Achebe had 4 jobs, while Rachel Carson had 3 jobs. Chinua Achebe had more diverse jobs than Rachel Carson.", "answer": "Chinua Achebe" }}
```
<doc>
{{KNOWLEDGE FOR THE QUESTION}}
</doc>
<Question>: Remember Me Ballin' is a CD single by Indo G that features an American rapper born in what year?
<Answer>:
```json
{{ "thought": "Remember Me Ballin' is the CD single by Indo G featuring Gangsta Boo. Gangsta Boo is Lola Mitchell's stage name, who was born in August 7, 1979, and is an American rapper.", "answer": "1979" }}
```
Now based on the given doc, answer the question after <Question>.
<doc>
{documents}
</doc>
<Question>: {question}
Let's think step by step.
<Answer>:
~~~~

Table 17: Detailed prompt for Iter-RetGen on HotpotQA

Iter-RetGen Prompting for MuSiQue

~~~
You should think step by step and answer the question after <Question> based on given knowledge embraced with <doc> and </doc>. Your answer should be after <Answer> in JSON format with key "thought" and "answer", their value should be string.
Here are some examples for you to refer to:
<doc>
{{KNOWLEDGE FOR THE QUESTION}}
</doc>
<Question>: In which year did the publisher of In Cold Blood form?
Let’s think step by step.
<Answer>:
```json
{{ "thought": "In Cold Blood was first published in book form by Random House. Random House was form in 2001.", "answer": "2011" }}
```
<doc>
{{KNOWLEDGE FOR THE QUESTION}}
</doc>
<Question>: Who was in charge of the city where The Killing of a Sacred Deer was filmed?
Let’s think step by step.
<Answer>:
```json
{{ "thought": "The Killing of a Sacred Deer was filmed in Cincinnati. The present Mayor of Cincinnati is John Cranley. Therefore, John Cranley is in charge of the city.", "answer": "John Cranley" }}
```
<doc>
{{KNOWLEDGE FOR THE QUESTION}}
</doc>
<Question>: Where on the Avalon Peninsula is the city that Signal Hill overlooks?
Let’s think step by step.
<Answer>:
```json
{{ "thought": "Signal Hill is a hill which overlooks the city of St. John’s. St. John’s is located on the eastern tip of the Avalon Peninsula.", "answer": "eastern tip" }}
```
Now based on the given doc, answer the question after <Question>.
<doc>
{documents}
</doc>
<Question>: {question}
Let’s think step by step.
<Answer>:
~~~

Table 18: Detailed prompt for Iter-RetGen on MuSiQue

Iter-RetGen Prompting for 2Wiki-MultihopQA

~~~~
You should think step by step and answer the question after <Question> based on given knowledge embraced with <doc> and </doc>. Your answer should be after <Answer> in JSON format with key "thought" and "answer", their value should be string.
Here are some examples for you to refer to:
<doc>
{{KNOWLEDGE FOR THE QUESTION}}
</doc>
<Question>: Which film came out first, Blind Shaft or The Mask Of Fu Manchu?
Let's think step by step.
<Answer>:
` ` ` json
{{ "thought": "Blind Shaft is a 2003 film, while The Mask Of Fu Manchu opened in New York on December 2, 1932. 2003 comes after 1932. Therefore, The Mask Of Fu Manchu came out earlier than Blind Shaft.", "answer": "The Mask Of Fu Manchu" }}
` ` `
<doc>
{{KNOWLEDGE FOR THE QUESTION}}
</doc>
<Question>: When did John V, Prince Of Anhalt-Zerbst's father die?
Let's think step by step.
<Answer>:
` ` ` json
{{ "thought": "John V, Prince Of Anhalt-Zerbst was the son of Ernest I, Prince of Anhalt-Dessau. Ernest I, Prince of Anhalt-Dessau died on 12 June 1516.", "answer": "12 June 1516" }}
` ` `
<doc>
{{KNOWLEDGE FOR THE QUESTION}}
</doc>
<Question>: Which film has the director who was born later, El Extrano Viaje or Love In Pawn?
Let's think step by step.
<Answer>:
` ` ` json
{{ "thought": "The director of El Extrano Viaje is Fernando Fernan Gomez, who was born on 28 August 1921. The director of Love In Pawn is Charles Saunders, who was born on 8 April 1904. 28 August 1921 comes after 8 April 1904. Therefore, Fernando Fernan Gomez was born later than Charles Saunders.", "answer": "El Extrano Viaje" }}
` ` `
Now based on the given doc, answer the question after <Question>
<doc>
{documents}
</doc>
<Question>: {question}
Let's think step by step.
<Answer>:
~~~~

Table 19: Detailed prompt for Iter-RetGen on 2Wiki-MultihopQA

Self-ask Prompting for HotpotQA

```
Solve the question with the given knowledge.
Each line should start with either "Intermediate answer:", "Follow up:", "So the final answer is:", or "Are follow up questions needed here:".
#
Question: What is the name of this American musician, singer, actor, comedian, and songwriter, who worked with Modern Records and born in December 5, 1932?
Are follow up questions needed here: Yes.
Follow up: Who worked with Modern Records?
Intermediate answer: Artists worked with Modern Records include Etta James, Little Richard, Joe Houston, Ike and Tina Turner and John Lee Hooker.
Follow up: Is Etta James an American musician, singer, actor, comedian, and songwriter, and was born in December 5, 1932?
Intermediate answer: Etta James was born in January 25, 1938, not December 5, 1932, so the answer is no.
Follow up: Is Little Richard an American musician, singer, actor, comedian, and songwriter, and was born in December 5, 1932?
Intermediate answer: Yes, Little Richard, born in December 5, 1932, is an American musician, singer, actor, comedian and songwriter.
So the final answer is: Little Richard
#
Question: Between Chinua Achebe and Rachel Carson, who had more diverse jobs?
Are follow up questions needed here: Yes.
Follow up: What jobs did Chinua Achebe have?
Intermediate answer: Chinua Achebe was a Nigerian (1) novelist, (2) poet, (3) professor, and (4) critic, so Chinua Achebe had 4 jobs.
Follow up: What jobs did Rachel Carson have?
Intermediate answer: Rachel Carson was an American (1) marine biologist, (2) author, and (3) conservationist, so Rachel Carson had 3 jobs.
Follow up: Did Chinua Achebe have more jobs than Rachel Carson?
Intermediate answer: Chinua Achebe had 4 jobs, while Rachel Carson had 3 jobs. 4 is greater than 3, so yes, Chinua Achebe had more jobs.
So the final answer is: Chinua Achebe
#
Question: Remember Me Ballin' is a CD single by Indo G that features an American rapper born in what year?
Are follow up questions needed here: Yes.
Follow up: Which American rapper is featured by Remember Me Ballin', a CD single by Indo G?
Intermediate answer: Gangsta Boo
Follow up: In which year was Gangsta Boo born?
Intermediate answer: Gangsta Boo was born in August 7, 1979, so the answer is 1979.
So the final answer is: 1979
#
Question: {question}
Are follow up questions needed here:
```

Table 20: Detailed prompt for self-ask on HotpotQA

Self-ask Prompting for MuSiQue

```
Solve the question with the given knowledge.
Each line should start with either "Intermediate answer:", "Follow up:", "So the final answer is:", or "Are follow up questions needed here:".
#
Question: In which year did the publisher of In Cold Blood form?
Are follow up questions needed here: Yes.
Follow up: What business published In Cold Blood?
Intermediate answer: In Cold Blood was published in book form by Random House.
Follow up: Which year witnessed the formation of Random House?
Intermediate answer: Random House was form in 2001.
So the final answer is: 2001
#
Question: Who was in charge of the city where The Killing of a Sacred Deer was filmed?
Are follow up questions needed here: Yes.
Follow up: In which city was The Killing of a Sacred Deer filmed
Intermediate answer: The Killing of a Sacred Deer was filmed in Cincinnati.
Follow up: Who was in charge of Cincinnati?
Intermediate answer: The present Mayor of Cincinnati is John Cranley, so John Cranley is in charge.
So the final answer is: John Cranley
#
Question: Where on the Avalon Peninsula is the city that Signal Hill overlooks?
Are follow up questions needed here: Yes.
Follow up: What city does Signal Hill overlook?
Intermediate answer: Signal Hill is a hill which overlooks the city of St. John's.
Follow up: Where on the Avalon Peninsula is St. John's located?
Intermediate answer: St. John's is located on the eastern tip of the Avalon Peninsula.
So the final answer is: eastern tip
#
Question: {question}
Are follow up questions needed here:
```

Table 21: Detailed prompt for self-ask on MuSiQue

Self-ask Prompting for 2Wiki-MultihopQA

```
Solve the question with the given knowledge.
Each line should start with either "Intermediate answer:", "Follow up:", "So the final answer is:", or "Are follow up questions needed here:".
Follow the examples below to answer the questions with natural language.
#
Question: Which film came out first, Blind Shaft or The Mask Of Fu Manchu?
Are follow up questions needed here: Yes.
Follow up: When did Blind Shaft come out?
Intermediate answer: Blind Shaft came out in 2003.
Follow up: When did The Mask Of Fu Manchu come out?
Intermediate answer: The Mask Of Fu Manchu came out in 1932.
So the final answer is: The Mask Of Fu Manchu
#
Question: When did John V, Prince Of Anhalt-Zerbst's father die?
Are follow up questions needed here: Yes.
Follow up: Who is the father of John V, Prince Of Anhalt-Zerbst?
Intermediate answer: The father of John V, Prince Of Anhalt-Zerbst is Ernest I, Prince of Anhalt-Dessau.
Follow up: When did Ernest I, Prince of Anhalt-Dessau die?
Intermediate answer: Ernest I, Prince of Anhalt-Dessau died on 12 June 1516.
So the final answer is: 12 June 1516
#
Question: Which film has the director who was born later, El Extrano Viaje or Love In Pawn?
Are follow up questions needed here: Yes.
Follow up: Who is the director of El Extrano Viaje?
Intermediate answer: The director of El Extrano Viaje is Fernando Fernan Gomez.
Follow up: Who is the director of Love in Pawn?
Intermediate answer: The director of Love in Pawn is Charles Saunders.
Follow up: When was Fernando Fernan Gomez born?
Intermediate answer: Fernando Fernan Gomez was born on 28 August 1921.
Follow up: When was Charles Saunders (director) born?
Intermediate answer: Charles Saunders was born on 8 April 1904.
So the final answer is: El Extrano Viaje
#
Question: {question}
Are follow up questions needed here:
```

Table 22: Detailed prompt for self-ask on 2Wiki-MultihopQA