CuriousLLM: Elevating Multi-Document QA with Reasoning-Infused Knowledge Graph Prompting

Zukang Yang
University of California, Berkeley
zukangy@berkeley.edu

Clara Zhu
University of California, Berkeley
zzhu248@berkeley.edu

Abstract

In the field of Question Answering (QA), unifying large language models (LLMs) with external databases has shown great success. However, these methods often fall short in providing the advanced reasoning needed for complex QA tasks. To address these issues, we improve over a novel approach called Knowledge Graph Prompting (KGP), which combines knowledge graphs with a LLM-based traversal agent to improve search accuracy. Nevertheless, the original KGP framework necessitates costly fine-tuning with large datasets. Therefore, we propose a reasoning-infused LLM agent to enhance this framework. This agent mimics human curiosity to ask follow-up questions to more efficiently navigate the search. This simple modification significantly boosts the LLMs' performance in multi-hop QA tasks without the high costs and latency associated with the initial KGP framework. Our ultimate goal is to further develop this approach, leading to more accurate, faster, and cost-effective solutions in the multi-document QA domain. The source code is provided here [1].

1 Introduction

Despite their advanced capabilities, LLMs face challenges like hallucination and outdated knowledge (Ji et al., 2023b; Yao et al., 2023a; Xu et al., 2024; Wei et al., 2024; Ji et al., 2023a). To mitigate these issues, current efforts have focused on vector-based information retrieval techniques (Gao et al., 2024; Brundha and Meera, 2022; Yubo et al., 2011) to arm LLMs with current and relevant information. Despite their effectiveness, these methods fall short of addressing the intricate reasoning required to navigate information which spreads across multiple documents.

Our questions generally fall into two categories shown in *Figure 1*: bridging and comparison. Bridging questions require a sequence of logical reasoning, building connections between different pieces of information; while comparison questions necessitate parallel consideration of various documents to evaluate differences or similarities (Wang et al., 2023). Thus, both question types often require synthesizing information from multiple frequently unrelated documents. While heuristic methods, such as TF-IDF (Ramos, 2003) and BM25 (Trotman et al., 2014; Chen and Wiseman, 2023), and Deep Leaning (DL)-based approaches, such as DPR (Karpukhin et al., 2020) and MDR (Xiong et al., 2021), have enabled more efficient information search, it remains a challenge in multi-document QA tasks.

Given above challenges, we propose a novel solution: a reasoning-infused LLM traversal agent designed for conducting multi-hop document searches within a knowledge graph (KG). This KG is constructed with nodes that represent individual passages and edges that symbolize relations among these passages. Our traversal agent, a decoder-only LLM, is fine-tuned to emulate the curious nature of a human researcher: it generates follow-up questions based on both the initial user query and passages retrieved in previous steps. These questions serve as a guide to identify the most relevant neighboring passages for the subsequent hops in the search process. An illustration of the overall logic is shown in *Figure 2*. Particularly, we highlight our contributions as follows:

- **Follow-upQA Dataset.** We have developed a novel dataset named Follow-upQA by leveraging HotpotQA (Yang et al., 2018) and prompt engineering. This dataset consists of a rich collection of questions, facts that support these questions, and follow-up questions that bridge the connections between distinct documents. We also provide a benchmark to inspire further research and development of the dataset.
- **A Reasoning-infused LLM Traversal Agent.**

[1] https://github.com/zukangy/KGP-CuriousLLM.git

We designed a curious LLM excelling in generating contextually relevant follow-up questions to mimic the logical reasoning of human researchers. This model has resulted in a more efficient KGP framework and competitive performance in two multi-hop QA datasets.

- **Comprehensive Experiments Verifying Our Framework.** Through rigorous comparative analysis, we have demonstrated the competitive performance of our framework against several baselines. Our examination sheds light on how each component of our framework contributes to its overall effectiveness, offering a valuable resource for future research endeavors in this field.

2 Related Work

2.1 Retrieval-Based Models

Current retrieval-based models primarily operate by fetching most relevant information from a document collection in response to user queries. Among the traditional methods, TF-IDF and BM25 employ a term-document relevance mechanism to retrieve information with lexical similarity to the user queries. These models work well for questions which share explicit keywords with the target documents. However, these models often struggle when the queries require an understanding in the context or when the necessary information is expressed through synonyms or nuanced language (Lan et al., 2022; Viji and Revathy, 2023; Modi et al., 2023; Cheng et al., 2022).

To bridge this gap, DL-based techniques have been developed. Examples inlcude RNN encoders (Das et al., 2019; Schmidt, 2019; Liu et al., 2019) and BERT-based encoders (Karpukhin et al., 2020; Devlin et al., 2019; Laskar et al., 2020) . These approaches leverage the power of DL to capture the semantic information of texts, thus enabling the retrieval of passages that, while not lexically identical, are semantically similar with the queries. Nevertheless, addressing the complexities of multi-hop QA presents additional challenges.

Additionally, (Xiong et al., 2021) introduced an iterative retrieval method that employs a BERT encoder to fetch a sequence of passages via multi-hop searches. However, these DL-based approaches pose computational burdens. These limitations hinder the scalability and real-time responsiveness of the systems, especially when dealing with vast and dynamically updating datasets.

2.2 Generative Models

The recent advancements in LLMs have enabled models such as GPT (Brown et al., 2020), Llama (Touvron et al., 2023) and Mistral (Jiang et al., 2023) to provide fluent responses to user queries. These models were trained on vast corpora and then further enhanced by Reinforcement Learning (RL) (Ziegler et al., 2020; Rafailov et al., 2023) to effortlessly compose responses mimicking human conversations. Their underlying transformer architecture (Vaswani et al., 2023) excels in capturing the nuanced relationships between words, phrases and sentences. This architecture also equips LLMs with the ability to recall information learned during training, allowing for immediate responses to user queries. However, the time and financial burdens of training, hosting and maintaining a LLM are beyond reach for many of us. Moreover, LLMs are subject to hallucination and knowledge cutoffs, restricting their ability in the QA domain.

2.3 Hybrid Models

Hybrid models represent a fusion of retrieval-based and generative models, equipping LLMs with a document retrieval system to provide relevant contextual information for response generation. This fusion effectively addresses common LLM issues. A popular instance of such hybrid models is Retrieval Augmented Generation (RAG) (Gao et al., 2024). Besides, (Pan et al., 2024) summarizes various strategies to unifying KGs and LLMs, including KG-enhanced LLMs (Shen et al., 2020), LLM-augmented KGs (Zhang et al., 2020; Xie et al., 2022), and Synergized LLMs + KGs (Zhu et al., 2023; Thoppilan et al., 2022). These approaches significantly enhance both LLMs and KGs, facilitating advanced reasoning that leverages both LLMs and world knowledge.

2.4 Knowledge Graph Prompting

Introduced by (Wang et al., 2023), KGP takes advantage of the reasoning power of LLMs to navigate the document search within a KG to achieve a significant improvement in search accuracy. Specifically, during graph traversal, the LLM generates a query based on the initial query and the previously retrieved passages to prompt the search towards the next relevant document. This technique is particularly advantageous in multi-document QA datasets. Building upon the foundation of tree-of-thoughts (Yao et al., 2023b) and KGP, our work enhances

KGP by incorporating a reasoning-infused LLM as the traversal agent, further advancing search speed and accuracy.

3 Methodology

3.1 Knowledge Graph Construction

Knowledge graphs are designed to model real-world knowledge by representing entities as nodes and relations between these entities as edges (Hogan et al., 2021). Formally, we define a knowledge graph as $G = (V, E)$, where $V = \{v_i\}_{i=1}^n$ denotes the set of nodes, and $E \subset V \times V$ represents the relations between pairs of nodes. In the QA domain, this structure allows for the representation of entities like sentences, paragraphs, or entire documents as nodes. These nodes are interconnected through edges that encapsulate various types of relationships, including lexical, semantic, or structural relationships (Wang et al., 2023). An advantage of using a graph is its inherent flexibility in storing and managing additional node or edge features. For example, let $X = \{X_i\}_{i=1}^n$ denote the features associated with nodes, where X_i could be the textual content of the passage, its embedding, or any other attributes of node v_i.

In multi-document QA, unrelated documents could be used together for answering complex questions. Therefore, we design a knowledge graph where passages are interconnected based on semantic similarity. However, we also ensure that passages which collaboratively contribute to answering questions in datasets like HotpotQA are positioned closely within the vector space, regardless of their direct semantic relatedness. This approach enhances the efficiency of our search process, enabling rapid identification of supporting facts even when they are not semantically similar, by ensuring they remain proximate within the graph.

To implemented this idea, we employed the MDR methodology (Xiong et al., 2021) to develop a passage encoder. This encoder is trained to reduce the vector space distance between passages utilized together for answering questions, while increasing the distance between unrelated passages through negative sampling. Then, we constructed our KG by encoding passages and connecting them based on cosine similarity.

3.2 Reasoning-infused LLM Traversal Agent

A KG traversal agent serves to navigate the search process efficiently, aiming to minimize latency while ensuring the accuracy of search results. (Wang et al., 2023) employed a T5 language model (Raffel et al., 2023) as a traversal agent, fine-tuning it to predict subsequent supporting facts. While this method demonstrated efficiency in their experiments, it also raised several concerns. First, predicting the next piece of evidence based on previously retrieved passages can be challenging, especially when the supporting facts for a question are not directly related. Second, this complexity often necessitates a large dataset and significant computational resources, potentially leading to computational bottlenecks and LLM hallucination.

To address these concerns, we introduce a curious LLM as our traversal agent. This approach is rooted in intuitive reasoning: for example, when asked to determine who is older, Bob Bryan or Mariaan de Swardt, and having had information on Bob Bryan's age, our instinct is to inquire about Mariaan de Swardt's age. This intuition forms the basis of our model's fine-tuning to enable it to generate follow-up questions. Specifically, even if facts about the ages of Bob Bryan and Mariaan de Swardt are unrelated, posing a question about Mariaan de Swardt's age creates a logical link between them, directing the search towards the answer. This method of prompting not only mirrors human reasoning more closely but also streamlines the search process by providing a more directed and intuitive path to gathering the necessary information.

After generating a follow-up question, the traversal agent assesses it against the neighboring passages of the current node in the KG. We additionally employ a pre-trained Multi-QA sentence transformer [2], which produces dense representations to minimize the semantic distance between the follow-up question and relevant passages. The search through the KG is conducted using a Breadth-First Search (BFS) strategy. This approach continues until we reach a predefined budget or the model determines that it has gathered sufficient information to respond to the query.

Mathematically, given a user query q_0, we obtain a set of seeding passages $V_j \subset V_s$ with TF-IDF. The agent accepts q_0 and the j-th seeding passage, and then generates a follow-up question q_1^j. Formally,

$$q_{h+1}^j = \underset{v \in N_j}{\arg\max}\, H(q_0, \|_{k=0}^j X_k) \tag{1}$$

[2] https://huggingface.co/sentence-transformers/multi-qa-MiniLM-L6-cos-v1

where $||_{k=0}^{j} X_k$ concatenates the texts of the previously retrieved passages/visited nodes on the same search path till the current node. The choice of H is a language model for next-token prediction. Moreover, the next passage s_{j+1} is obtained as follows:

$$s_{j+1} = \arg\max_{v \in N_j} \phi(g(q_{h+1}^{j}), g(X_n))) \quad (2)$$

where g is a sentence transformer, X_n is all neighboring nodes of node i and ϕ is any similarity functions.

3.3 LLM Response Generation

Upon gathering sufficient evidence, we utilize the capabilities of LLMs to return a human-readable response to the user's query. In this phase, we designed a prompt to direct the LLMs, GPT-3.5 and Mistral-7B [3], towards generating an informed and coherent answer based on the accumulated facts.

3.4 Follow-upQA

With the absence of a dataset specifically designed for our needs, we developed Follow-upQA with GPT-3.5 by leveraging the HotpotQA dataset and prompt engineering. This process involved prompting GPT with the task of "generating a follow-up question" given an initial question and a piece of supporting evidence. To introduce realism, we ensured that 10% of the questions were already complete with the necessary evidence, prompting GPT to respond with “NA” to signal that no further information was needed. This methodology equips the fine-tuned LLM with the dual capability of formulating relevant follow-up questions and recognizing when the information at hand suffices. This process resulted in a dataset of 50K samples.

4 Experiments

For our evaluation, we utilized two validation sets from (Wang et al., 2023), each comprising 500 questions derived from HotpotQA and 2WikiMQA (Ho et al., 2020), respectively. Additionally, to simulate realistic information retrieval challenges, both sets included distracting information collected from Wikipedia. The retrieved articles were segmented into individual sentences, resulting in test sets with 270K and 120K unique passages, respectively, each serving as a distinct node within our constructed knowledge graphs. Given computational constraints, our experiment further narrowed the validation sets to a subset of 100 questions from each validation set, equally divided between bridge and comparison questions, and limited the search space to a 2-hop search. Moreover, we adjusted parameters to ensure each technique retrieved an average of 30 passages for each query.

4.1 Metrics

Accuracy. In this context, accuracy measures the proportion of correctly answered questions. For each evaluation set, we prompted GPT3.5 to compare each generated response with its corresponding ground truth to grade for overall accuracy.

Exact Match (EM). EM evaluates the accuracy of information retrieval by calculating the proportion of facts correctly identified by a retriever in comparison to a set of pre-defined "golden" facts. The implementation introduced by (Xiong et al., 2021) compares retrieved passages to their golden references on a token-by-token basis. However, we noticed that passages from the validation sets do not always align perfectly with their golden counterparts, potentially leading to an underestimation of the true performance. To address this discrepancy, we instead matched passages based on cosine similarity. Our analysis showed that while most accurate matches exhibit similarity scores around 0.99, pairs with scores as low as 0.85 were also deemed equivalent.

4.2 Impact of the LLM Traversal Agent

We employed several retrievers as the traversal agent, including keyword-based TF-IDF and BM25, DL-based MDR, and a strong baseline, KGP-T5, introduced by (Wang et al., 2023). Additionally, we included the "Golden" method by providing GPT-3.5 with golden supporting passages, and the "None" method by giving only the questions. These two methods serve as the lower and upper bound in performance.

From *Table 1*, both KGP models consistently outperformed others, demonstrating their superior ability in handling complex queries. While BM25 displayed competitive accuracy in responding to HotpotQA questions, it fell short in the 2WikiMQA dataset. Interestingly, despite significant variations in accuracy scores, the EM scores for most agents appeared similar in both datasets. A deep dive into the datasets revealed that the golden passages were not the only paths to address many of the

[3] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2

MD-QA Performance on Different Agents (%)												
Method	HotpotQA						2WikiMQA					
	All Questions		50 Bridge		50 Comparison		All Questions		50 Bridge		50 comparison	
	Acc	EM	Acc	EM	Acc	EM	Acc	EM	Acc	EM	Acc	EM
None	50.0	-	40.0	-	60.0	-	20.0	-	16.0	-	24.0	-
TF-IDF	51.0	69.39	60.0	64.23	42.0	**75.93**	27.0	57.20	20.0	59.00	34.0	56.00
BM25	62.0	69.39	66.0	65.69	58.0	74.07	24.0	57.60	22.0	58.00	26.0	**57.33**
MDR	57.0	**71.43**	64.0	**68.61**	40.0	75.00	30.0	**60.80**	28.0	**67.00**	32.0	56.67
KGP_T5	**64.0**	71.02	68.0	**68.61**	**60.0**	74.07	37.0	60.40	40.0	66.00	34.0	56.67
Ours	63.0	70.20	**74.0**	65.69	52.0	**75.93**	**42.0**	59.60	**42.0**	66.00	**42.0**	55.33
Golden	73.0	100	86.0	100	60.0	100	66.0	100	72.0	100	60.0	100

Table 1: The table shows the overall MD-QA performance of several traversal agents across two datasets. "Ours" outperforms most other methods. Our proposed framework is based on KGP_T5 which is used as a strong baseline. The dashes (-) indicate missing data or inapplicable entries in the respective cells.

MD-QA Performance without KG (%)												
Method	HotpotQA						2WikiMQA					
	All Questions		50 Bridge		50 Comparison		All Questions		50 Bridge		50 comparison	
	Acc	EM	Acc	EM	Acc	EM	Acc	EM	Acc	EM	Acc	EM
TF-IDF	59.0	73.47	70.0	64.96	48.0	84.26	29.0	57.20	18.0	59.00	40.0	56.00
BM25	62.0	69.39	66.0	64.23	58.0	75.92	24.0	57.60	22.0	58.00	26.0	**57.33**
MDR	62.0	**81.22**	**72.0**	**72.26**	52.0	**92.59**	30.0	**60.80**	34.0	**67.00**	26.0	56.67
KGP_T5	**64.0**	71.02	68.0	68.61	**60.0**	74.07	37.0	60.40	40.0	66.00	34.0	56.67
Ours	63.0	70.20	**74.0**	65.69	52.0	75.93	**42.0**	59.60	**42.0**	66.00	**42.0**	55.33
Golden	73.0	100	86.0	100	60.0	100	66.0	100	72.0	100	60.0	100

Table 2: The table showcases the overall MD-QA performance for techniques without knowledge graph; and the performance for the two KGP approaches (which used KG) are included for comparison.

questions. This finding sheds light on KGP, despite not always achieving the highest EM scores, still managed to receive competitive accuracy levels. It adeptly utilized a broader range of valid evidence to arrive at the correct answers, proving the flexibility and effectiveness of our approach.

During our evaluation of KGP-T5, we observed instances of hallucination despite fine-tuning. Although the T5 could generate informative search queries, it frequently included extraneous or incorrect words, obscuring the search process. For example, in response to the query, "which magazine was started first: Arthur's Magazine or First for Women?" and knowing the birth year of Arthur's Magazine, an ideal query would seek the founding year of "First for Women" magazine. However, T5 might generate a query referencing the magazine's publication in London from 1921 to 1927, introducing misleading details such as "London" and "from 1921 to 1927." These inaccuracies add unnecessary complexity to subsequent searches. Our proposed model addresses this by simplifying the query to focus strictly on the needed information, like "what year was First for Women magazine started?" This approach ensures that the search is guided by only the most relevant information, enhancing the precision of the search process.

4.3 Impact of the Constructed Knowledge Graph

We implemented three techniques, TF-IDF, BM25 and MDR, to evaluate the effectiveness of our knowledge graph. For TF-IDF and BM25, we configured them to fetch a predetermined number of passages in response to user queries. Conversely, MDR was designed to perform a 2-hop search in the vector space. Neither of the aforementioned techniques were given the knowledge graph.

In *Table 2*, we showcase the performance of TF-IDF, BM25, and MDR, alongside the two KGP models. The metrics indicate that the three methods significantly improved their accuracy and EM compared to their roles as traversal agents, as seen in *Table 1*. This suggests that these methods excel when searching across the entirety of the document space, rather than being limited to the more constrained search space of a KG. Notably, in the 2WikiMQA dataset, the KGP models markedly surpassed the performance of the three techniques. A closer examination indicated that TF-IDF and BM25 struggle with the reasoning required for the MD-QA task, as they often only identified a subset of the necessary facts, especially in 2WikiMQA. Although MDR was specifically designed to enhance reasoning within the search process, it still fell short of the KGP models. This outcome emphasizes the effectiveness of KGs as part of the KGP framework.

4.4 Impact of the Response Generation LLM

In *Table 3*, we compared the response quality generated by GPT-3.5 and Mistral-7B. Mistral-7B demonstrated superior reasoning abilities in comparison questions, thus achieving significantly higher overall accuracy than GPT-3.5. Notably, Mistral-7B enhanced the performance of the TF-IDF technique from 51% to 61%. On the other hand, however, GPT-3.5 presented its stronger reasoning capabilities in bridging questions. Besides, GPT-3.5 exhibited better adherence to instructions, providing concise answers that often comprised no more than 6 tokens. In contrast, Mistral-7B tended to generate longer responses, averaging 15 tokens, sometimes incorporating erroneous or extraneous tokens that impacted readability. This comparison suggests a trade-off between achieving higher accuracy and maintaining readability when selecting a LLM for response generation.

Correct Rates from Two Response Generation LLMs (%)						
Method	HotpotQA					
	All Questions		50 Bridge		50 Comparison	
	GPT	Mistral	GPT	Mistral	GPT	Mistral
None	50.0	34.0	40.0	24.0	44.0	60.0
TF-IDF	51.0	61.0	60.0	60.0	42.0	62.0
BM25	62.0	63.0	66.0	62.0	58.0	64.0
MDR	62.0	62.0	72.0	66.0	52.0	58.0
KGP_T5	64.0	66.0	68.0	62.0	60.0	70.0
Ours	63.0	**71.0**	**74.0**	60.0	52.0	**82.0**
Golden	73.0	87.0	86.0	86.0	60.0	88.0

Table 3: Above table showcases the performance of two response generation models.

4.5 Follow-upQA Benchmark

We evaluated the performance of the Mistral-7B fined-tuned with QLora (Dettmers et al., 2023) and Follow-upQA. We applied a train-validation-test split of 95%-1%-4%, resulting in 2K samples in the test set. We utilized MLX [4] to train the model. Our training included saving checkpoints every 300 epochs over a total of 1,500 epochs. We also conducted a grid search on various decoding parameters, such as temperature, top-p, and maximum token length.

Figure 3 presents histogram distributions and line plots for ROUGE-1 and ROUGE-L scores, both of which concentrate at around 0.4, with peak performances highlighted in the line plots. Additionally, a mean cosine similarity score of around 0.6 indicates a close semantic alignment between the generated and the golden questions. The line plots reveal that the models achieved optimal performance at the 600-epoch mark, with peak performance at epoch 1,200. Notably, the initial performance of the raw Mistral-7B model at epoch 0 was the worst, indicating the gradual improvement of our models in generating more accurate follow-up questions as training progressed.

4.6 Latency Study

While the KGP framework demonstrates superior performance in MD-QA tasks, latency poses as a significant challenge. In our experiments conducted on a MacBook M2-Max, without leveraging parallel computing, the process time varied between models. For instance, under the same setting, KGP-T5 took an average of 90 seconds to gather supporting facts. In contrast, KGP-Mistral showed a 50% improvement in speed, averaging around 45 seconds. However, even with this enhancement, the latency remains a considerable obstacle. Given that KGP operates through the integration of multiple language models, its extended runtime is somewhat expected. Consequently, the choice of framework for information retrieval in practical applications will largely hinge on the available resources and specific requirements of the project, balancing the trade-offs between performance benefits and operational constraints.

5 Conclusion

The Knowledge Graph Prompting framework has showcased its efficacy and potential in multi-hop QA. Through our analysis, we've not only assessed the overall impact of the KGP framework but have also delved into the intricacies of its various components. This examination reveals numerous avenues for further research aimed at optimizing these elements to enhance the framework's efficiency. Moreover, we've established a benchmark on the Follow-upQA dataset, demonstrating the critical role of generating precise follow-up questions in streamlining document retrieval and, by extension, improving QA performance. We are optimistic that this paper will serve as a valuable resource, guiding and inspiring subsequent advancements in the field.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

J. Brundha and K.N. Meera. 2022. Vector model based information retrieval system with word embedding transformation. In *2022 10th International Conference on Emerging Trends in Engineering and Technology - Signal and Information Processing (ICETET-SIP-22)*, pages 01–04.

[4] https://github.com/ml-explore/mlx

Xiaoyin Chen and Sam Wiseman. 2023. Bm25 query augmentation learned end-to-end.

Yanming Cheng, Zhigang Yu, Je Hu, and Mingchuan Yang. 2022. A chinese short text classification method based on tf-idf and gradient boosting decision tree. In *2022 International Conference on Image Processing, Computer Vision and Machine Learning (ICICML)*, pages 164–168.

Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, and Andrew McCallum. 2019. Multi-step retriever-reader interaction for scalable open-domain question answering.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey.

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*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia D'amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann. 2021. Knowledge graphs. *ACM Computing Surveys*, 54(4):1–37.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023a. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Ziwei Ji, Tiezheng Yu, Yan Xu, Nayeon Lee, Etsuko Ishii, and Pascale Fung. 2023b. Towards mitigating LLM hallucination via self reflection. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1827–1843, Singapore. Association for Computational Linguistics.

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. 2023. Mistral 7b.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Fei Lan et al. 2022. Research on text similarity measurement hybrid algorithm with term semantic information and tf-idf method. *Advances in Multimedia*, 2022.

Md Tahmid Rahman Laskar, Enamul Hoque, and Jimmy Xiangji Huang. 2020. Utilizing bidirectional encoder representations from transformers for answer selection.

Ye Liu, Chenwei Zhang, Xiaohui Yan, Yi Chang, and Philip S Yu. 2019. Generative question refinement with deep reinforcement learning in retrieval-based qa system. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1643–1652.

Anshul Modi, Yuvraj Singh Dhanjal, and Anamika Larhgotra. 2023. Semantic similarity for text comparison between textual documents or sentences. In *2023 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES)*, pages 1–5.

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*, page 1–20.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer.

Juan Enrique Ramos. 2003. Using tf-idf to determine word relevance in document queries.

Robin M. Schmidt. 2019. Recurrent neural networks (rnns): A gentle introduction and overview.

Tao Shen, Yi Mao, Pengcheng He, Guodong Long, Adam Trischler, and Weizhu Chen. 2020. Exploiting structured knowledge in text via graph-guided representation learning. *arXiv preprint arXiv:2004.14224*.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

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

Andrew Trotman, Antti Puurula, and Blake Burgess. 2014. Improvements to bm25 and language models examined. In *Proceedings of the 19th Australasian Document Computing Symposium*, pages 58–65.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need.

D Viji and S Revathy. 2023. A hybrid approach of poisson distribution lda with deep siamese bi-lstm and gru model for semantic similarity prediction for text data. *Multimedia Tools and Applications*, 82(24):37221–37248.

Yu Wang, Nedim Lipka, Ryan A. Rossi, Alexa Siu, Ruiyi Zhang, and Tyler Derr. 2023. Knowledge graph prompting for multi-document question answering.

Jiaheng Wei, Yuanshun Yao, Jean-Francois Ton, Hongyi Guo, Andrew Estornell, and Yang Liu. 2024. Measuring and reducing llm hallucination without gold-standard answers via expertise-weighting.

Xin Xie, Ningyu Zhang, Zhoubo Li, Shumin Deng, Hui Chen, Feiyu Xiong, Mosha Chen, and Huajun Chen. 2022. From discrimination to generation: Knowledge graph completion with generative transformer. In *Companion Proceedings of the Web Conference 2022*, pages 162–165.

Wenhan Xiong, Xiang Lorraine Li, Srini Iyer, Jingfei Du, Patrick Lewis, William Yang Wang, Yashar Mehdad, Wen tau Yih, Sebastian Riedel, Douwe Kiela, and Barlas Oğuz. 2021. Answering complex open-domain questions with multi-hop dense retrieval.

Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. 2024. Hallucination is inevitable: An innate limitation of large language models.

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)*.

Jia-Yu Yao, Kun-Peng Ning, Zhen-Hui Liu, Mu-Nan Ning, and Li Yuan. 2023a. Llm lies: Hallucinations are not bugs, but features as adversarial examples.

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

Jia Yubo, Dong Xing, Wang Yi, and Fan Hongdan. 2011. A document-based information retrieval model vector space. In *2011 Second International Conference on Networking and Distributed Computing*, pages 65–68.

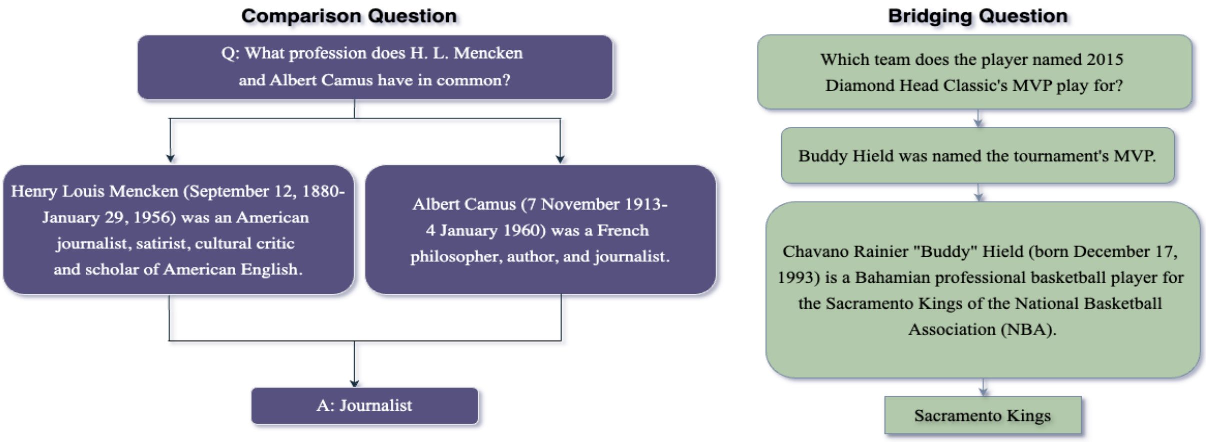
Zhiyuan Zhang, Xiaoqian Liu, Yi Zhang, Qi Su, Xu Sun, and Bin He. 2020. Pretrain-KGE: Learning knowledge representation from pretrained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 259–266, Online. Association for Computational Linguistics.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2020. Fine-tuning language models from human preferences.

A Appendix

A.1 Illustration of the Two Question Types



Q: What profession does H. L. Mencken and Albert Camus have in common?

Henry Louis Mencken (September 12, 1880-January 29, 1956) was an American journalist, satirist, cultural critic and scholar of American English.

Albert Camus (7 November 1913-4 January 1960) was a French philosopher, author, and journalist.

A: Journalist

Which team does the player named 2015 Diamond Head Classic's MVP play for?

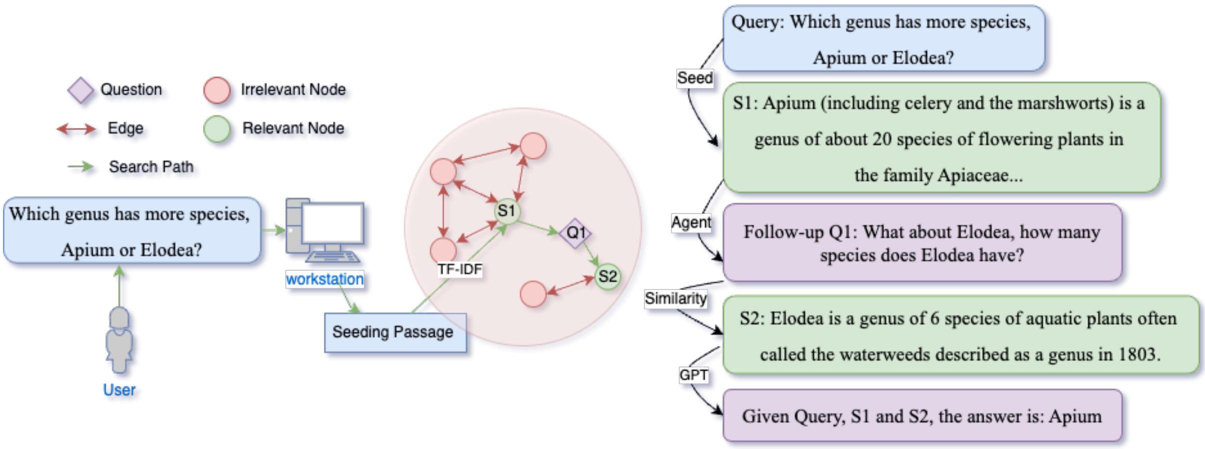
Buddy Hield was named the tournament's MVP.

Chavano Rainier "Buddy" Hield (born December 17, 1993) is a Bahamian professional basketball player for the Sacramento Kings of the National Basketball Association (NBA).

Sacramento Kings

Figure 1: Two common types of questions. **(1) Comparison questions** require parallel reasoning over different documents. **(2) Bridging questions** require sequential reasoning.

A.2 Illustration of the KGP-Mistral Pipeline



Which genus has more species, Apium or Elodea?

Query: Which genus has more species, Apium or Elodea?

S1: Apium (including celery and the marshworts) is a genus of about 20 species of flowering plants in the family Apiaceae...

Follow-up Q1: What about Elodea, how many species does Elodea have?

S2: Elodea is a genus of 6 species of aquatic plants often called the waterweeds described as a genus in 1803.

Given Query, S1 and S2, the answer is: Apium

Figure 2: Given a user query, the system starts search for relevant documents; with follow-up question **Q1** generated by the traversal agent, the unrelated **S1** and **S2** form a search path leading to the final answer.

A.3 Follow-upQA Benchmark

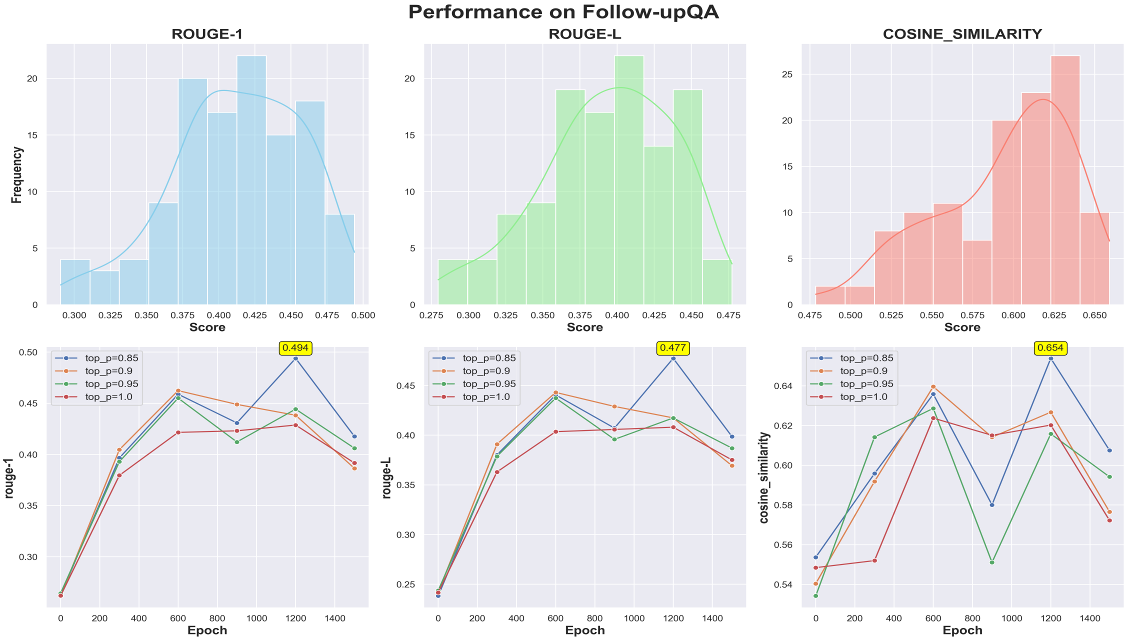


Figure 3: Benchmark for Follow-upQA. The first row shows distribution plots for ROUGE-1 F1, ROUGE-L F1 and cosine similarity across multiple models and varying sets of hyper-parameters. The second row shows the performance of Mistral-7B at different epochs. The peak performance is annotated. The peak performance across three metrics come from the same model: Mistral-7B at epoch $1,200$ with temperature of 0.6, top_p of 0.85 and max_token_len of 50.