CuriousLLM: Elevating Multi-Document Question Answering with LLM-Enhanced Knowledge Graph Reasoning

Zukang Yang[1], Zixuan Zhu[1], Xuan Zhu[1],

[1]School of Information, University of California, Berkeley

Correspondence: zukangy, zzhu248, zhuxuan@berkeley.edu

Abstract

Large Language Models (LLMs) have achieved significant success in open-domain question answering. However, they continue to face challenges such as hallucinations and knowledge cutoffs. These issues can be mitigated through in-context learning by providing LLMs with relevant context before generating answers. Recent literature proposes Knowledge Graph Prompting (KGP) which integrates knowledge graphs with an LLM-based traversal agent to substantially enhance document retrieval quality. However, KGP requires costly fine-tuning with large datasets and remains prone to hallucination. In this paper, we propose CuriousLLM, an enhancement that integrates a curiosity-driven reasoning mechanism into an LLM agent. This mechanism enables the agent to generate relevant follow-up questions, thereby guiding the information retrieval process more efficiently. Central to our approach is the development of the new Follow-upQA dataset, which includes questions and supporting evidence as input, with follow-up questions serving as ground truths. These follow-up questions either inquire about what is still missing to fully answer the user's query or use special tokens to signify that the retrieved evidence is sufficient. Our experiments show that CuriousLLM significantly boosts LLM performance in multi-document question answering (MD-QA), circumventing the substantial computational costs and latency from the original KGP framework. Source code: https://github.com/zukangy/KGP-CuriousLLM.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable success in open-domain question answering. However, they continue to face challenges such as hallucinations and knowledge cutoffs (Ji et al., 2023a,b; Yao et al., 2023; Xu et al., 2024; Wei et al., 2024). To address these issues, recent research has explored in-context learning with LLM, using external knowledge sources such as retrieved documents or knowledge graphs (KG) to improve their accuracy and reasoning ability (Hogan et al., 2021; Pan et al., 2024; Agrawal et al., 2024; Shen et al., 2020; Zhang et al., 2019; Rosset et al., 2021; Zhang et al., 2020; Kumar et al., 2020a; Zhu et al., 2023a). Among the popular approaches are RAG (Lewis et al., 2020) and KAPING (Baek et al., 2023), which provide context by retrieving relevant documents or KG triplets to support the LLM reasoning process. Although these methods have proven effective, few approaches have fully integrated LLMs into the retrieval process, leaving a gap in the ability to efficiently navigate and extract relevant information from vast knowledge sources.

Recently, Wang et al. (2023) propose Knowledge Graph Prompting (KGP), which incorporates a fine-tuned LLM agent into the KG traversal process. This agent predicts missing evidence based on the initial query and the retrieved documents. The predictions are then used to identify relevant passages from neighboring nodes in the KG through similarity ranking. With this prompt reformulation approach, KGP achieves state-of-the-art results in several benchmarks for factual consistency.

However, during our experiments, we notice that the original KGP technique involves fine-tuning a T5 (Raffel et al., 2020) agent with a large corpus. Despite this, the agent's performance remains limited due to hallucination: While correctly predicts missing evidence, it often uses irrelevant or erroneous keywords, which obscure the search for the actual piece of missing evidence, as shown in Table 1. Additionally, this technique tends to exhaust its preset search budget because it lacks a mechanism to determine when to stop the search, even when the retrieved documents are sufficient to answer the question. The more passages supplied to the LLM for response generation, the higher the latency of the QA system, since the LLM must reason through

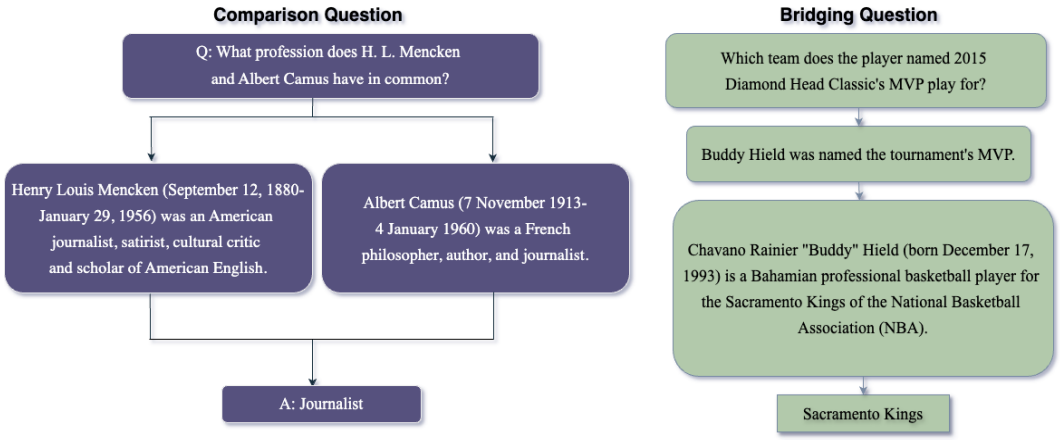

Figure 1: Two common types of questions in HotpotQA (Yang et al., 2018): (1) Comparison questions require parallel reasoning over different documents. (2) Bridging questions require sequential reasoning.

all passages to arrive at an answer.

To address these challenges, we propose a novel framework by fine-tuning an LLM agent to emulate the curious nature of a human researcher. Instead of predicting missing evidence, our agent asks follow-up questions to more efficiently guide the search toward missing evidence. Compared to the original approach, our CuriousLLM: 1) requires significantly fewer training samples for fine-tuning, 2) markedly improves MD-QA performance, and 3) can terminate the search before exhausting the preset search budget, thereby reducing latency. Our contributions are as follows.

- **Follow-upQA Dataset:** We introduce a new dataset specifically designed to train LLMs to generate pertinent follow-up questions that enhance the retrieval process within the KGP framework for MD-QA tasks. In addition, we offer this dataset as a benchmark to inspire further research.
- **CuriousLLM Agent:** We design an LLM agent to ask follow-up questions, thus improving the efficiency and precision of the KGP framework without requiring extensive fine-tuning.
- **Experimental Validation:** We present comprehensive experimental results that demonstrate significant improvements in both the performance and efficiency of the KGP framework using our approach. Furthermore, an ablation study highlights the enhanced reasoning capabilities of our LLM agent, particularly after fine-tuning on the Follow-upQA dataset.

2 Methodology

2.1 Follow-upQA Dataset

We derive the new Follow-upQA dataset from the HotpotQA dataset (Yang et al., 2018). HotpotQA is a multi-hop QA dataset containing questions and supporting passage pairs collected from Wikipedia. The questions in the dataset have three key features: 1) they cover a wide variety of topics; 2) they primarily consist of comparison and bridging questions (Figure 1), both of which are common in MD-QA tasks; and 3) they can be answered with at most two supporting passages (Xiong et al., 2021). The first two features allow our LLM agent to learn from a diverse range of topics across both types of questions, while the third feature simplifies implementation and facilitates demonstration of Follow-upQA.

We create Follow-upQA through the following steps: First, we randomly sample questions from HotpotQA without replacement. Second, for the comparison questions, we randomly remove one of the two supporting passages. For the bridging questions, which require sequential reasoning, we retain only the first passage in the reasoning sequence. Next, we ask GPT-3.5 to generate a follow-up question based on the initial question and the supporting passage provided. Finally, if the selected question is a single-hop question, we prompt GPT-3.5-turbo [1] to respond with "NA" to indicate that no further information is needed. In these steps, the follow-up question or the "NA" response serves as the ground truth for the question. We repeat this process until

[1] https://platform.openai.com/docs/models/gpt-3-5

Question:	Which magazine was started first: Arthur's Magazine or First for Women?
Retrived Evidence:	Arthur's Magazine (1844 - 1846) was an American literary periodical published in Philadelphia in the 19th century.
Missing Evidence:	First for Women is a woman's magazine published by Bauer Media Group in the USA. The magazine was started in 1989.
T5 Prediction:	The publication of a woman's magazine is in London from 1921 to 1927.
CuriousLLM Prediction:	When was First for Women Magazine first started?

Table 1: Instance of T5 hallucination during graph traversal. Due to the erroneous keywords, T5 fails to identify the missing evidence. However, CuriousLLM succeeds in finding the missing evidence.

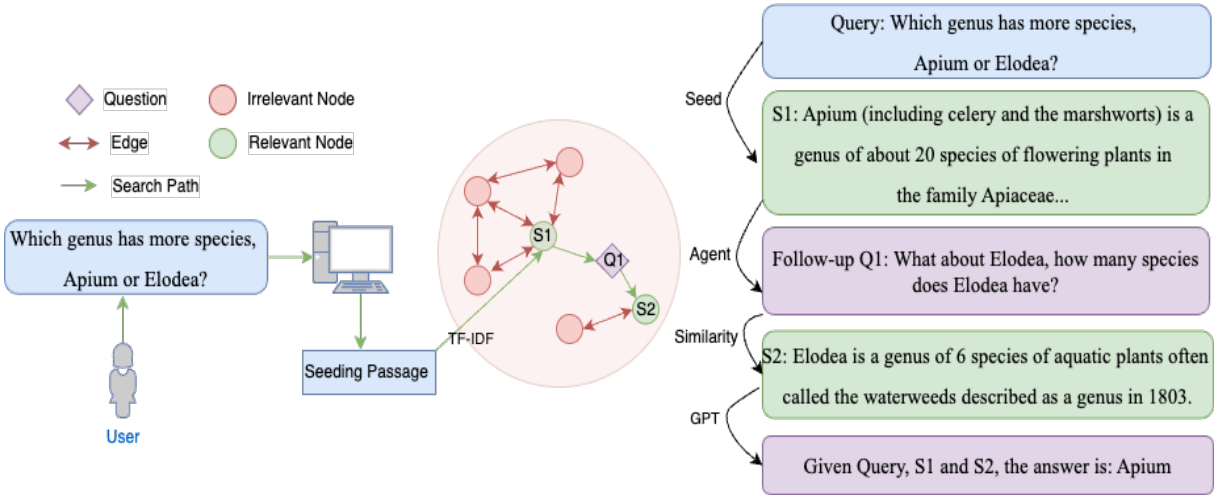

Figure 2: Overview of the CuriousLLM workflow and an Follow-upQA example. Given a query, the system obtains seeding passages, and then starts searching for relevant documents; with follow-up question **Q1** generated by the LLM agent, the unrelated passages S1 and S2 form a search path leading to the final answer.

the budget is reached.

This process yields Follow-upQA, a dataset of $\sim 50K$ samples, with 59.5% bridging questions, 26% comparison questions, and 14.5% "NA" or single-hop questions.

In Table 2, we present examples of MD-QA tasks, illustrating the types of input questions and the corresponding given information, as well as the generated follow-up questions. These examples highlight how MD-QA systems, including our proposed model, are designed to handle complex questions that require reasoning across multiple pieces of evidence. For example 1, the follow-up question guides the graph traversal to look for information about University of Missouri's location. On the other hand, example 2 shows that the output is a special token "NA", signaling that sufficient information has been collected.

2.2 Knowledge Graph Construction

Formally, we define a KG as $G = (V, E, X)$, 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 our experimentation, each v_i represents a passage and $X = \{x_i\}_{i=1}^{n}$ denotes a collection of dense representations with x_i representing the passage embedding for v_i.

In the original KGP experiments, constructing KG by multi-hop dense retriever (MDR-KG) (Wang et al., 2023) outperforms other KG construction approaches, such as k-nearest neighbors (KNN) (Cunningham and Delany, 2021) and TF-IDF. Following the methodology for building MDR-KG , we employ the MDR training technique (Xiong et al., 2021) to develop a BERT-based passage encoder using the HotpotQA dataset. This encoder is trained to predict subsequent supporting passages given an initial retrieved passage. The goal is to minimize the distances between passage pairs that are used together to answer questions in HotpotQA while simultaneously increasing the distances between unrelated passages through negative sampling. This training technique equips the encoder with reasoning capabilities, enabling it to understand the logical associations between different passages. Finally, we construct our KGs by

Example 1	Example 2
(Input) Question: In what city is the university, for which Kim English played college basketball, located?	**(Input) Question:** Tina Charles and Maya Moore were teammates on the UConn women's team that won championships in what years?
(Input) Given: Kim English played college basketball for the University of Missouri before being selected by the Detroit Pistons with the 44th overall pick in the 2012 NBA draft.	**(Input) Given:** In 2009 and 2010, Tina Charles and her teammate Maya Moore led the Connecticut Huskies to two undefeated national championships.
(Ground Truth) Follow-up Question: In which city is the University of Missouri located?	**(Ground Truth) Follow-up Question:** NA

Table 2: Follow-upQA examples of input questions, given information, and the corresponding follow-up questions generated.

encoding passages and connecting them based on cosine similarity.

2.3 Curious LLM Traversal Agent

We introduce CuriousLLM as our graph traversal agent to enhance the KGP framework. This approach is rooted in intuitive reasoning. For example, when asked to determine who is older, Bob Bryan or Mariaan de Swardt, and given information about Bob's age, one would intuitively ask a follow-up question about Mariaan's age. This follow-up question guides the search for relevant information. This intuition forms the basis of Follow-upQA and our model's training objective. The advantage of this approach is that, although the passages about Bob's and Mariaan's ages are unrelated, or, in other words, not semantically similar, the follow-up question about Mariaan's age creates a logical link between them. As a result, instead of matching two unrelated passages, once we find one passage, we can identify the other through follow-up questions. Furthermore, we train the LLM agent to know when to end the search, and this early termination mechanism significantly reduces the latency associated with the original T5 agent.

We use the Mistral-7B model (Jiang et al., 2023) [2] as the backbone and fine-tune it on the Follow-upQA dataset with the objective of next-token prediction. Specifically, for each sample in the dataset, as shown in Table 2, we concatenate the question, the given passage, and the follow-up question. We employ QLoRA (Dettmers et al., 2023) to train the model: we load the model at 8 bit precision and then train a LoRA adapter using a LoRA rank of 32, a learning rate of 10^{-5}, and a batch size of 12. The training is implemented with a split between the training validation test of 90% - 5% - 5%. Compared to the time required to train an identical T5 model in the original KGP framework, we reduce the training time of our model by 85% using the same computing resources.

After generating a follow-up question at the current node in the KG, the question is compared against the neighboring passages of the node. We also employ a pre-trained Multi-QA sentence transformer [3], which produces dense representations to minimize the semantic distance between the follow-up question and relevant passages. The search through KG is carried out using a breadth-first search strategy (BFS), as shown in Algorithm 1. This iterative process continues until reaching a predefined budget or the model deems it has sufficient information to answer the query.

Mathematically, given a user query q_0, we obtain a set of seeding passages $v_j \subset \mathcal{V}^s$ 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,

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

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

$$s_{j+1} = \underset{v \in N_j}{\arg\max}\ \phi(g(q_h^j), g(X_n))) \tag{2}$$

where g is a sentence transformer, X_n is passages

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

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

from all neighboring nodes of node v_j and ϕ is any similarity functions. See Algorithm 2.

2.4 LLM Response Generation

After gathering sufficient evidence, we leverage LLMs' capabilities to provide a human-readable response to the user's query. We applied prompt engineering to guide the GPT4o-mini [4], using the accumulated facts to generate an informed and coherent answer.

3 Experiments

To evaluate the multi-document question answering (MD-QA) capabilities of our CuriousLLM agent, KGP-Mistral, we adopt the experimental setup used by KGP-T5 (Wang et al., 2023). In the original KGP-T5 evaluation, four MD-QA validation sets are used, each comprising 500 questions sampled from the following datasets: HotpotQA, 2WikiMQA (Ho et al., 2020), IIRC (Ferguson et al., 2020), and MuSiQue (Trivedi et al., 2022). These datasets include $270K$, $120K$, $470K$, and $173K$ unique passages, respectively, which serve as supporting evidence or distracting passages.

Following our approach to KG construction, we treat each of these passages as a distinct node within the KG. For this evaluation, we limit the search scope to a 2-hop traversal, meaning that the system can retrieve and consider information from up to two edges away from the starting node in the graph. To ensure a fair comparison with KGP-T5, we standardize the retrieval parameters by selecting the top 30 passages based on a similarity search to generate the answers.

3.1 Evaluation Metrics

We employ two evaluation metrics to assess the performance of our MD-QA system. First, we use accuracy (Acc) to measure the proportion of correctly answered questions. To evaluate accuracy, we prompt GPT-4o mini to compare each predicted answer with its corresponding ground truth. Second, we use exact match (EM) to assess the accuracy of information retrieval by calculating the proportion of facts correctly identified by the retriever compared to a set of golden facts. In particular, the EM implementation [5] introduced by Xiong et al. (2021) compares retrieved passages to their golden references token by token. However, in our experiments, we observe that passages from the validation sets do not always perfectly align with their golden counterparts due to the chunking strategy with overlaps used in the KGP-T5 experiments. This misalignment could potentially lead to an underestimation of the true EM scores. To address this issue, we opt to match passages based on cosine similarity instead. Our analyses indicate that while most exact matches exhibit similarity scores around 0.99, some pairs with scores as low as 0.9 are also considered equivalent.

3.2 Performance and Analysis

We study the MD-QA capability of our CuriousLLM agent (KGP-Mistral) employing several retrieval techniques as traversal agents for comparison, including the classic keyword-based methods TF-IDF and BM25, the encoder-based MDR, and a strong baseline, KGP-T5. Additionally, we include the "Golden" method, where the response generation LLM is provided with golden supporting passages, and the "None" method, where only the questions are given. As shown in Table 3, KGP-Mistral consistently achieves the highest performance across all benchmark datasets, with accuracy improvements of up to 9% compared to BM25 on HotpotQA. Furthermore, compared to the original KGP framework (KGP-T5), KGP-Mistral delivers consistent accuracy gains across all datasets, averaging a 3% improvement overall.

Keyword-based approaches such as TF-IDF and BM25 lack the reasoning capabilities necessary for complex MD-QA tasks. These methods primarily focus on retrieving passages that share keywords with the query or retrieved documents, without the ability to establish logical connections between different pieces of information. This limitation reduces their effectiveness in answering questions that require synthesizing information from multiple sources, as reflected in their lower accuracy scores compared to the LLM-based methods. The MDR method offers a more sophisticated approach by training a sentence encoder to bring passages used to answer questions closer in semantic space. This technique enhances the retrieval of related passages that are likely part of the evidence pair needed to answer the question. However, the LLM-based method, which builds on the MDR approach by incorporating a reasoning-driven LLM agent, shows better performance across all datasets. Furthermore, we show the impact of early traversal

[4] https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/

[5] https://github.com/facebookresearch/multihop_dense_retrieval

Method	HotpotQA		2WikiMQA		IIRC		MuSiQue	
	Acc	EM	Acc	EM	Acc	EM	Acc	EM
None	38.20	-	30.20	-	21.94	-	22.80	-
TF-IDF	67.40	45.60	42.80	47.66	28.09	28.27	28.60	<u>46.77</u>
BM25	64.80	42.60	42.20	46.84	28.72	28.27	28.60	44.51
MDR	70.60	50.12	44.20	45.38	31.45	30.96	32.80	**47.59**
KGP_T5	71.60	<u>51.77</u>	<u>46.80</u>	<u>49.55</u>	33.30	31.57	34.00	45.85
KGP_Mistral_ET (Ours)	<u>72.20</u>	51.27	46.00	48.25	<u>34.80</u>	<u>31.67</u>	<u>34.80</u>	45.85
KGP_Mistral (Ours)	**73.80**	**52.42**	**49.40**	**50.29**	**36.69**	**32.07**	**36.50**	45.95
Golden	81.40	100.00	69.80	100.00	64.57	100.00	53.80	100.00

Table 3: Performance (%) on 4 multi-document question answering (MD-QA) benchmark datasets. KGP_Mistral is our method. KGP_Mistral_ET is the version of our method with early termination. KGP_T5 is a strong baseline. None: no passages but only the question is provided. Golden: supporting facts are provided along with the question. The best and run-up scores are in **bold** and <u>underlined</u>.

termination in Section C.

In summary, our KGP-Mistral outperforms all other methods in accuracy and efficiency, establishing itself as a robust solution for multi-document question answering.

4 Follow-upQA Benchmark

4.1 Evaluation on Follow-upQA

We evaluate the fine-tuned Mistral-7B model on the Follow-upQA test set. We apply a train-validation-test split of 90%-5%-5%, resulting in $2.5K$ samples in the test set. The model is trained for $1,500$ steps, with a checkpoint saved every 300 steps. In addition, we conduct a grid search for various decoding parameters, including temperature, top p, and maximum token length.

Figure 3 presents histograms and line plots for ROUGE-1, ROUGE-L, and cosine similarity scores evaluating Mistral-7B on Follow-upQA. Most ROUGE scores concentrate around 0.4, with a range of approximately 0.3 to 0.5. The cosine similarity scores range primarily from 0.475 to 0.65, indicating varying degrees of semantic alignment. The line plots reveal the performance of Mistral-7B at different checkpoints and highlight the impact of decoding parameters on the model's performance. The models achieve optimal performance at the 600-step mark, with peak performance at step $1,200$. Specifically, the highest ROUGE-1 score is 0.494 (top_p=0.85). The best ROUGE-L score is 0.477, achieved under the same conditions. For cosine similarity, the highest score is 0.654, indicating close semantic alignment between the generated and golden questions.

Furthermore, the line graphs in Figure 3 reveal that the initial performance of the raw Mistral-7B model is the lowest across all metrics, indicating a significant improvement through training. The gradual increase in scores demonstrates the model's enhanced ability to generate more accurate follow-up questions as training progresses. The peak performances annotated in the plots are achieved using the same model configuration: Mistral-7B at step $1,200$, with a temperature of 0.6, top-p of 0.85, and a maximum token length of 50. Consequently, this model is selected for the MD-QA experiments. For a detailed analysis of Mistral-7B's performance across different training checkpoints, refer to the ablation study in Section D.

5 Conclusion

Our new CuriousLLM-enhanced KGP framework represents a significant advance in the field of MD-QA by integrating an LLM-guided prompt reformulation mechanism into the KG traversal process. By introducing the Follow-upQA dataset and a curiosity-driven LLM traversal agent, the system effectively addresses challenges like hallucination, inefficient retrieval, and the limitations of the original KGP framework. Extensive experiments show that this approach enhances MD-QA accuracy and efficiency while reducing computational overhead, making it a practical solution for real-world applications. This work paves the way for future research into optimizing LLM-guided retrieval processes by offering a scalable, robust framework balancing performance and resource efficiency.

Figure 3: Benchmark Mistral-7B for Follow-upQA. First row: distribution plots for ROUGE-1, ROUGE-L, and cosine similarity across hyper-parameters. Second row: Mistral-7B performance at different training checkpoints.

References

Garima Agrawal, Tharindu Kumarage, Zeyad Alghamdi, and Huan Liu. 2024. Can knowledge graphs reduce hallucinations in llms? : A survey. *Preprint*, arXiv:2311.07914.

Jinheon Baek, Alham Fikri Aji, and Amir Saffari. 2023. Knowledge-augmented language model prompting for zero-shot knowledge graph question answering. *Preprint*, arXiv:2306.04136.

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. *Preprint*, arXiv:2005.14165.

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.

Pádraig Cunningham and Sarah Jane Delany. 2021. k-nearest neighbour classifiers - a tutorial. *ACM Computing Surveys*, 54(6):1–25.

Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, and Andrew McCallum. 2019. Multi-step retriever-reader interaction for scalable open-domain question answering. *Preprint*, arXiv:1905.05733.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Preprint*, arXiv:2305.14314.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Preprint*, arXiv:1810.04805.

James Ferguson, Matt Gardner, Hannaneh Hajishirzi, Tushar Khot, and Pradeep Dasigi. 2020. Iirc: A dataset of incomplete information reading comprehension questions. *Preprint*, arXiv:2011.07127.

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. *Preprint*, arXiv:2310.06825.

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*.

Abhijeet Kumar, Abhishek Pandey, Rohit Gadia, and Mridul Mishra. 2020a. Building knowledge graph using pre-trained language model for learning entity-aware relationships. In *2020 IEEE International Conference on Computing, Power and Communication Technologies (GUCON)*, pages 310–315. IEEE.

Abhijeet Kumar, Abhishek Pandey, Rohit Gadia, and Mridul Mishra. 2020b. Building knowledge graph using pre-trained language model for learning entity-aware relationships. In *2020 IEEE International Conference on Computing, Power and Communication Technologies (GUCON)*, pages 310–315.

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. *Preprint*, arXiv:2011.07208.

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

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. *Preprint*, arXiv:2305.18290.

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

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

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

Corby Rosset, Chenyan Xiong, Minh Phan, Xia Song, Paul Bennett, and Saurabh Tiwary. 2021. Knowledge-aware language model pretraining. *Preprint*, arXiv:2007.00655.

Robin M. Schmidt. 2019. Recurrent neural networks (rnns): A gentle introduction and overview. *Preprint*, arXiv:1912.05911.

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. *Preprint*, arXiv:2302.13971.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Musique: Multi-hop questions via single-hop question composition. *Preprint*, arXiv:2108.00573.

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. *Preprint*, arXiv:2308.11730.

Margaret Warren, Ayman Shamma, and Patrick Hayes. 2021. Knowledge engineering with image data in real-world settings. In *Proceedings of the AAAI Spring Symposium on Combining Machine Learning and Knowledge Engineering*, 2846.

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. *Preprint*, arXiv:2402.10412.

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

Xin Xie, Ningyu Zhang, Zhoubo Li, Shumin Deng, Hui Chen, Feiyu Xiong, Mosha Chen, and Huajun Chen. 2022b. From discrimination to generation: Knowledge graph completion with generative transformer. In *Companion Proceedings of the Web Conference 2022*, WWW '22. ACM.

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. *Preprint*, arXiv:2009.12756.

Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. 2024. Hallucination is inevitable: An innate limitation of large language models. *Preprint*, arXiv:2401.11817.

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. 2023. Llm lies: Hallucinations are not bugs, but features as adversarial examples. *Preprint*, arXiv:2310.01469.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. Ernie: Enhanced language representation with informative entities. *arXiv preprint arXiv:1905.07129*.

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. 2023a. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023b. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *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. *Preprint*, arXiv:1909.08593.

A Related Work

In the field of multi-document question answering (MD-QA), there has been significant progress in developing models that can efficiently retrieve and generate relevant information. MD-QA systems face unique challenges, as they require the model to process and integrate information from multiple documents while maintaining coherence and accuracy in the generated response. In this section, we provide an overview of the existing MD-QA approaches, categorized into three main types: retrieval-based models, generative models, and hybrid models that combine the strengths of both retrieval and generation. Each of these approaches brings distinct advantages and limitations to the task of answering complex questions that require reasoning over multiple documents.

Retrieval-based Models. Current retrieval-based models, such as TF-IDF (Ramos, 2003) and BM25 (Robertson et al., 2009), employ a term-document relevance mechanism to retrieve information based on lexical similarity to the query. Although these models perform well for questions that share explicit keywords with target documents, they often struggle when the query requires a deeper semantic understanding of the context (Lan et al., 2022; Viji and Revathy, 2023; Modi et al., 2023; Cheng et al., 2022). To bridge this gap, encoder-based techniques, such as 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), leverage the power of deep learning to capture semantic information in texts. However, addressing the complexities of MD-QA presents additional challenges.

Generative Models. Recent advancements in LLMs have allowed 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 are trained in vast corpora and further enhanced through Reinforcement Learning (RL) (Ziegler et al., 2020; Rafailov et al., 2023) to effortlessly compose responses that mimic human conversations. However, the time and financial burdens associated with training, hosting, and maintaining an LLM are beyond the reach of many. In addition, LLMs are subject to issues like hallucination and knowledge cut-offs, limiting their effectiveness in the MD-QA domain.

Hybrid Models. Hybrid models represent a fusion of retrieval-based and generative approaches, equipping LLMs with a document retrieval system to provide relevant contextual information for response generation. This fusion effectively addresses the common issues faced by LLMs. Popular examples of such hybrid models include RAG and KAPING. Furthermore, (Pan et al., 2024; Agrawal et al., 2024) summarize various strategies for unifying KGs and LLMs, including KG-enhanced LLMs (Shen et al., 2020; Zhang et al., 2019; Rosset et al., 2021), LLM-augmented KGs (Zhang et al., 2020; Xie et al., 2022a,b; Kumar et al., 2020b) and synergized LLM + KGs (Zhu et al., 2023a,b; Thoppilan et al., 2022; Warren et al., 2021). These approaches significantly enhance the QA capabilities of LLMs.

B Algorithms

In this section, we present two key algorithms designed to improve the performance of our CuriousLLM agent in the MD-QA framework. Algorithm 1 focuses on leveraging a KG to traverse and retrieve relevant information given a content-based user query. This process ensures that the model accesses the most pertinent context to answer complex questions. Algorithm 2 generates follow-up questions based on the retrieved passages to iteratively refine the search, enhancing the model's ability to gather more specific information. Together, these algorithms form the core of our system, enabling efficient and accurate responses in MD-QA tasks.

C Impact of Early Traversal Termination

Our evaluation of early traversal termination using the CuriousLLM agent demonstrates substantial improvements in efficiency while maintaining competitive accuracy in MD-QA tasks. Our mistral model, fine-tuned to generate a termination signal ("NA") when it has gathered sufficient evidence, significantly reduces traversal time without compromising the quality of the final answer. This early termination feature allows the agent to stop the search process as soon as necessary information is identified, thus minimizing unnecessary computation and latency.

We compare three different agents: Mistral with early termination (Mistral-ET), the standard Mistral without early termination, and the T5 model. We collect questions that are closed early by

Algorithm 1 LLM-based KG Traversal Algorithm for Retrieving Relevant Context Given a Content-based User Query

Require: An initial query q over a set of documents $\mathcal{D}$, the constructed KG $G = \{\mathcal{V}, \mathcal{E}, \mathcal{X}\}$ over $\mathcal{D}$, the LLM graph traversal agent F_{agent}, the preset passage budget K, the TF-IDF seeding passage retriever g

Initialize seed passages $\mathcal{V}^s = g(\mathcal{V}, \mathcal{X}, q)$
Initialize the retrieved passage queue $\mathcal{P} = \{[v_i] \mid v_i \in \mathcal{V}^s\}$
Initialize the candidate neighbor queue $\mathcal{C} = [\mathcal{N}_i \mid v_i \in \mathcal{V}^s]$
Initialize the retrieved passage counter $k = \sum_{P_i \in \mathcal{P}} |P_i|$
while queue $\mathcal{P}$ **and queue** $\mathcal{C}$ **are not empty do**
  $P_i \leftarrow \mathcal{P}$.**dequeue**(), $C_i \leftarrow \mathcal{C}$.**dequeue**()
  $\mathcal{V}'_i =$ **Traversal Agent**(q, P_i, C_i)
  if $\mathcal{V}'_i = \emptyset$ **then**
    Terminate the loop
  end if
  for each $v \in \mathcal{V}'_i$ **do**
    $\mathcal{P}$.**enqueue**$(P_i \cup \{v\})$, $\mathcal{C}$.**enqueue**$(\mathcal{N}_v)$
    $k \leftarrow k + 1$
    if $k > K$ **then**
      Terminate the loop
    end if
  end for
end while
return Retrieved Passage Queue $\mathcal{P}$

Algorithm 2 CuriousLLM to Ask Follow-up Questions to Guide the Search

Require: q as initial query, P_i as list of retrieved passages, C_i as list of neighbor passages (these are inputs to Traversal Agent from **Algorithm 1**), preset top_k context budget, CuriousLLM $\mathcal{LM}$, sentence transformer Emb, similarity function f_{sim}, and ranker $\mathcal{R}$.

$q_{\text{new}} \leftarrow \text{CONCAT}(\{q\}, P_i)$
$q_{\text{follow\_up}} \leftarrow \mathcal{LM}(q_{\text{new}})$ from Eq (1)
if $q_{\text{follow\_up}} ==$ 'NA' **then**
  return $\{\}$
else
  Sim_scores $\leftarrow f_{\text{sim}}(\text{Emb}(q_{\text{follow\_up}}), \text{Emb}(C_i))$
  Candidates $\leftarrow \mathcal{R}$(Sim_scores) from Eq. (2)
  return the top_k in Candidates
end if

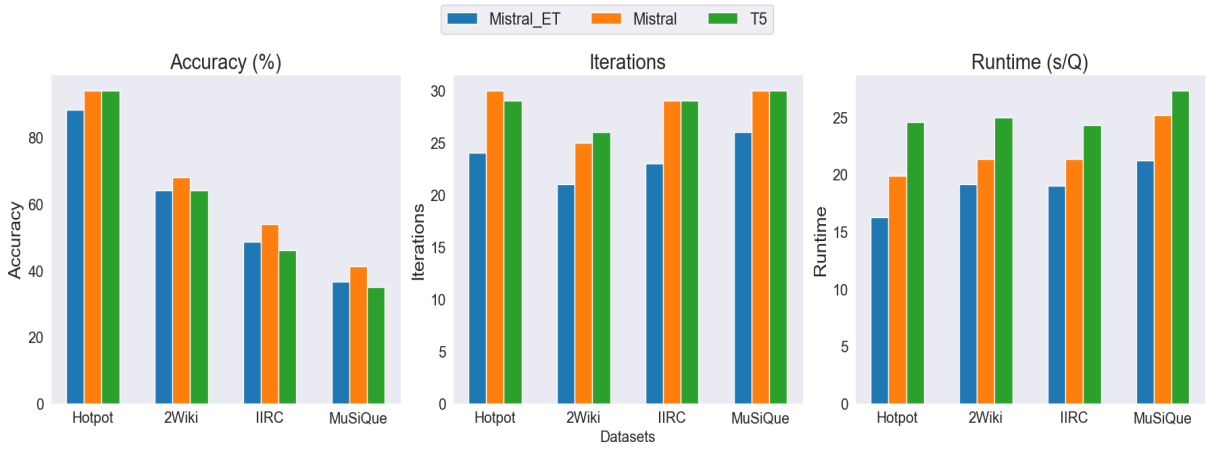

Figure 4: A comparison of MD-QA across LLM agents. Mistral_ET is Mistral agent with early traversal termination. Accuracy calculates the correct rate of the questions that are early terminated by Mistral_ET. Iterations can be interpreted as the number of nodes visited. Runtime records the average runtime in second per question.

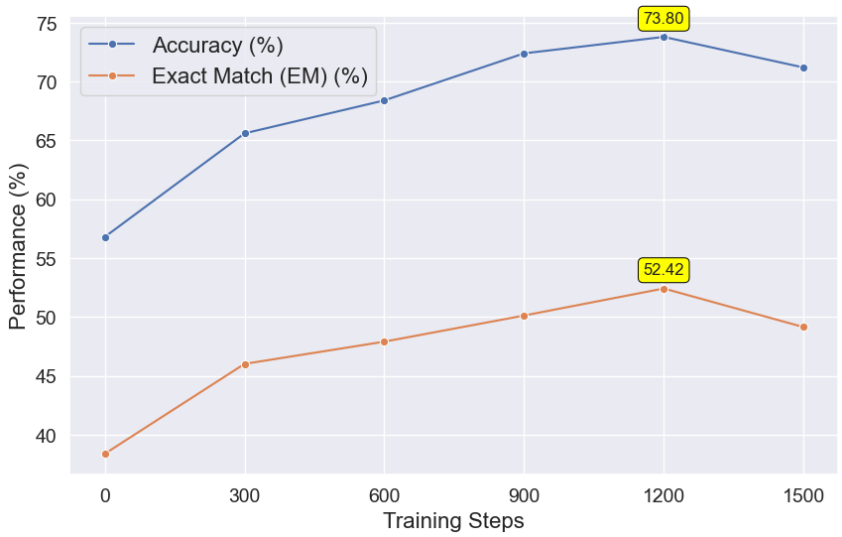

Figure 5: MD-QA performance on HotpotQA with Mistral agent at different training checkpoints.

Mistral-ET and evaluate the accuracy scores on these questions across all LLM agents. Moreover, we record the number of iterations or prompt reformulations for all questions, since fewer iterations typically result in lower latency. We also track the run-time across all questions.

The results in Table 3 indicate that the early termination capability of Mistral-ET enables it to achieve accuracy levels similar to those of the T5 model, with fewer traversal iterations and reduced runtime. In this experiment, iterations refer to the number of nodes in the KG that need to be visited to gather sufficient evidence. Specifically, the accuracy of Mistral-ET matches closely that of T5, yet Mistral-ET consistently requires fewer node visits and less time per query, as illustrated in Figure 4. Additionally, Mistral-ET shows only marginal differences in accuracy compared to its non-terminating counterpart, with the added benefit of faster execution. This efficiency gain is particularly relevant in real-world applications where quick response times are critical. Furthermore, since the experiments are conducted on a MacBook M2 Max, we anticipate even faster runtimes across all LLM agents with more powerful computing resources.

D Ablation Study

Training Step Analysis for MD-QA Optimization. We assess the MD-QA capabilities of the Mistral models at different end-to-end checkpoints on HotpotQA. In Figure 5, we observe a clear trend indicating that the MD-QA capability peaks at $1,200$ steps. At this checkpoint, the model achieves its highest accuracy and EM scores. This suggests that the model benefits significantly from training up to this point, with both accuracy and EM scores improving steadily from 0 to $1,200$ steps. However, beyond $1,200$ steps, there is a slight decline in both metrics. Specifically, at $1,500$ steps, both the accuracy and EM scores decrease, suggesting that the model may begin to overfit the training data or that additional training steps introduce noise, slightly degrading performance.

Overall, the results underscore the importance of selecting an optimal number of training steps to maximize the performance of the KG traversal agent. Training up to $1,200$ steps appears to be the most effective strategy for this particular model and dataset, balancing sufficient learning with avoiding overfitting.

E Limitations

While CuriousLLM demonstrates significant advancements in multi-document question answering (MD-QA), there are several limitations that merit discussion:

Question Scope. This system primarily focuses on addressing comparison and bridging questions, which require reasoning across multiple pieces of evidence. However, it leaves other common question types, such as what, where, and how, unexplored. Expanding the system to handle these broader types of questions would increase its applicability and robustness in diverse real-world scenarios.

Hardware and Scalability. The experiments conducted in this study used a single GPU, which limits the exploration of parallel processing and distributed computation. While the current setup validates the system's feasibility, its scalability to large-scale deployments in real-world environments will require more advanced hardware infrastructure and efficient parallelization techniques.