Multimodal Multi-Hop Question Answering Through a Conversation Between Tools and Efficiently Finetuned Large Language Models

Hossein Rajabzadeh[1,2], **Suyuchen Wang** [2,3,4],
Hyock Ju Kwon[1], **Bang Liu**[2,3,4]
[1]University of Waterloo
[2]Université de Montréal
[3]Huawei Noah's Ark Lab
[4]Mila
{hossein.rajabzadeh, hjkwon}@uwaterloo.ca,
{suyuchen.wang, bang.liu}@umontreal.ca

Abstract

We employ a tool-interacting divide-and-conquer strategy enabling large language models (LLMs) to answer complex multimodal multi-hop questions. In particular, we harness the power of large language models to divide a given multimodal multi-hop question into unimodal single-hop sub-questions to be answered by the appropriate tool from a predefined set of tools. After all corresponding tools provide the LLM with their answers, the LLM generates the next relevant unimodal single-hop question. To increase the reasoning ability of LLMs, we prompt chatGPT to generate a tool-interacting divide-and-conquer dataset. This dataset is then used to efficiently finetune the corresponding LLM. To assess the effectiveness of this approach, we conduct an evaluation on two recently introduced complex question-answering datasets. The experimental analysis demonstrate substantial improvements over existing state-of-the-art solutions, indicating the efficacy and generality of our strategy.

1 Introduction

Large language models (LLMs), such as ChatGPT OpenAI [2022], have demonstrated conspicuous reasoning abilities in few-shot settings across various tasks. However, answering multi-hop questions, which require sequential and compositional reasoning, remains a formidable challenge for LLMs Brown et al. [2020], Bommasani et al. [2021], Wei et al. [2022]. This challenge exposes serious limitations of LLMs, such as untruthfulness, false reasoning, and hallucination Bang et al. [2023], Alkaissi and McFarlane [2023], Press et al. [2022], Azamfirei et al. [2023], Kasneci et al. [2023], Ferrara [2023], Creswell and Shanahan [2022].

To enable LLMs to break down the input question into a reasoning chain, several techniques have been proposed, such as Self-Ask Press et al. [2022], Least-to-most prompting Zhou et al. [2022], ReAct Yao et al. [2022], mm-ReAct Yang et al. [2023], Vipergpt Surís et al. [2023], Chameleon Lu et al. [2023], ART Paranjape et al. [2023], and CoTWei et al. [2022]. These methods work well for single-hop questions, but they face difficulties when dealing with multimodal multi-hop (MMH) reasoning questions, where the answers of the previous sub-questions affect the formulation of the subsequent ones. Besides, ReAct and mm-ReAct do not necessarily break down a given MMH question into a chain of easy to hard unimodal single-hop (USH) sub-questions which can further results in minimum number of calls to the tools.

37th Conference on Neural Information Processing Systems - ENLSP (NeurIPS 2023).

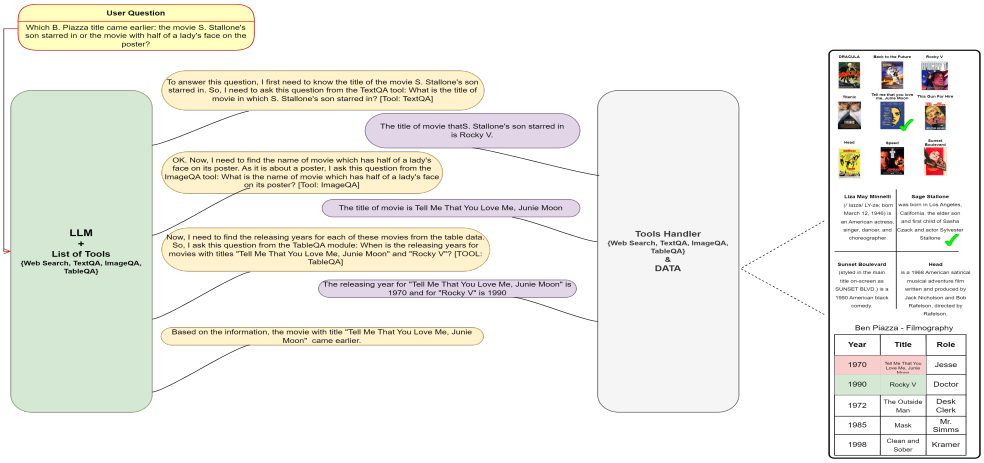

Figure 1: An illustration of the interactive strategy between LLMs and Tools to answer a MMH question. The LLM first divides the question into a USM sub-question and determines its associated tool. The sub-question is then answered by the associated tool, replying the answer back to LLM. The LLM then asks its next USH question based on the tool's reply.

A possible explanation for the poor performance of LLMs on MMH reasoning tasks is their limitation to access the relevant information. A potential solution to this problem is to equip LLMs with a set of predefined tools that can help target and process specialized information. This approach can include adding information retrieval and web search capabilities to LLMs Xu et al. [2023], Pereira et al. [2023], Peng et al. [2023] or providing LLMs with more advanced tools Paranjape et al. [2023], Schick et al. [2023], Zhang et al. [2023], Wu et al. [2023]. However, adding tools to LLMs does not guarantee a successful MMH question-answering (QA) solution, because there are still challenges related to complex reasoning, modality variations, and the accumulation of errors Lu et al. [2023], Yao et al. [2022]. To improve the MMH QA model, it is essential to enable LLMs to perform a dynamic reasoning chain that interacts with tools and integrates their outputs to generate the final answers efficiently and effectively. Another important factor contributing in the success of LLMs in reasoning tasks is the model size, where the higher model's capacity leads to a more powerful reasoning ability Stolfo et al. [2022], Shridhar et al. [2022], Magister et al. [2022].

In this work, we mainly focus on the problem of MMH question answering using an interactive strategy between LLMs and a set of tools. Figure 1 illustrates our proposed strategy with an example. First, the LLM receives the input MMH question and simplifies it into a unimodal single-hop (USH) sub-question while specifying the tool's name required to answer that sub-question. A tools handler[1] then receives this USH sub-question, calls the associated tool, and returns the answer. The LLM uses the answer to generate the next USH sub-question. The interaction between the LLM and tools continues until the final answer is found.

We evaluate this strategy on two complex MMH datasets and benchmark the performance of three LLMs on these datasets. Therefore, the primary contributions of this study can be summarized as follows: 1) We propose an interactive strategy that enables LLMs to communicate with tools and generate a sequence of sub-questions through a divide-and-conquer approach, allowing LLMs to decompose the MMH questions into USH sub-questions and answer the original question. 2) We generate a divide-and-conquer dataset to efficiently finetune smaller-size LLMs, thereby enhancing their reasoning capabilities on MMH QA tasks. 3) We evaluate our strategy on two recent MMH QA datasets and compare the results using LLMs of different sizes.

2 Tool-interacting Divide-and-Conquer Prompting for MMH QA

Let us consider a scenario where a question requires the use of multiple tools to be answered, i.e. multimodal reasoning. We also assume that the order of tool invocation matters, such that the output

[1]We simply postprocess the LLM's output to extract the sub-question and its associated tool.

of tool A may serve as the input for tool B, exemplifying multi-hop reasoning. Furthermore, we define tools as USH QA models that can answer a USH question based on the provided data.

To answer MMH questions, LLMs need to extract an initial USH sub-question that can be addressed with the corresponding tool. The feature of unimodality ensures the LLM calls the correct tool for a sub-question. Moreover, the simplicity feature increases the chance that the LLM receives the correct answer from the tools. Ultimately, the interactive fusion of these two attributes provides a divide-and-conquer strategy, successfully should lead the LLM to the final answer.

LLM as the Divider: Assuming the example in Figure 1, the LLM first divides the MMH original question into a USH sub-question by asking for the movie's title from the relevant tools. After receiving the tool's reply, it asks the next sub-question, forming another piece of information required for building the final answer. The LLM continues such division behaviour till it gets all necessary pieces of information to answer the original question. **Tools as the Conqueror**: Considering the example in Figure 1, each time that the LLM asks a sub-question, the corresponding tool is invoked to find the requested answer. As the sub-question is a USH question, it is highly likely that the tool successfully obtains the answer. This behaviour introduces tools as a powerful conqueror for the divider. Here, we assume that tools have access only to their associated data modalities, and the answer is given in the corresponding modality.

To enhance the reasoning and tool-interacting capabilities of typical-sized LLMs, such as 7, 13, 30, and 40 billions [2], we efficiently finetune LLMs of different sizes for one epoch using QLoRA Dettmers et al. [2023] on a tool-interacting divide-and-conquer dataset. This one epoch of finetuning encourage the corresponding LLM to follow the divide-and-conquer strategy while interacting with the required tools.

Generating a Tool-interacting Divide-and-Conquer Dataset: To build a dataset that answers MMH questions through the proposed tool-interacting divide-and-conquer strategy, we prompted ChatGPT by providing it with manually created few-shot examples. We use eight-shot examples, consisting of three main modality of Text, Table, and image data. Additionally, four different tools are considered to interact with the associated LLM. For each benchmark, we prompted ChatGPT and pass a random subset[3] of the corresponding training set and proceed to generate a tool-interacting divide-and-conquer dataset.

3 Experiments

This section evaluates the effectiveness of our tool-interacting divide-and-conquer strategy for several large language models on two recent MMH QA benchmarks. The comparative analysis involves different combinations of Language Models with varying sizes and strategies. The results are compared in terms of exact matching (EM), $F-$scores (F_1), and average number of Tool Calls, i.e. the average number of times that the LLM calls a tool to answer a question. The subsequent sections provide more details about the benchmarks, LLMs, and the obtained results.

We employ two MMH QA benchmarks for evaluation and comparison: MultiModalQA Talmor et al. [2021] and MMCoQA Li et al. [2022]. These benchmarks provide different data modalities while offering complex multi-hop questions. Specifically, MultiModalQA contains 29,918 question-answer pairs and encompasses three distinct modalities, namely text data, table data, and image data. Notably, each question in this dataset requires the integration of varying combinations of text, table, and image inputs for accurate answering. Furthermore, MMCoQA is a multimodal conversational QA benchmark, incorporating four modalities: text, table, image, and conversation. This benchmark comprises 1,179 conversations, with an average of 4.88 question-answer pairs per conversation.

To assess the effectiveness of our tool-interacting divide-and-conquer strategy, we employ five distinct LLMs for evaluation purposes: StableLM-7b[4], Pathia-12b[5], LLaMA-13b [6], Falcon-40b [7], and ChatGPT. Except for ChatGPT, other LLMs are finetuned for one epoch on the associated

[2]For simplicity, we denote various Language Model sizes as 7b, 13b, 30b, and 40b.

[3]The size of training subset is 2k for each benchmark.

[4]`https://huggingface.co/OpenAssistant/stablelm-7b-sft-v7-epoch-3`

[5]`https://huggingface.co/OpenAssistant/oasst-sft-4-pythia-12b-epoch-3.5`

[6]`https://huggingface.co/decapoda-research/llama-13b-hf`

[7]`https://huggingface.co/OpenAssistant/falcon-40b-sft-top1-560`

Table 1: Left: validation results on the MultiModalQA benchmark. Right: test results on the MMCoQA benchmark. All methods are presented with an identical set of tools. "NA" denotes data that is not available. The maximum number of Tool Calls per each question is set to 12. "*" means the corresponding LLM is finetuned using QLoRA for one epoch through its associated strategy.

LLM	Size	Strategy	EM	F_1	Average Tool Calls
StableLM	7b	*ToolsAnswer*	0.0	9.36	4
Pathia	12b	*ToolsAnswer*	0.3	15.24	4
LLaMA	13b	*ToolsAnswer*	1.21	17.32	4
Falcon	40b	*ToolsAnswer*	14.12	33.92	4
ChatGPT	NA	*ToolsAnswer*	15.45	46.42	4
StableLM*	7b	*mm-ReAct*	1.32	14.50	11.52
Pathia*	12b	*mm-ReAct*	6.51	23.18	10.60
LLaMA*	13b	*mm-ReAct*	9.87	27.45	10.53
Falcon*	40b	*mm-ReAct*	18.94	45.34	8.24
ChatGPT	NA	*mm-ReAct*	21.30	52.16	6.22
StableLM*	7b	*Ours*	18.50	25.12	9.69
Pathia*	12b	*Ours*	21.32	31.14	8.14
LLaMA*	13b	*Ours*	23.14	35.21	8.75
Falcon*	40b	*Ours*	41.18	56.74	6.34
ChatGPT	NA	*Ours*	43.71	61.03	5.07
Human	-	-	86.2	91.2	-

LLM	Size	Strategy	EM	F_1	Average Tool Calls
StableLM	7b	*ToolsAnswer*	0.0	8.35	4
Pathia	12b	*ToolsAnswer*	0.0	12.41	4
LLaMA	13b	*ToolsAnswer*	0.0	13.95	4
Falcon	40b	*ToolsAnswer*	3.45	22.15	4
ChatGPT	NA	*ToolsAnswer*	8.91	46.10	4
StableLM*	7b	*mm-ReAct*	2.24	17.42	11.56
Pathia*	12b	*mm-ReAct*	6.15	18.51	10.35
LLaMA*	13b	*mm-ReAct*	7.84	21.32	10.41
Falcon*	40b	*mm-ReAct*	18.37	41.08	8.96
ChatGPT	NA	*mm-ReAct*	41.33	52.17	6.35
StableLM*	7b	*Ours*	7.11	16.36	10.22
Pathia*	12b	*Ours*	10.76	24.21	9.34
LLaMA*	13b	*Ours*	11.27	26.39	6.54
Falcon*	40b	*Ours*	38.91	56.51	4.96
ChatGPT	NA	*Ours*	47.05	58.82	3.80
Human	-	-	NA	NA	-

tool-interacting divide-and-conquer datasets [8]. To assess the reasoning capability of the proposed strategy, we compare it with two other strategies. In the first strategy, the original question is independently processed by each tool, and the returned answers are considered as a prompt to the LLM. Subsequently, the LLM is asked to answer the original question given the prompt. We call this setting *ToolsAnswer*. The second strategy is mm-ReAct, where we follow the vanilla mm-ReAct presented with the tools descriptions. Additionally, we apply our proposed strategy to each LLM, denoted as *Ours* for reference.

Each data modality needs a specific tool to handle that particular modality. As there are three main modalities, we employ three generic tools: 1) **TextQA** uses Instructor-large[9] Su et al. [2022] which is a text embedding model that has undergone fine-tuning specifically for instructional purposes. 2) **TableQA** employs TAPAS[10] Herzig et al. [2020], utilizing BERT's structure for covering table-based QA. 3) **ImageQA** is based on BLIP-2[11] Li et al. [2023], that receives an image and a question as inputs and returns the corresponding text answer as the output. 4) **Web Search** tool is invoked when the remaining tools fail to provide informative answers. In such case, LLM is allowed to do a web search request, retrieving relevant information from the internet. The evaluation of different methods over MultiModalQA and MMCoQA datasets are reported in Table 1. In particular, our strategy (labeled as "Ours") consistently outperforms other strategies across LLMs of different sizes, as indicated by higher Exact Match (EM) scores and F_1 scores. Additionally, the "Average Tool Calls" column demonstrates that our strategy maintains a relatively low number of tool calls, indicating efficiency in resource utilization [12].

4 Conclusion

This study presents a tool-interacting strategy, leveraging a divide-and-conquer interaction between large language models and a set of tools to effectively answer multimodal multi-hop questions. Our strategy facilitates the division of MMH questions into USH sub-questions, allowing LLMs to interact with a predefined set of tools for obtaining intermediate answers. We assessed the performance of different-sized LLMs in three different reasoning strategies. The obtained results demonstrate the effectiveness of our strategy. For the possible future directions, we will include exploring inter-tool communication, handling non-predefined modalities, and improving the performance of smaller LLMs for MMH QA tasks.

[8] In the case of mm-ReAct, the tool-interacting dataset is generated in accordance to mm-ReAct strategy

[9] `https://huggingface.co/hkunlp/instructor-large`

[10] `https://huggingface.co/google/tapas-base-finetuned-wtq`

[11] `https://huggingface.co/docs/transformers/model_doc/blip-2`

[12] In the case of ToolsAnswer, the Average Tool Calls is always equal to 4, as this strategy calls each tool only once.

References

OpenAI. Chatgpt. *OpenAI Blog*, 1(8):9, 2022.

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

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*, 2022.

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

Hussam Alkaissi and Samy I McFarlane. Artificial hallucinations in chatgpt: implications in scientific writing. *Cureus*, 15(2), 2023.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. *arXiv preprint arXiv:2210.03350*, 2022.

Razvan Azamfirei, Sapna R Kudchadkar, and James Fackler. Large language models and the perils of their hallucinations. *Critical Care*, 27(1):1–2, 2023.

Enkelejda Kasneci, Kathrin Seßler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, et al. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and Individual Differences*, 103:102274, 2023.

Emilio Ferrara. Should chatgpt be biased? challenges and risks of bias in large language models. *arXiv preprint arXiv:2304.03738*, 2023.

Antonia Creswell and Murray Shanahan. Faithful reasoning using large language models. *arXiv preprint arXiv:2208.14271*, 2022.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Olivier Bousquet, Quoc Le, and Ed Chi. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*, 2022.

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

Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. Mm-react: Prompting chatgpt for multimodal reasoning and action. *arXiv preprint arXiv:2303.11381*, 2023.

Dídac Surís, Sachit Menon, and Carl Vondrick. Vipergpt: Visual inference via python execution for reasoning. *arXiv preprint arXiv:2303.08128*, 2023.

Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. Chameleon: Plug-and-play compositional reasoning with large language models. *arXiv preprint arXiv:2304.09842*, 2023.

Bhargavi Paranjape, Scott Lundberg, Sameer Singh, Hannaneh Hajishirzi, Luke Zettlemoyer, and Marco Tulio Ribeiro. Art: Automatic multi-step reasoning and tool-use for large language models. *arXiv preprint arXiv:2303.09014*, 2023.

Shicheng Xu, Liang Pang, Huawei Shen, Xueqi Cheng, and Tat-seng Chua. Search-in-the-chain: Towards the accurate, credible and traceable content generation for complex knowledge-intensive tasks. *arXiv preprint arXiv:2304.14732*, 2023.

Jayr Pereira, Robson Fidalgo, Roberto Lotufo, and Rodrigo Nogueira. Visconde: Multi-document qa with gpt-3 and neural reranking. In *Advances in Information Retrieval: 45th European Conference on Information Retrieval, ECIR 2023, Dublin, Ireland, April 2–6, 2023, Proceedings, Part II*, pages 534–543. Springer, 2023.

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, et al. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *arXiv preprint arXiv:2302.12813*, 2023.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *arXiv preprint arXiv:2302.04761*, 2023.

Beichen Zhang, Kun Zhou, Xilin Wei, Wayne Xin Zhao, Jing Sha, Shijin Wang, and Ji-Rong Wen. Evaluating and improving tool-augmented computation-intensive math reasoning, 2023.

Chenfei Wu, Shengming Yin, Weizhen Qi, Xiaodong Wang, Zecheng Tang, and Nan Duan. Visual chatgpt: Talking, drawing and editing with visual foundation models, 2023.

Alessandro Stolfo, Zhijing Jin, Kumar Shridhar, Bernhard Schölkopf, and Mrinmaya Sachan. A causal framework to quantify the robustness of mathematical reasoning with language models. *arXiv preprint arXiv:2210.12023*, 2022.

Kumar Shridhar, Alessandro Stolfo, and Mrinmaya Sachan. Distilling multi-step reasoning capabilities of large language models into smaller models via semantic decompositions. *arXiv preprint arXiv:2212.00193*, 2022.

Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. Teaching small language models to reason. *arXiv preprint arXiv:2212.08410*, 2022.

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

Alon Talmor, Ori Yoran, Amnon Catav, Dan Lahav, Yizhong Wang, Akari Asai, Gabriel Ilharco, Hannaneh Hajishirzi, and Jonathan Berant. Multimodalqa: Complex question answering over text, tables and images. *arXiv preprint arXiv:2104.06039*, 2021.

Yongqi Li, Wenjie Li, and Liqiang Nie. Mmcoqa: Conversational question answering over text, tables, and images. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4220–4231, 2022.

Hongjin Su, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A Smith, Luke Zettlemoyer, Tao Yu, et al. One embedder, any task: Instruction-finetuned text embeddings. *arXiv preprint arXiv:2212.09741*, 2022.

Jonathan Herzig, Paweł Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Martin Eisenschlos. Tapas: Weakly supervised table parsing via pre-training. *arXiv preprint arXiv:2004.02349*, 2020.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023.