Solution for Meta KDD Cup' 25: A Comprehensive Three-Step Framework for Vision Question Answering

Zijian Zhang*
MeiTuan
Shanghai, China
zhangzijian14@meituan.com

Xiaocheng Zhang*
MeiTuan
BeiJing, China
zhangxiaocheng@meituan.com

Yang Zhou*
MeiTuan
ShangHai, China
zhouyang96@meituan.com

Zhimin Lin*
MeiTuan
BeiJing, China
linzhimin@meituan.com

Peng Yan†
MeiTuan
BeiJing, China
yanpeng04@meituan.com

Abstract

Vision Large Language Models (VLLMs) have improved multi-modal understanding and visual question answering (VQA), but still suffer from hallucinated answers. Multi-modal Retrieval-Augmented Generation (RAG) helps address these issues by incorporating external information, yet challenges remain in visual context comprehension, multi-source retrieval, and multi-turn interactions. To address these challenges, Meta constructed the CRAG-MM benchmark and launched the CRAG-MM Challenge at KDD Cup 2025, which consists of three tasks. This paper describes the solutions of all tasks in Meta KDD Cup'25 from **BlackPearl** team. We use a single model for each task, with key methods including data augmentation, RAG, reranking, and multi-task fine-tuning. Our solution achieve automatic evaluation rankings of 3rd, 3rd, and 1st on the three tasks, and win second place in Task3 after human evaluation.

CCS Concepts

• **Computing methodologies** → **Natural language generation**.

Keywords

Vision Large Language Models, RAG

1 Introduction

Vision Large Language Models (VLLMs) have made significant progress in enabling multi-modal understanding and visual question answering (VQA). However, they still struggle with generating hallucinated answers and handling complex or long-tail queries that require abilities such as recognition, OCR, and knowledge integration[7, 11]. The Retrieval-Augmented Generation (RAG) paradigm extends to multi-modal (MM) input and shows promise in overcoming VLLM's knowledge limitations. Given an image and a question, an MM-RAG system generates a search query, retrieves relevant external information, and provides grounded answers[2]. Despite its potential, MM-RAG still faces challenges in understanding visual context, retrieving relevant information, integrating multi-source data, and supporting multi-turn conversations.

To address these issues, Meta introduces CRAG-MM with the aim of reliably evaluating MM-RAG QA systems[8]. CRAG-MM is a visual question-answering benchmark focused on factual questions, featuring 5,000 diverse images—including 3,000 egocentric photos from RayBan Meta smart glasses—across 13 domains. It includes four types of questions, from simple image-based queries to complex ones requiring multi-source retrieval and reasoning, as well as both single-turn and multi-turn conversations for comprehensive evaluation of MM-RAG solutions.

Based on this benchmark, the CRAG-MM Challenge is the sole event in the 2025 KDD Cup, aiming to encourage the development and evaluation of advanced MM-RAG systems. Meta designed three competition tasks. Task1 and Task2 contain single-turn questions, where the former provides image-KG-based retrieval, and the latter additionally introduces web retrieval; Task3 focuses on multi-turn conversations:

(1) **Task1: Single-source Augmentation.** Only an image-based mock KG is provided to test the basic answer generation capability of MM-RAG systems.
(2) **Task2: Multi-source Augmentation.** An additional web search mock API is provided to test how well the MM-RAG system synthesizes information from different sources.
(3) **Task3: Multi-turn QA.** To test context understanding for smooth multi-turn conversations.

The solution of each team must be submitted for inference online, with each generated answer having to be produced in 30 seconds and restricted to the use of the Llama model.

We form the BlackPearl team and participate in all three tasks, achieving automatic evaluation rankings of 3rd, 3rd, and 1st, respectively. After human evaluation, we secure 2nd place in Task3. This paper provides a detailed description of our solutions for all three tasks, with major improvements including data augmentation, RAG, reranking, and multi-task fine-tuning. Our code and data are available on github [1].

2 Solution to Task1

This section presents our solution for Task1, including key components such as image retrieval, data augmentation, and model fine-tuning. Some of these techniques are also applied to Task2 and Task3. Figure 1 illustrates our inference framework for Task1.

*All authors contributed equally to this research.
†Corresponding Author

[1] https://github.com/BlackPearl-Lab/KddCup-2025-CRAG-MM-Winning-Solution

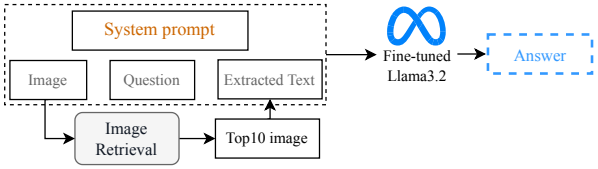

Figure 1: Illustration of Task1 framework.

2.1 Image Retrieval

We utilize the official retrieval tool for image retrieval, representing the image as a vector, and retrieving the top 10 images most similar to the current input image. Instead of directly adding the retrieved images to the model input, we first extract the textual information from the structured data associated with these images. This textual information is then incorporated into the model input as additional context, thereby enhancing the model's response quality.

2.2 Data augmentation

This subsection introduces our data augmentation(DA) approach for Task1, with the overall workflow illustrated in Figure 2. For each sample, we first use the image retrieval module described in the previous subsection to obtain relevant information. Then, the original question, image, and retrieved information are fed into Llama3.2 for inference, resulting in an initial answer from the model. Next, we use GPT-4o mini to verify the answer against the ground truth label. If the answer is identified as a hallucination, the label for this sample is set to "I don't know"; otherwise, the sample is retained for the next stage. Specifically, we use GPT-4o mini to generate n similar labels (n=10) for the verified label. All generated labels are then re-verified to filter out hallucinated labels. The remaining m labels, together with the original question and image, are used to construct m additional samples, which are utilized to enhance the training process.

2.3 Model Fine-Tuning and Inference

Base Model. We follow the contest instructions to use the LLama series LLM [2]. Considering the limited running time, we use the Llama-3.2-11B-Vision-Instruct model as the base model.

Fine-Tuning Data. We randomly split the original Task1 dataset into training and validation sets at an 8:2 ratio based on images. The data augmentation methods described in Sec.2.2 are applied to the training set to construct the fine-tuning data.

Fine-Tuning. Due to the high requirements for memory efficiency and training speed during the competition, parameter-efficient fine-tuning methods are more suitable. Therefore, we adopted LoRA (Low-Rank Adaptation)[3], which enables efficient fine-tuning with only a small number of additional parameters, thus saving resources. More details on the fine-tuning parameters can be found in Sec.4.1.

Inference. To meet the inference time constraints of the competition, we utilized vLLM[5] for model inference. vLLM offers significantly higher throughput and lower latency compared to standard inference frameworks, making it well-suited for efficient large-scale deployment. The overall inference process is shown in Figure 1. First, we retrieve the top 10 most similar images and extract only their associated textual information. This textual information is then combined with the original input and fed into the model for inference.

[2]https://llama.meta.com/

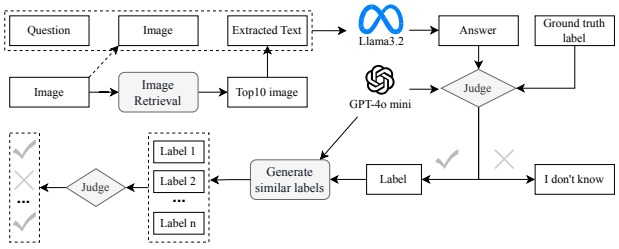

Figure 2: Data augmentation method for Task1.

3 Solution to Task2 and Task3

This section presents our solution for Task2 and Task3. The framework is illustrated in Figure 3. The key components of our solution include web retrieval, reranking, and multi-task fine-tuning.

3.1 Web retrieval

Step1 in Figure 3 illustrates our web retrieval process. First, we concatenate the retrieval prompt, image, question, and historical questions to construct the input for the Llama model. This input is then fed into the model for inference to generate a query for retrieval. This process is repeated several times with randomization to obtain multiple queries. For each query, we use the official tool to perform a retrieval and obtain multiple groups of results. Finally, we retain the group with the largest number of results as the final retrieval result.

3.2 Reranking

Due to limitations on input length and online inference time, increasing the proportion of high-quality results in the retrieval output becomes crucial. To address this, we designed a reranking module to prioritize high-quality results as much as possible. Notably, the winning solution of the 2024 KDD Cup also adopted a reranking approach[9]. To better accommodate the data formats of other tasks and facilitate more convenient training, we did not adopt the top-ranking solutions from related Kaggle competitions[4, 6]. We developed a listwise reranking method, as illustrated in Step2 of Figure 3. Specifically, we concatenate the reranking prompt, question, image, and the retrieval results obtained from Step1 as the model input, where each retrieval result is labeled with identifiers such as '1', '2', and '3'. This input is then fed into the model for inference. The model should output the most relevant info number list in the format $[x, xx, xxx, ...]$. If there are no relevant items, output $[]$.

3.3 Multi-Task Fine-Tuning and Inference

Since vLLM does not support loading MllLama's LoRA weights, we adopt multi-task fine-tuning to enable the unified model to adapt to various task formats. The prompts corresponding to each task are provided in Appendix A.

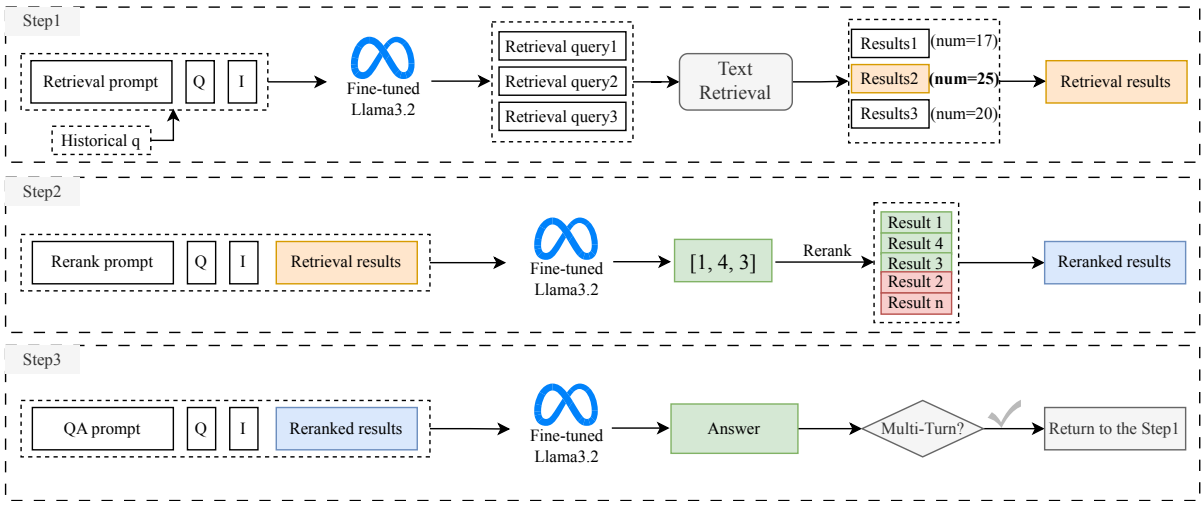

Figure 3: Illustration of Task2 and Task3 framework. The entire process consists of three steps. 'Q' and 'I' represent Question and Image, respectively.

Base Model. As in Task 1, we choose Llama-3.2-11B-Vision-Instruct as the base model.

Fine-Tuning Data. We randomly split the original Task2 and Task3 datasets into training and validation sets at an 8:2 ratio based on images. Each multi-turn dialogue session is divided into multiple samples. All fine-tuning data consists of three parts: retrieval query generation task data, re-ranking task data, and question answering task data.

For the **retrieval query generation task data**, we use the retrieval prompt combined with the historical questions (for multi-turn scenarios), the current question, and the image to construct the input, which is then fed into the original Llama model to generate the query. This generated query serves as the label required for fine-tuning in this task.

For the **reranking task data**, We take the question, image, and one of the retrieval results as input and feed them into GPT-4o. GPT-4o then determines whether the retrieval result is helpful for answering the question and outputs either True or False. The detailed prompt is shown in Appendix A.4.

For the **question answering task data**, similar to Task1, we use the QA prompt combined with the question, image, historical questions, and the retrieved results as input to Llama for answer generation. The maximum number of retrieved items per sample is set to 30. **Refusal Data Construction**: We use GPT-4o mini to determine the consistency between the ground truth answer and the predicted answer. If they are inconsistent, the fine-tuning label for that sample is changed to "I don't know." If they are consistent, we use the model-generated answer as the fine-tuning label instead of the ground truth. This approach allows the model to focus more on learning the task pattern itself rather than the more challenging transfer of specific text styles. In the final training data for our Task2 and Task3, the proportion of "I don't know." responses is 60% and 40%, respectively.

Fine-Tuning. Similar to Task1, we use LoRA for model fine-tuning. Detailed parameter settings can be found in Section 4.1. The loss function for all training tasks is cross-entropy loss.

Inference. Figure 3 illustrates the overall inference process, where a unified model is used to perform multiple tasks. For each sample, multiple diverse queries are first generated for retrieval. The retrieval tool is then called with these queries to obtain multiple groups of results, and the group with the largest number of results is retained (up to 30 results). This group is further reranked, and up to the top 10 results are kept. These results are then used as additional information and fed into the model for inference to obtain the answer. For Task2, the process ends here; for Task3, the workflow proceeds to the next round of answering, returning to Step1 to repeat the process. The inference acceleration framework also utilizes vLLM.

4 Experiments

In this section, we present our main results and ablation studies for some crucial components.

4.1 Experiment Settings.

Metrics This competition adopts exactly the same metrics and methods used in the CRAG[10] competition to assess the performance of MM-RAG systems. For each question in the evaluation set, the answer is scored as:

- *Perfect* (fully correct) → Score: 1.0
- *Acceptable* (useful but with minor non-harmful errors) → Score: 0.5
- *Missing* (e.g., "I don't know", "I'm sorry I can't find …") → Score: 0.0
- *Incorrect* (wrong or irrelevant answer) → Score: -1.0
- *Truthfulness Score*: The average score across all examples in the evaluation set for a given MM-RAG system.

Table 1: The results of different evaluation metrics for the top 6 teams on the Task 1 leaderboard. M, H, and A are abbreviations for Accuracy, Hallucination, and Missing, respectively.

Teams	Score	M	H	A
Dianping-Trust-Safety	0.073	0.711	0.108	0.181
db3	0.053	0.801	0.073	0.126
BlackPearl(Our)	0.043	0.897	0.030	0.073
y3h2	0.036	0.860	0.052	0.088
USTGZ-KIMI	0.029	0.940	0.015	0.044
Team_NVIDIA	0.026	0.920	0.027	0.053
	.....			

For multi-turn conversations, the evaluation is terminated if two consecutive answers are incorrect, and all remaining turns in the conversation are assigned a score of zero[1]. The final result is the average score across all multi-turn conversations.

Parameter Settings Our implementations are based on Pytorch. For Task1, the number of training epochs and the learning rate are set to 2 and 5e-5, respectively. For Task2 and Task3, the number of training epochs and the learning rate are set to 10 and 5e-6, respectively. The rank, alpha, and dropout parameters of LoRA are set to 64, 128, and 0.05, respectively. The warmup ratio is set to 0.03. For all tasks, during the inference phase, the maximum input length is set to 8192, and the maximum output length is set to 75. The temperature in vLLM is set to 0.0 for all cases, except when sampling retrieval queries, where it is set to 0.8. The maximum number of retrieval results for Task 2 and Task 3 is set to 30. We used the GPT-4o mini model as our LLM judge. The specific judge prompt is provided in the Appendix A.7. Fine-tuning is performed on $8 \times A100$ GPUs.

4.2 Overall Performance

Tables 1, 2, and 3 show the leaderboard results of our solution for each task, all evaluated automatically. We designed a dedicated solution for Task1, achieving a score of 0.043 and ranking third on the leaderboard. The overall framework for Task2 and Task3 is the same, with the difference being that the input for Task3 includes historical information. Our solution achieved scores of 0.107 and 0.175 for Task2 and Task3, ranking third and first, respectively.

Since the automatic evaluation of VLLMs can be somewhat uncertain, the organizers conducted a human evaluation for the top 10 teams, correcting test samples that were judged incorrect by the automatic evaluation but actually should be considered correct. As a result, the scores from human evaluation are generally higher than those from automatic evaluation. Because our solution has a relatively low hallucination rate, the improvement from human evaluation was smaller compared to other teams. Therefore, our final ranking was slightly lower than the automatic leaderboard. Table 4 presents the final scores and rankings after human evaluation, where our solution ranked second in Task3 with a score of 30.9%.

Table 2: The results of different evaluation metrics for the top 6 teams on the Task 2 leaderboard.

Teams	Score	M	H	A
db3	0.124	0.688	0.094	0.218
Dianping-Trust-Safety	0.119	0.625	0.128	0.247
BlackPearl(Our)	0.107	0.723	0.085	0.192
Team_NVIDIA	0.075	0.774	0.075	0.151
AcroYAMALEX	0.057	0.665	0.139	0.196
zmf	0.051	0.789	0.080	0.131
	.....			

Table 3: The results of different evaluation metrics for the top 6 teams on the Task 3 leaderboard.

Teams	Score	M	H	A
BlackPearl(Our)	0.175	0.638	0.094	0.269
db3	0.172	0.533	0.147	0.319
Dianping-Trust-Safety	0.121	0.706	0.086	0.208
Team_NVIDIA	0.119	0.713	0.084	0.203
y3h2	0.104	0.827	0.035	0.138
AcroYAMALEX	0.100	0.679	0.111	0.211
	.....			

Table 4: Final evaluation process and team scores

Task	Team	Score
Task1	Dianping-Trust-Safety	12.8
	db3	8.4
	cruise	6.7
Task2	Team_NVIDIA	23.3
	db3	22.1
	AcroYAMALEX	21.4
Task3	db3	36.8
	BlackPearl(Our)	30.9
	Dianping-Trust-Safety	29.7

4.3 Representative Experimental Results

We selected several representative experimental results to more clearly demonstrate the effectiveness of our approach.

Table 5 presents some experimental results for Task1. The fine-tuned model without retrieval achieved a score of only 0.01. After adding image retrieval, the score increased to 0.02. With data augmentation to enrich the fine-tuning data, the score further improved to 0.043.

Table 6 shows some experimental results for Task3 as an example. The score of the model fine-tuned with original RAG data was 0.07.

Table 5: Representative experimental results from the Task1 leaderboard.

Methods	Score(%)
Fine-tuning	0.01
Fine-tuning(*+RAG*)	0.02
Fine-tuning(*+RAG, +DA*)	0.043

Table 6: Representative experimental results from the Task3 leaderboard.

Methods	Score(%)
Fine-tuning(*RAG*)	0.0700
+Refusal Data Construction	0.1322
+Multi-query Retrieval	0.1471
+Reranking	0.1505
+Reducing the Proportion of Refusal Data	0.1755

After adding refusal data, the score increased to 0.1322. We then sampled multiple queries for retrieval to expand the retrieval results, raising the score to 0.1471. Incorporating reranking to increase the proportion of high-quality data in the input further improved the score to 0.1505. Finally, slightly reducing the proportion of refusal data in the training set (from 60% to 40%) led to a score of 0.1755, ranking first in the automatic evaluation. These methods significantly improved the scores, demonstrating their effectiveness.

5 Conclusion

The Meta CRAG-MM Challenge is the first MM-RAG competition for the KDD Cup and serves as an important driver for the development of VLLMs and VQA. We have presented our approaches to all three tasks in the contest. Due to differences in retrieval sources, we developed distinct solutions for Task 1 and Task 2/3, each with its own focus. In Task 1, we proposed a novel data augmentation strategy, while in Task 2 and Task 3, we adopted diversified retrieval, re-ranking, and multi-task fine-tuning to enhance performance. As a result, our solution achieved automatic evaluation rankings of 3rd, 3rd, and 1st in the three tasks, and won second place in Task 3 after human evaluation.

References

[1] Ge Bai, Jie Liu, Xingyuan Bu, Yancheng He, Jiaheng Liu, Zhanhui Zhou, Zhuoran Lin, Wenbo Su, Tiezheng Ge, Bo Zheng, et al. 2024. Mt-bench-101: A fine-grained benchmark for evaluating large language models in multi-turn dialogues. *arXiv preprint arXiv:2402.14762* (2024).

[2] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997* 2, 1 (2023).

[3] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR* 1, 2 (2022), 3.

[4] Jules King, L Burleigh, Simon Woodhead, Panagiota Kon, Perpetual Baffour, Scott Crossley, Walter Reade, and Maggie Demkin. 2024. Eedi - Mining Misconceptions in Mathematics. https://kaggle.com/competitions/eedi-mining-misconceptions-in-mathematics. Kaggle.

[5] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient Memory Management for Large Language Model Serving with PagedAttention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

[6] Wei lin Chiang, Evan Frick, Lisa Dunlap, Anastasios Angelopoulos, Joseph E. Gonzalez, Ion Stoica, Sohier Dane, Maggie Demkin, and Nate Keating. 2024. WSDM Cup - Multilingual Chatbot Arena. https://kaggle.com/competitions/wsdm-cup-multilingual-chatbot-arena. Kaggle.

[7] Jielin Qiu, Andrea Madotto, Zhaojiang Lin, Paul A Crook, Yifan Ethan Xu, Xin Luna Dong, Christos Faloutsos, Lei Li, Babak Damavandi, and Seungwhan Moon. 2024. Snapntell: Enhancing entity-centric visual question answering with retrieval augmented multimodal llm. *arXiv preprint arXiv:2403.04735* (2024).

[8] Jiaqi Wang, Xiao Yang, Kai Sun, Parth Suresh, Sanat Sharma, Adam Czyzewski, Derek Andersen, Surya Appini, Arkav Banerjee, Sajal Choudhary, et al. 2025. CRAG-MM: Multi-modal Multi-turn Comprehensive RAG Benchmark. *arXiv preprint arXiv:2510.26160* (2025).

[9] Yikuan Xia, Jiazun Chen, and Jun Gao. 2024. Winning Solution For Meta KDD Cup'24. *arXiv preprint arXiv:2410.00005* (2024).

[10] Xiao Yang, Kai Sun, Hao Xin, Yushi Sun, Nikita Bhalla, Xiangsen Chen, Sajal Choudhary, Rongze Gui, Ziran Jiang, Ziyu Jiang, et al. 2024. Crag-comprehensive rag benchmark. *Advances in Neural Information Processing Systems* 37 (2024), 10470–10490.

[11] Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. 2023. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490* (2023).

A Prompts Used in the Competition

A.1 VQA Prompt for Task1

Figure 4 shows the VQA prompt for Task1.

A.2 Data Augmentation Prompt for Task1

As shown in the prompt in Figure 5, we instruct the model to generate diverse answers along both simple and complex dimensions while preserving the original meaning.

A.3 Retrieval Query Generation Prompt for Task2 and Task3

Figure 6 shows the retrieval query generation prompt for Task2 and Task3.

A.4 Prompt for Reranking Data Construction in Task2 and Task3

Figure 7 shows the prompt we used with a large model to construct fine-tuning data for the reranking task.

A.5 Rerank Prompt for Task2 and Task3

Figure 8 shows the reranking prompt.

A.6 VQA Prompt for Task2 and Task3

Figure 9 shows the prompt used for generating the final answer after obtaining retrieval results in Task2 and Task3.

A.7 Judge Prompt

Figure 10 shows the prompt used to evaluate the consistency between the model prediction and the ground truth.

QA Prompt for Task1

You are a helpful assistant that truthfully answers user questions about the provided image with informations that might be related to the question.

Keep your response concise and to the point. If you don't know the answer, respond with ***'I don't know'***.

Figure 4: QA prompt for Task1.

Data Augmentation Prompt for Task1

Given a question and a standard answer, please help me create 20 similar standard answers. The answers should either be simplified or made more complex, but must not exceed 50 words. Output format:

1. xx (Rule: Simplified|Complexified)
2. xx (Rule: Simplified|Complexified)
3. xx
...
Question: {query}
Standard answer: {ans_full}

Figure 5: Data augmentation prompt for Task1.

Retrieval Query Generation Prompt for Task2&3

You are a web retrieval and query reformulation agent. Based on the history dialog, the current original question, and the provided image, please generate a search phrase for retrieval.

Image: {image}
The History Dialog is: {history_dialog_str}
The Origin Query is: {ori_query}

Figure 6: Retrieval query generation prompt for Task2/3.

Prompt for Reranking Data Construction in Task2&3

You need to determine whether the retrieved content is relevant to the query.
Output a JSON object with a single field 'is_relevance' whose value is a boolean (True or False).

The Origin Query is: {ori_query}
The Retrieval Infos are: {rag_content}

Figure 7: Prompt for reranking data construction in Task2/3.

Rerank Prompt for Task2&3

You need to determine the content of the search and which info can help you answer the query. The query has and retrieval information already been provided.
Output the most relevant Info number List like [x, xx, xxx, ...]. If no relevant items, output [].

The original provided image is: {image}
The retrieval information is: {rag_info}
Please rerank the information to ask: {query}

Figure 8: Rerank prompt for Task2/3.

VQA Prompt for Task2&3

You are a helpful assistant that truthfully answers user questions about the provided image and the retrieval information.
The retrieval information may not related to the provided query and image.
Please pay attention to identifying that information and answer the query with image. And the correct answer is satisfied following rules:

1. The answer is correct if it captures all the key information.
2. The answer is correct even if phrased differently as long as the meaning is the same.
3. The answer is incorrect if it contains incorrect information or is missing essential details. For example, when answer a question about time, it's better to answer with day, month and year.

Remeber the above rules and keep your response concise and to the point. Note that the answer must in short!!!!

The original provided image is: {image}
The retrieval information is: {rag_info}
History messages: {history_messages}
Please ask: {query}

Figure 9: VQA prompt for Task2/3.

Judge Prompt

You are an expert evaluator for question answering systems.
Your task is to determine if a prediction correctly answers a question based on the ground truth.

Rules:
1. The prediction is correct if it captures all the key information from the ground truth.
2. The prediction is correct even if phrased differently as long as the meaning is the same.
3. The prediction is incorrect if it contains incorrect information or is missing essential details.

Output a JSON object with a single field 'accuracy' whose value is true or false.

Question: {query}
Ground truth: {ground_truth}
Prediction: {agent_response}

Figure 10: Judge prompt.