

CIIR@LiveRAG 2025: Optimizing Multi-Agent Retrieval Augmented Generation through Self-Training

Alireza Salemi
University of Massachusetts Amherst
United States
asalemi@cs.umass.edu

Mukta Maddipatla
University of Massachusetts Amherst
United States
mmaddipatla@umass.edu

Hamed Zamani
University of Massachusetts Amherst
United States
zamani@cs.umass.edu

ABSTRACT

This paper presents mRAG, a multi-agent retrieval-augmented generation (RAG) framework composed of specialized agents for sub-tasks such as planning, searching, reasoning, and coordination. Our system uses a self-training paradigm with reward-guided trajectory sampling to optimize inter-agent collaboration and enhance response generation. Evaluated on DataMorgana-derived datasets during the SIGIR 2025 LiveRAG competition, mRAG outperforms conventional RAG baselines. We further analyze competition outcomes and showcase the framework’s strengths with case studies, demonstrating its efficacy for complex, real-world RAG tasks.

1 INTRODUCTION

RAG enhances large language models (LLMs) by integrating external retrieval mechanisms, addressing limitations like static knowledge and lack of grounding in current, verifiable sources [1, 9, 13]. Unlike traditional LLMs limited by fixed training data, RAG enables real-time access to relevant documents from sources such as search engines or databases [14, 15], improving accuracy and relevance without retraining. Recently, autonomous agents have emerged as a powerful extension of RAG [8, 17], capable of complex reasoning, tool use, and multi-step decision-making. Their modular design supports dynamic, goal-directed information synthesis, making them effective for real-world tasks requiring flexibility and depth.

While single-agent RAG systems are useful, they struggle with scalability, specialization, and effective context management. A single agent juggling multiple tasks—like query formulation, retrieval, synthesis, and validation—can suffer from context overlap, leading to inefficiencies and degraded performance. Multi-agent RAG systems address these issues through modular design. By assigning distinct roles (e.g., searcher, planner, summarizer, validator) to separate agents, each operates within a focused context, improving efficiency and task alignment. Inter-agent communication further enables better task decomposition and parallel execution, enhancing robustness and adaptability in complex reasoning workflows.

We introduce mRAG, a framework for building and optimizing multi-agent RAG systems. It consists of a set of task-specialized agents—such as a planner, searcher, and reasoner—each handling a distinct subtask. A central coordinator agent orchestrates the workflow by dynamically invoking agents, routing information, and monitoring progress toward a final response. The coordinator has control over the decision process and terminates the workflow once a satisfactory output is produced. To train agents, we adopt a self-training approach inspired by Singh et al. [16], generating diverse agent interaction trajectories for each input. These are evaluated with a reward model, and high-reward trajectories are

used as supervision. Agents are trained to reproduce these high-rewarded behaviors, promoting effective generation in future runs.

We present our experimental results from the LiveRAG¹ competition, demonstrating that the proposed multi-agent framework outperforms the traditional retrieve-then-read RAG paradigm on a dataset derived from DataMorgana [3], the competition’s data generation tool. Additionally, we analyze our performance on the competition test day and offer insights into the evaluation process. Finally, we include case studies highlighting examples where our multi-agent system achieves notably strong performance. We release our code to facilitate further research on this topic.²

2 DATA CREATION WITH DATAMORGANA

To construct training and validation data for mRAG, we used DataMorgana [3] to generate a set of QA pairs. We defined 10 different question categories that were subsequently grouped into 5 combinations (definition of each category is reported in Appendix A):

- **User Expertise:** This category specifies the user’s level of expertise on the topic by distinguishing between expert and novice users of the system (Figure 3 in Appendix A).
- **Question Type:** This specifies the question phrasing style, varying along dimensions such as length (short vs. long), formality (natural vs. query-like), and verbosity (concise vs. verbose), resulting in six distinct combinations (Figure 4 in Appendix A).
- **Answer Type:** This defines the answer phrasing, varying along the dimensions of natural vs. formal and concise vs. verbose, resulting in four distinct combinations (Figure 10 in Appendix A).
- **Question Intent:** This defines the intention behind the question, encompassing categories such as clarification, opinion, comparison, yes/no, and hypothetical questions (Figure 7 in Appendix A).
- **Answer Intent:** This specifies intents behind responses, categorized as either factual or open-ended (Figure 5 in Appendix A).
- **Premise Inclusion:** This indicates whether the question includes user-specific information, resulting in two variations: with premise and without premise (Figure 6 in Appendix A).
- **Lexical Similarity:** This specifies the lexical alignment between questions and documents: using similar terms, different terms, or referencing unpopular entities (Figure 12 in Appendix A).
- **Aspect Granularity:** This defines whether the question captures a single or multiple aspects of the topic (Figure 8 in Appendix A).
- **Interaction Type:** This defines whether the question initiates a conversation or is a follow-up question (Figure 9 in Appendix A).
- **Document Granularity:** This specifies whether one or two documents are used to generate the question, using the feature in the DataMorgana platform (Figure 11 in Appendix A).

¹Competition website can be found at: <https://liverag.tii.ae/index.php>

²Available at: <https://github.com/muktac5/CIIR-LiveRAG>

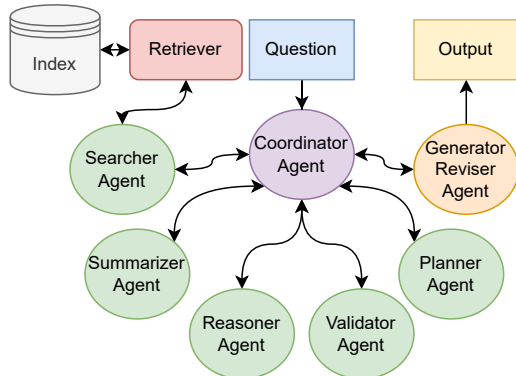


Figure 1: Overview of multi-agent RAG.

After interacting with DataMorgana, we find that using all categories simultaneously does not yield high-quality outputs. Consequently, we grouped them into five combinations when invoking DataMorgana for data generation. All combinations include the following core dimensions: User Expertise, Question Type, Answer Type, Document Granularity, Interaction Type, and Aspect Granularity. The remaining categories in each combination are as follows:

- Answer Intent and Lexical Similarity.
- Premise Inclusion.
- Question Intent.
- Premise Inclusion and Question Intent.
- Lexical Similarity and Premise Inclusion.

where we generate 4,500 question-answer pairs with DataMorgana, allocating 3,500 for training and 1,000 for evaluation of our system. Details of the data generation process are provided in Appendix A.

3 THE MRAG FRAMEWORK

Our framework employs six action-performing agents and one coordinator agent to generate responses. An overview of mRAG is shown in Figure 1. This section provides implementation and training details of the multi-agent system.

3.1 Agents in mRAG

This section describes the implementation of the agents.

Coordinator. This agent serves as the entry point of mRAG. It receives a question and a set of agents with defined input and output formats, and is responsible for assigning tasks to these agents based on their respective expertise. In each turn, the coordinator selects an appropriate agent based on the current state alongside the rationale behind selection of this agent, formats and passes the necessary inputs, and waits for the agent to complete its task and return outputs in the expected format. Upon receiving the output, the coordinator updates its internal state by appending the new outputs to the conversation history, and updates either the response (if one was generated) or the set of supporting documents (if new information was retrieved). This process is repeated iteratively. Once the computation budget is exhausted or the coordinator determines that no further actions are required and a high-quality response has been produced, it terminates the process and returns the final response along with the supporting documents. This agent

use instruct Qwen 2.5 with 7B parameters [12] with the prompt shown in Figure 15 in Appendix D.1 in each step. The detailed implementation is provided in Algorithm 1 in Appendix D.1.

Searcher. To access the corpus and collect information necessary for answering the question, the coordinator can invoke the searcher agent by providing the question, the information gathered so far, and a suggested set of aspects to guide the search. The searcher agent begins by generating a search query, which is executed using the sparse Lion retrieval model with 1B parameters (detailed implementation and corpus processing described in Appendix C) [20] to retrieve two documents. Next, the searcher agent evaluates the relevance of each retrieved document, providing justifications and marking the relevant ones. At this stage, the agent has three main options: (1) continue the search using the same query to retrieve the next two documents, (2) modify the search query and explain the rationale for the new query, or (3) terminate the search, explaining why, and return the relevant documents found so far. This process continues until either the maximum retrieval budget is reached or the agent decides that sufficient information has been gathered. This design allows the searcher agent to dynamically determine the appropriate amount of information to collect for each query. The agent uses the instruct Qwen 2.5 model with 7B parameters [12] and the prompt shown in Figure 16 in Appendix D.1 in each step. The detailed implementation of this agent is provided in Algorithm 2.

Planner. This agent is responsible for generating a sequence of steps required to produce a response to the question, based on the given question and the information collected so far. Although it can be invoked at any stage by the coordinator agent, it is recommended to be called at the beginning of the response generation process. The prompt used for this agent is shown in Figure 17 in Appendix D.1. This agent uses an instruct Qwen 2.5 with 7B parameters [12].

Summarizer. As the conversation between agents grows longer, it becomes increasingly challenging for the coordinator to track all relevant details. To mitigate this, the coordinator can invoke this agent and provide selected information to summarize the conversation or the retrieved and collected content up to that point, using the prompt shown in Figure 18 in Appendix D.1. This agent uses an instruct Qwen 2.5 with 7B parameters [12].

Reasoner. When step-by-step reasoning or analysis is required regarding the retrieved information, actions taken, or any other aspect of the process, this agent can be invoked by providing the question, relevant information, and a specific aspect to reason about. This agent uses the prompt shown in Figure 19 in Appendix D.1. This agent uses an instruct Qwen 2.5 with 7B parameters [12].

Validator. Sometimes, questions may specify certain criteria that must be satisfied. This agent can be invoked when a response has been generated and the coordinator needs to ensure that all criteria are addressed. The coordinator provides the question, retrieved information, and a response to this agent, which then extracts the criteria from the question and verifies whether they are fulfilled in the response and provides rationals for each of its decisions. This process uses the prompt shown in Figure 20 in Appendix D.1 using an instruct Qwen 2.5 with 7B parameters [12].

Generator/Reviser. To generate a response, the coordinator invokes this agent by providing the question, supporting information collected during the process, a response plan, and a set of key aspects that should be included. The agent then generates a response that incorporates these elements. Additionally, if the coordinator determines that the response requires revision—either due to newly collected information or unmet criteria—it can call this agent again with suggestions outlining the deficiencies and how to address them. This agent uses the prompts shown in Figure 21 in Appendix D.1. Due to competition constraints, we use the instruction-tuned Falcon 3 model with 10B parameters [19] for this agent.

3.2 Optimization through Self-Training

End-to-end training of multi-agent systems is challenging due to computational inefficiency and the difficulty of propagating gradients across agents [4, 5]. To simplify the training process, we make a key assumption: agents operate independently and function solely based on their provided inputs. With this assumption, the probability of a trajectory of actions taken by the agents, denoted as $\tau = a_1 a_2 \dots a_n$, simplifies from the full joint distribution

$$p(a_1 \dots a_n | x) = p(a_1 | x) p(a_2 | a_1, x) \dots p(a_n | a_{n-1}, \dots, a_1, x)$$

to the following factorized form:

$$p(\tau | x) = p(a_1 \dots a_n | x) = p(a_1 | x) p(a_2 | x) \dots p(a_n | x),$$

assuming conditional independence of agent actions given the input x . With this simplification, we can use Self-Training [16] to optimize the system end to end. We first sample $T = 8$ diverse trajectories for each input in the training set by applying a high sampling temperature. A reward model RM is then applied to the final response of each trajectory for scoring. We retain only the trajectory with the highest reward score for each input (allowing up to three trajectories in case of ties, to avoid overfitting on simpler examples). These selected trajectories are used to train the agents to reproduce the optimal sequence of actions. Following Singh et al. [16], we normalize the reward by setting the reward $r(\tau, x)$ of the best-performing trajectory to 1 (by setting the threshold score to match the highest observed reward) and others to zero. Therefore, the training objective is to maximize:

$$\mathbb{E}_{\tau \sim D} [r(\tau, x) \log p(\tau | x)] = \mathbb{E}_{\tau \sim D} \left[\sum_{a_i \in A} \log p(a_i | x) \right]$$

where A is the set of actions performed by trainable agents (i.e., all agents except the generator/reviser that is fixed due to competition constraints) and D is the set of all selected trajectories after applying reward model. This objective is equivalent to supervised fine-tuning [18] of the agents on their own best-performing outputs within the selected trajectories. This encourages each agent to replicate the behavior observed in the highest-reward (i.e., most successful) trajectory, promoting consistency with effective sequences. The training details and parameters are provided in Appendix D.2.

Reward Model. The competition does not provide a publicly available scoring function. However, based on the provided guidelines, we define and utilize two distinct reward signals (more implementation details are explained in Appendix D.3):

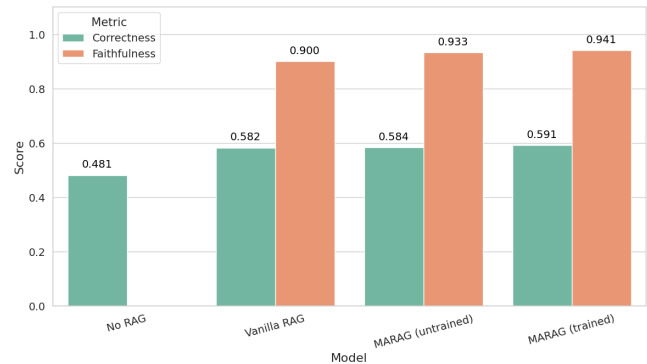


Figure 2: Evaluation of mRAG and baselines using the introduced reward models in Section 3.2 and Appendix D.3.

- **correctness:** We adopt the recall-oriented nugget-based reward proposed by Pradeep et al. [11]. In this approach, we first extract the atomic aspects that should be present in the output from the ground truth response using the prompt shown in Figure 22 in Appendix D.3. Each extracted aspect is then evaluated for alignment with the generated response using the prompt shown in Figure 23 in Appendix D.3, with scores assigned on a scale from -1 to 2 . These scores are subsequently normalized to the range $[0, 1]$. The final correctness reward is computed as the average normalized score across all nugget aspects. The implementation is detailed in Algorithm 3 in Appendix D.3.
- **Faithfulness:** This reward model evaluates the faithfulness of the generated response with respect to the retrieved information. Following Es et al. [2], we first extract the atomic aspects from the generated response using the prompt in Figure 24 in Appendix D.3. Each aspect is then scored for how well it is supported by the retrieved documents using the prompt in Figure 25 in Appendix D.3, on a scale from -1 to 1 , which is subsequently normalized to the range $[0, 1]$. The final faithfulness score is computed as the average normalized score across all extracted aspects. The implementation is detailed in Algorithm 4 in Appendix D.3.

To compute the final reward, we run each reward model five times and take the average score for each. We observed that mRAG performs worst on the correctness reward. To emphasize improvement in this area, we assign a weight of 4 to the correctness reward and a weight of 1 to faithfulness reward, and compute a weighted average to obtain the final reward. We use instruct Qwen 2.5 with 14B parameters [12] as the LLM for all reward functions.

4 EXPERIMENTS

This section presents our findings. The experimental setup is described in Appendix D.4. For comparison, we consider two baseline models: a standard language model without retrieval augmentation (vanilla LLM) and a retrieval-augmented generation model (vanilla RAG) that follows a retrieve-then-read paradigm. Detailed implementation of the baseline systems is provided in Appendix B.

Results on the created test set from DataMorgana. We compare mRAG with two baseline approaches: a vanilla LLM without retrieval augmentation (RAG) and a vanilla RAG system employing a retrieve-then-read paradigm. Evaluation results on the test set

Table 1: Results of the top 20 participating teams in the LiveRAG competition. The competition-defined evaluation metrics assign scores in the range of -1 and 2 for correctness, and -1 and 1 for faithfulness. The final ranking is determined by sorting the teams based on correctness scores.

Rank	Team	Correctness	Faithfulness
1	Magikarp	1.231	0.656
2	UDInfo	1.200	0.623
3	RMIT-ADMS	1.199	0.477
4	RAGtifier	1.134	0.552
5	HLTCOE	1.070	0.340
6	Ragmatazz	1.011	0.519
7	mRAG	0.996	0.418
8	RUC_DeepSearch	0.969	0.387
9	Ped100X	0.928	0.043
10	PRMAS-DRCA	0.922	0.410
11	Emorag	0.890	0.556
12	Graph-Enhanced RAG	0.875	0.529
13	Hybrid Search with Graph	0.875	0.315
14	Multi-Agent Adaptive RAG	0.836	0.200
15	Starlight	0.818	0.433
16	BagBag	0.694	-0.911
17	UniClustRAG	0.685	0.460
18	METURAG	0.673	0.325
19	NoobRAG	0.655	0.154
20	DeepRAG	0.566	0.097

derived from DataMorgana are presented in Figure 2. The findings indicate that mRAG, when optimized using the method described in Section 3.2, outperforms both baselines in terms of correctness and faithfulness. Notably, even without training, mRAG achieves comparable correctness to the vanilla RAG while demonstrating superior faithfulness. Across all configurations, RAG-based methods—mRAG and vanilla RAG—consistently outperform the non-RAG baseline. Based on these results, mRAG demonstrates the best overall performance on the constructed dataset from DataMorgana.

Results on the test set from the competition. The results of mRAG and other top 20 participating systems on the test set provided by the LiveRAG competition organizers during the official test day are presented in Table 1. The specific implementation details of the correctness and faithfulness metrics reported in Table 1 are not publicly available. However, the correctness scores range from -1 to 2, while the faithfulness scores range from -1 to 1, as defined by the competition organizers. We report the raw, unnormalized scores as provided by the competition evaluation process. According to these results, which are ranked based on the correctness metric, mRAG achieves the 7th position among all participating teams.

Unstability of the recall-oriented nugget-based evaluation of correctness. In one of our experiments, we found that simply prompting the LLM to produce longer responses—without any additional training—raised the nugget-based recall reward from 0.528 to 0.584. This highlights a key limitation in current evaluation methods: as responses become longer, they are more likely to include a greater number of nuggets, thereby inflating recall-oriented

rewards. However, this comes at the cost of conciseness and clarity, resulting in redundant and potentially less helpful output for users. This suggests that recall-focused metrics may encourage verbosity over precision. To address this, future evaluation strategies should aim to balance both recall and precision by incorporating metrics that also account for relevance and brevity, better aligning with user expectations for clear and informative responses.

5 CASE STUDY

To show the effectiveness of mRAG, we present two case studies that illustrate the system’s decision-making trajectories and inter-agent interactions in response to user queries. Detailed descriptions of these case studies are provided in Appendix E. We examine two notable behaviors exhibited by the system after training, as observed in its responses to the queries: “safety concerns hydrogen steam reforming” and “How did John Ball influence golf history at Hoylake, and what strategic challenges does the course present regarding out of bounds?”. These two notable behaviors emerged:

- **Breaking down the question and collecting information about each aspect:** In both cases, as illustrated in Table 2 and Table 3 in Appendix E, the system decomposes each question into multiple aspects that must be addressed to formulate a comprehensive response. The coordinator invokes the searcher agent twice in each case to retrieve information corresponding to two distinct facets of the query. The retrieved information is then relayed to other agents responsible for response generation, reasoning, and validation. These interactions demonstrate that the coordinator has effectively learned how to orchestrate the contributions of other agents—particularly the searcher, which plays a critical role in supporting information acquisition.
- **Changing search query after retrieval failure:** In both examples presented in Table 2 and Table 3 in Appendix E, the searcher agent demonstrates adaptive behavior in response to retrieval failures. When the initial query fails to yield sufficient or relevant results, the searcher does not immediately reformulate the query. Instead, it reuses the same query for up to five retrieval attempts, aiming to extract useful information from lower-ranked documents—accounting for potential imperfections in the retriever. Only after these repeated attempts does the agent revise the query to explore alternative formulations. This behavior indicates that the searcher has learned a robust strategy for balancing persistence with query reformulation in the presence of retrieval noise or deficiencies.

Further details of the case studies are provided in Appendix E, which highlight several notable emergent behaviors exhibited by the system after training.

6 CONCLUSION

This paper presented mRAG, a multi-agent framework designed for RAG, along with a training methodology to optimize inter-agent communication. We trained mRAG using data generated from DataMorgana and participated in the SIGIR 2025 LiveRAG competition. Experimental results demonstrated that mRAG consistently outperforms standard vanilla RAG baselines. Additionally, mRAG achieved a 7th place ranking based on automatic evaluation in the competition, highlighting its effectiveness in real-world settings.

REFERENCES

- [1] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. arXiv:2310.11511 [cs.CL] <https://arxiv.org/abs/2310.11511>
- [2] Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. 2024. RAGAs: Automated Evaluation of Retrieval Augmented Generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, Nikolaos Aletras and Orphee De Clercq (Eds.). Association for Computational Linguistics, St. Julians, Malta, 150–158. <https://aclanthology.org/2024.eacl-demo.16/>
- [3] Simone Filice, Guy Horowitz, David Carmel, Zohar Karmin, Liane Lewin-Eytan, and Yoelle Maarek. 2025. Generating Diverse Q&A Benchmarks for RAG Evaluation with DataMorgana. arXiv:2501.12789 [cs.CL] <https://arxiv.org/abs/2501.12789>
- [4] Kailash Gogineni, Yongsheng Mei, Karthikeya Gogineni, Peng Wei, Tian-Shing Lan, and Guru Venkataramani. 2024. Characterizing and Optimizing the End-to-End Performance of Multi-Agent Reinforcement Learning Systems. *2024 IEEE International Symposium on Workload Characterization (IISWC) (2024)*, 224–235. <https://api.semanticscholar.org/CorpusID:272602792>
- [5] Kailash Gogineni, Peng Wei, Tian Lan, and Guru Venkataramani. 2023. Towards Efficient Multi-Agent Learning Systems. arXiv:2305.13411 [cs.MA] <https://arxiv.org/abs/2305.13411>
- [6] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The Curious Case of Neural Text Degeneration. arXiv:1904.09751 [cs.CL] <https://arxiv.org/abs/1904.09751>
- [7] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*. <https://openreview.net/forum?id=nZevKeeFyf9>
- [8] Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-R1: Training LLMs to Reason and Leverage Search Engines with Reinforcement Learning. arXiv:2503.09516 [cs.CL] <https://arxiv.org/abs/2503.09516>
- [9] To Eun Kim, Alireza Salemi, Andrew Drozdov, Fernando Diaz, and Hamed Zamani. 2024. Retrieval-Enhanced Machine Learning: Synthesis and Opportunities. arXiv:2407.12982 [cs.LG] <https://arxiv.org/abs/2407.12982>
- [10] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR* abs/1412.6980 (2014). <https://api.semanticscholar.org/CorpusID:6628106>
- [11] Ronak Pradeep, Nandan Thakur, Shivani Upadhyay, Daniel Campos, Nick Craswell, and Jimmy Lin. 2025. The Great Nugget Recall: Automating Fact Extraction and RAG Evaluation with Large Language Models. arXiv:2504.15068 [cs.IR] <https://arxiv.org/abs/2504.15068>
- [12] Qwen, , An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 Technical Report. arXiv:2412.15115 [cs.CL] <https://arxiv.org/abs/2412.15115>
- [13] Alireza Salemi and Hamed Zamani. 2024. Evaluating Retrieval Quality in Retrieval-Augmented Generation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Washington DC, USA) (SIGIR '24). Association for Computing Machinery, New York, NY, USA, 2395–2400. <https://doi.org/10.1145/3626772.3657957>
- [14] Alireza Salemi and Hamed Zamani. 2024. Learning to Rank for Multiple Retrieval-Augmented Models through Iterative Utility Maximization. arXiv:2410.09942 [cs.CL] <https://arxiv.org/abs/2410.09942>
- [15] Alireza Salemi and Hamed Zamani. 2024. Towards a Search Engine for Machines: Unified Ranking for Multiple Retrieval-Augmented Large Language Models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Washington DC, USA) (SIGIR '24). Association for Computing Machinery, New York, NY, USA, 741–751. <https://doi.org/10.1145/3626772.3657733>
- [16] Avi Singh, John D. Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J. Liu, James Harrison, Jaehoon Lee, Kelvin Xu, Aaron Parisi, Abhishek Kumar, Alex Alemi, Alex Rizkowsky, Azade Nova, Ben Adlam, Bernd Bohnet, Gamaleldin Elsayed, Hanie Sedghi, Igor Mordatch, Isabelle Simpson, Izzeddin Gur, Jasper Snoek, Jeffrey Pennington, Jiri Hron, Kathleen Kenealy, Kevin Swersky, Kshiteej Mahajan, Laura Culp, Lechao Xiao, Maxwell L. Bileschi, Noah Constant, Roman Novak, Rosanne Liu, Tris Warkentin, Yundi Qian, Yamini Bansal, Ethan Dyer, Behnam Neyshabur, Jascha Sohl-Dickstein, and Noah Fiedel. 2024. Beyond Human Data: Scaling Self-Training for Problem-Solving with Language Models. arXiv:2312.06585 [cs.LG] <https://arxiv.org/abs/2312.06585>
- [17] Aditi Singh, Abul Ehtesham, Saket Kumar, and Tala Talaei Khomei. 2025. Agentic Retrieval-Augmented Generation: A Survey on Agentic RAG. arXiv:2501.09136 [cs.AI] <https://arxiv.org/abs/2501.09136>
- [18] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2* (Montreal, Canada) (NIPS'14). MIT Press, Cambridge, MA, USA, 3104–3112.
- [19] Falcon-LLM Team. 2024. The Falcon 3 Family of Open Models. <https://huggingface.co/blog/falcon3>
- [20] Hansi Zeng, Julian Killingback, and Hamed Zamani. 2025. Scaling Sparse and Dense Retrieval in Decoder-Only LLMs. arXiv:2502.15526 [cs.IR] <https://arxiv.org/abs/2502.15526>

A DETAILS OF CREATING DATA WITH DATAMORGANA

This section provides verbatim definitions of all categorization schemas and their corresponding probability distributions used to control generation with DATAMORGANA. The following figures present the individual category schemata used to structure prompt-based question generation with DATAMORGANA:

- **User Expertise:** the definition of each subcategory and its probability of generating data with DataMorgana are shown in Figure 3.
- **Question Type:** the definition of each subcategory and its probability to generate data with DataMorgana are shown in Figure 4.
- **Answer Type:** the definition of each subcategory and its probability of generating data with DataMorgana are shown in Figure 10.
- **Question Intent:** the definition of each subcategory and its probability of generating data with DataMorgana are shown in Figure 7.
- **Answer Intent:** the definition of each subcategory and its probability of generating data with DataMorgana are shown in Figure 5.
- **Premise Inclusion:** the definition of each subcategory and its probability of generating data with DataMorgana are shown in Figure 6.
- **Lexical Similarity:** the definition of each subcategory and its probability of generating data with DataMorgana are shown in Figure 12.
- **Aspect Granularity:** the definition of each subcategory and its probability of generating data with DataMorgana are shown in Figure 8.
- **Interaction Type:** the definition of each subcategory and its probability of generating data with DataMorgana are shown in Figure 9.
- **Document Granularity:** the definition of each subcategory and its probability of generating data with DataMorgana are shown in Figure 11.

As noted before, we observed that jointly conditioning on all categories often led to incoherent or degraded outputs. To mitigate this, we designed the following five compatible category combinations for controlled generation:

- (1) User Expertise, Question Type, Answer Type, Document Granularity, Interaction Type, Aspect Granularity, Answer Intent, and Lexical Similarity.
- (2) User Expertise, Question Type, Answer Type, Document Granularity, Interaction Type, Aspect Granularity, and Premise Inclusion.
- (3) User Expertise, Question Type, Answer Type, Document Granularity, Interaction Type, Aspect Granularity, and Question Intent.

```

user_expertise = {
  "categorization_name": "user-expertise",
  "categories": [
    {
      "name": "expert",
      "description": "an expert on the subject discussed in the
documents, therefore he asks complex questions.",
      "probability": 0.50
    },
    {
      "name": "novice",
      "description": "a person with very basic knowledge on the topic
discussed in the topic. Therefore, he asks very simple questions.",
      "probability": 0.50
    }
  ]
}

```

Figure 3: User expertise categorization schema and probabilities.

- (4) User Expertise, Question Type, Answer Type, Document Granularity, Interaction Type, Aspect Granularity, Premise Inclusion, and Question Intent.
- (5) User Expertise, Question Type, Answer Type, Document Granularity, Interaction Type, Aspect Granularity, Lexical Similarity, and Premise Inclusion.

B BASELINES IMPLEMENTATIONS DETAILS

As baselines in this paper, we utilize two distinct models:

- **No-Retrieval Baseline:** We use the same generator as in mRAG, namely the instruction-tuned Falcon model with 10B parameters³ [19], but without any retrieval. The model directly answers the question using the prompt shown in Figure 13.
- **Vanilla RAG:** Again employing the instruction-tuned Falcon 10B model [19], this baseline incorporates retrieval. Documents are first retrieved using the same retrieval model and configuration as mRAG, with the question itself as the query. The retrieved documents are then passed along with the question to the LLM to generate a response using the prompt shown in Figure 14.

where both baselines use nucleus sampling [6] with a temperature of 0.1 for generation. For retrieval, we consistently retrieve two documents, similar to mRAG.

C RETRIEVER IMPLEMENTATION DETAILS

Fineweb Corpus: To enable efficient indexing and document-level retrieval, we preprocessed the original FineWeb corpus⁴ by splitting long documents into smaller overlapping chunks. Each chunk was limited to 512 tokens with an 80-token overlap to preserve local coherence. This chunking strategy expanded the dataset from approximately 14 million original documents to nearly 29 million indexed units. For consistency during inference and indexing, we assigned sequential numerical IDs to these chunks (e.g., 1, 2, 3, ...), and stored a mapping between these IDs and the original FineWeb document identifiers for traceability.

³Available at: <https://huggingface.co/tiiuae/Falcon3-10B-Instruct>

⁴Available at: <https://huggingface.co/datasets/HuggingFaceFW/fineweb/viewer/sample-10BT>

```

question_phrasing_categorization = {
  "categorization_name": "question-phrasing-type",
  "categories": [
    {
      "name": "concise and natural",
      "description": "You should generate a concise, direct, and naturally
phrased question consisting of a few words.",
      "probability": 0.35
    },
    {
      "name": "verbose and natural",
      "description": "You should generate a relatively long question
consisting of more than 9 words, written in fluent, natural-sounding
language.",
      "probability": 0.35
    },
    {
      "name": "short search query",
      "description": "You should generate a short query consists of less
than 7 words phrased as a typed web query for search engines only
keywords, without punctuation and without a natural-sounding structure.",
      "probability": 0.1
    },
    {
      "name": "long search query",
      "description": "You should generate a long query consists of more
than 7 words phrased as a typed web query for search engines only
keywords, without punctuation and without a natural-sounding structure.",
      "probability": 0.1
    },
    {
      "name": "concise and fancy",
      "description": "You should generate a short, well-structured question
expressed in a stylistically rich or sophisticated manner.",
      "probability": 0.05
    },
    {
      "name": "verbose and fancy",
      "description": "You should generate a long and elaborate question
phrased with refined, elevated, or formal language, often using complex
sentence structures.",
      "probability": 0.05
    }
  ]
}

```

Figure 4: Question type categorization schema and probabilities.

Retrieval Model: For our retrieval setup, we used a state-of-the-art sparse retrieval model, Lion, from Hugging Face [20].⁵ This 1-billion-parameter model was used to index the documents and retrieve results, following the prior methodology from our lab [20]. This approach achieved state-of-the-art performance on the MS MARCO dataset and demonstrated comparable results on the FineWeb dataset.

Indexing: During indexing, we encoded each document chunk into a sparse vector of dimension 128k (corresponding to the model's

⁵<https://huggingface.co/hzeng/Lion-SP-1B-llama3-marco-mntp>

```
factuality_categorisation = {
  "categorization_name": "factuality",
  "categories": [
    {
      "name": "factoid",
      "description": "A question seeking a specific, concise piece
of information or a short fact about a particular subject, such as a
name, date, or number.",
      "probability": 0.50
    },
    {
      "name": "open-ended",
      "description": ("question inviting detailed or exploratory
responses, encouraging discussion or elaboration."
      "e.g., 'what caused the French revolution?"),
      "probability": 0.50
    }
  ]
}
```

Figure 5: Answer Intent categorization schema and probabilities.

```
premise_categorization = {
  "categorization_name": "premise-categorization",
  "categories": [
    {
      "name": "without premise",
      "description": "a question that does not contain any premise
or any information about the user.",
      "probability": 0.70
    },
    {
      "name": "with premise",
      "description": "a question starting with a very short premise,
where the users reveal their needs or some information about
themselves.",
      "probability": 0.30
    }
  ]
}
```

Figure 6: Premise Inclusion categorization schema and probabilities.

vocabulary size), retaining only non-zero token-weight pairs. These representations were stored in an inverted index on disk. The resulting index occupies approximately 571 GB of storage.

Retrieval setup: At inference time, incoming queries were tokenized and encoded into sparse query vectors using the same model and projection head. Retrieval was performed via CPU-side sparse dot-product computation between the query vector and the inverted index, accelerated using Numba⁶ for multi-threaded execution. We

⁶Available at: <https://github.com/numba/numba>

```
question_intent_categorization = {
  "categorization_name": "question-intent-type",
  "categories": [
    {
      "name": "clarification",
      "description": "A question seeking further explanation or details about a
specific concept, term, or methodology",
      "probability": 0.20,
    },
    {
      "name": "opinion",
      "description": "A question asking for a subjective viewpoint.",
      "probability": 0.20,
    },
    {
      "name": "comparison",
      "description": "A question that compares the information in the document
to other studies, perspectives, or contexts.",
      "probability": 0.20,
    },
    {
      "name": "yes_no_question",
      "description": "A question expecting a yes or no answer",
      "probability": 0.20,
    },
    {
      "name": "hypothetical",
      "description": "A question imagining a what-if scenario, asking about
potential futures, counterfactual histories, or theoretical cases. Use this category
only when the document includes reflective or interpretive content such as
commentary on history, society, science, technology, philosophy, or human
behavior. Ex: What if climate data from the 1800s had been digitized?",
      "probability": 0.20,
    }
  ]
}
```

Figure 7: Question Intent categorization schema and probabilities.

```
question_aspect_categorisation = {
  "categorization_name": "aspect-type",
  "categories": [
    {
      "name": "single-aspect",
      "description": "A question focused on one specific aspect or dimension of
a concept or entity (e.g., What are the benefits of using AI in diagnostics?).",
      "probability": 0.50
    },
    {
      "name": "multi-aspect",
      "description": "A question about two different aspects of the same
entity/concept (e.g., What are the advantages of AI-powered diagnostics, and
what are the associated risks of bias in medical decision-making?).",
      "probability": 0.50
    }
  ]
}
```

Figure 8: Aspect Granularity categorization schema and probabilities.

used a top-*k* retrieval strategy with *k* = 2 by default, and applied a threshold of 0.0 to filter low-scoring documents. The entire retrieval

```

question_turn_categorisation = {
  "categorization_name": "question-turn",
  "categories": [
    {
      "name": "single-turn",
      "description": "The initial question of a conversation, self-contained and understandable without any prior context.",
      "probability": 0.50,
    },
    {
      "name": "two-turn",
      "description": "A follow-up or compound question that either builds on previous context or combines two sub-questions on related or evolving concepts.",
      "probability": 0.50,
    }
  ]
}

```

Figure 9: Interaction type categorization schema and probabilities.

```

answer_type = {
  "categorization_name": "answer-phrasing-type",
  "categories": [
    {
      "name": "concise and natural",
      "description": "Expects a brief, direct answer, typically a short phrase or sentence using everyday, accessible language.",
      "probability": 0.4
    },
    {
      "name": "verbose and natural",
      "description": "Expects a detailed yet conversational response, typically one or more full sentences.",
      "probability": 0.4
    },
    {
      "name": "concise and fancy",
      "description": "Expects a brief answer, typically one sentence or phrase, but expressed in an elevated, stylistically rich, or formal tone.",
      "probability": 0.1
    },
    {
      "name": "verbose and fancy",
      "description": "Expects a longer and more elaborate answer, delivered in a refined, ornate, or sophisticated style, often using complex sentence structures and rich vocabulary.",
      "probability": 0.1
    }
  ]
}

```

Figure 10: Answer type categorization schema and probabilities.

```

multi_doc_categorization = {
  "categorization_name": "number-of-documents",
  "categories": [
    {
      "name": "multi-doc",
      "description": (
        "The information required to answer the question needs to come from two documents, specifically, "
        "the first document must provide information about the first entity/concept, while the second must "
        "provide information about the second entity/concept."),
      "probability": 0.5,
      "is_multi_doc": True
    },
    {
      "name": "single-doc",
      "description": (
        "The information required to answer the question can be found in a single document, "
        "which contains all the necessary information about the entity/concept."),
      "probability": 0.5,
      "is_multi_doc": False
    }
  ]
}

```

Figure 11: Document Granularity categorization schema and probabilities.

```

document_type_categorization = {
  "categorization_name": "linguistic-variation-type",
  "categories": [
    {
      "name": "similar-to-document",
      "description": "phrased using the same terminology and phrases appearing in the document (e.g., for the document 'The Amazon River has an average discharge of about 215,000–230,000 m3/s', 'what is the average discharge of the Amazon river')",
      "probability": 0.3
    },
    {
      "name": "distant-from-document",
      "description": "phrased using terms completely different from the ones appearing in the document (e.g., for a document 'The Amazon River has an average discharge of about 215,000–230,000 m3/s', 'How much water run through the Amazon?')",
      "probability": 0.4
    },
    {
      "name": "unpopular-entities",
      "description": "a question focusing on rare, less-documented, or emerging entities not widely mentioned in the fineweb corpora.",
      "probability": 0.3
    }
  ]
}

```

Figure 12: Lexical Similarity categorization schema and probabilities.

Non-RAG Baseline Prompt

```
You are a helpful assistant. You will be given a question and you need to answer it.

# question: {QUERY}
```

Figure 13: Prompt used with the Non-RAG baseline.

RAG Baseline Prompt

```
You are a helpful assistant. You will be given a question and a set of supporting documents. You can use the documents to answer the question accurately. You should generate a response that is relevant to the question and supported by the documents.

# input:
- "question": the question you need to answer
- "supporting documents": a list of documents that can be used to answer the question.

# output: You should generate a valid JSON object enclosed in the ``json`` block. The JSON object should contain the following fields:
- "response": the answer to the question. The answer should be relevant to the question and supported by the documents.

# question: {QUERY}
# documents:

Document 1: [DOC_1]
...
Document K: [DOC_K]
```

Figure 14: Prompt used with the RAG baseline that uses retrieve and read paradigm.

stack was exposed through a FastAPI⁷ service that supports both single and batched queries.

D MRAG'S IMPLEMENTATIONS DETAILS

D.1 Prompts and Algorithms

This section outlines the implementation details of the agents that collectively constitute mRAG. Each agent is instantiated with a specific role and is prompted accordingly using either the Qwen 2.5 model with 7B parameters⁸ [12] or, in the case of response generation and revision, the Falcon 3 model with 10B parameters [19]. The coordinator agent, whose prompt and logic are detailed in Figure 15 and Algorithm 1, manages the overall workflow by dispatching tasks to other agents based on context and agent specialization. It iteratively processes agent outputs, updating the conversation history and deciding when to terminate with a final answer. The planner (Figure 17) generates a high-level sequence of reasoning

steps, while the searcher (Figure 16, Algorithm 2) retrieves relevant documents using the Lion retrieval model [20] and dynamically adapts its search strategy. To support scalability and maintain coherence, the summarizer agent (Figure 18) compresses accumulated content, aiding the coordinator in maintaining context. When in-depth analytical reasoning is required, the reasoner (Figure 19) is invoked to process information on a specific aspect. The validator (Figure 20) ensures that the generated response meets all question-specific criteria by analyzing alignment between the question and the output. Finally, the generator/reviser agent (Figure 21), powered by Falcon 3 [19], produces and, if needed, refines the response to ensure completeness and coherence. Together, the agents operate in a tightly orchestrated loop, with their individual behaviors formalized in corresponding figures and algorithms.

D.2 Training Setup

To train the model, we first sample $T = 8$ diverse trajectories for each input in the training set. For all agents except the response generator/reviser, we use a temperature of 0.7 using nucleus sampling to encourage exploration and promote trajectory diversity. The response generator/reviser agent is sampled using nucleus sampling [6] with a temperature of 0.1, as it is not trainable due to competition constraints. The lower temperature reduces randomness, leading to more deterministic outputs, which allows the trainable agents to better adapt to the fixed behavior of the generator/reviser.

To train the agents, we use a single LoRA [7] adaptor shared across all trainable agents, rather than a separate adaptor for each agent. This adaptor has a rank of $r = 16$ and is used to optimize all linear layers in the instruction-tuned Qwen 2.5 with 7B parameters LLM [12]. We use Unsloth⁹ for efficient and accelerated training. The Adam optimizer [10] is employed with a learning rate of 10^{-4} . Gradient clipping with a maximum norm of 1 is applied. Training proceeds for up to 5,000 steps, including a 50-step warmup phase following a linear learning rate schedule. Models are evaluated every 500 steps on the test set, and the checkpoint with the best performance is selected. The combined maximum input and output length is set to 16,000 tokens. The batch size for all experiments is 128. All training is conducted on a single NVIDIA A100 GPU.

D.3 Reward Models

The competition does not provide an official scoring function, so we define and implement two complementary reward signals—correctness and faithfulness—based on the provided evaluation guidelines. correctness is assessed using a recall-oriented nugget-based reward following Pradeep et al. [11], wherein atomic aspects are first extracted from the ground truth (Figure 22) and then aligned with the generated response using a scoring prompt (Figure 23). These scores, normalized to $[0, 1]$, are averaged to produce the final correctness reward, with the full implementation provided in Algorithm 3. Faithfulness follows the methodology of Es et al. [2], involving the extraction of atomic claims (Figure 24) and verifying their support against the retrieved documents (Figure 25), with final scores computed as the average normalized score (Algorithm 4). To reduce variance and ensure stable reward modeling and estimation, each reward model is executed five times with a temperature of 0.5

⁷Available at: <https://github.com/fastapi>

⁸Available at: <https://huggingface.co/Qwen/Qwen2.5-7B-Instruct>

⁹Available at: <https://github.com/unslothai/unsloth>

Coordinator Agent Prompt

You are a highly capable agent, and your goal is to generate a response to the given question. This is a multi-turn task and you don't have to do it in a single turn. You have access to a set of capable agents, each with a specific skill, and you can choose the most suitable agent for each turn to help you in generating the response. In each turn, you should select the most suitable agent from a provided list of agents to help you generate the response. The agents can help you in different aspects such as validating the response, searching for information, analyzing the information, and summarizing the information. Your goal is to generate a response that is informative and relevant to the question. You can use the agents to help you in different aspects of generating the response. You can also use the agents multiple times to generate the response. You can also use the agents in any order you like. To choose an agent, you need to provide the a valid json object in ``json`` block that contains the following fields:

- "agent": the name of the agent you want to choose.
- "input": the input you want to provide to the agent. This should be a valid json object that contains the required fields for the chosen agent.
- "reason": a brief explanation of why you chose the agent for the given input.

In response, the agent will provide you with the output in the specified format for the chosen agent. The list of agents and their input and output formats are provided below:

Agents:

validator: This agent can help you verify if the generated response meets the criteria for the given question.

input:

- "question": the question the user wants to answer.
- "information": the information the you has gathered so far and want to share with the agent. This can be a summary or highlights of the information you have gathered so far and does not need to be the full information. This can be empty if you have not gathered any information yet.
- "response": the response generated to the question.

output:

- "extracted_criteria": a list of criteria that are extracted from only the user's question (e.g., being detailed, concise, short, long, etc.), containing the following fields:
 - "criteria": the criteria extracted from the user's question.
 - "criteria_explanation": an explanation of why extracted this criteria.
 - "is_response_valid": a boolean value indicating whether the response is valid according to the extracted criteria.
 - "is_response_valid_feedback": feedback on whether the response is valid according to the extracted criteria and how it can be improved.
- "is_groundedly_supported": a boolean value indicating whether the all parts of the response is grounded with supporting information.
- "is_groundedly_supported_feedback": feedback on whether the response is grounded with supporting information and how it can be improved.
- "is_correctly_answered": a boolean value indicating whether the response is correct.
- "is_correctly_answered_feedback": feedback on whether the response is correct and how it can be improved.

summarizer: This agent can help you summarize the information you have gathered so far.

input:

- "question": the question the user wants to answer.
- "information": the information you have gathered so far and want to summarize. This can be empty if you have not gathered any information yet.

output:

- "summary": the summary of the information that the agent has generated.

planner: This agent can help you plan a strategy to generate the response to the given question. It is suggested to use this agent at the beginning of the task to plan the strategy. You can also use this agent multiple times in any time during the task to plan the strategy.

input:

- "question": the question the user wants to answer.
- "information": the information you have gathered so far and want to share with the agent. This can be a summary or highlights of the information you have gathered so far and does not need to be the full information. This can be empty if you have not gathered any information yet.

output:

- "plan": the plan that the agent has generated to generate the response to the given question.

reasoner: This agent can help you reason about the information you have gathered so far about specific aspects of the question. You can use this agent to reason about the information you have gathered if you need help with reasoning about the information.

input:

- "question": the question the user wants to answer.
- "information": the information you have gathered so far and want to reason about. This can be a summary or highlights of the information you have gathered so far and does not need to be the full information. This can be empty if you have not gathered any information yet.
- "aspect": the aspect of the question you want to reason about.

output:

- "analysis": the reasoning about the information that the agent has generated.

answerer: This agent can help you generate the response to the given question. You can use this agent to generate the response to the question. You can use this agent multiple times in any time during the task to generate the response.

input:

- "question": the question the user wants to answer.
- "guidance": a guidance on how should the agent structure its response and what to include in this response. This should help the agent to generate a better response based on the information you have gathered so far, but it should not be the answer itself.
- "important_information": a string that outlines the most important information that should be included in the response.

output:

- "response": the response that the agent has generated to the given question.

reviser: This agent can help you revise the response generated by the answerer agent. You can use this agent to revise the response generated by the answerer agent if you need help with revising the response. Note that you cannot use this agent before answerer agent.

input:

- "question": the question the user wants to answer.
- "suggestion": a string that outlines the suggested revisions to the response.

output:

- "response": the revised response that the agent has generated.

searcher: This agent can help you search for information that can help you answer the given question. You can use this agent to search for information that can help you answer the question. You can use this agent multiple times in any time during the task to search for information.

input:

- "question": the question the user wants to answer.
- "information": the information you have gathered so far and want to share with the agent. This can be a summary or highlights of the information you have gathered so far and does not need to be the full information. This can be empty if you have not gathered any information yet.
- "suggestions": a set of suggestions about what aspect to search for.

output:

- "found_information": a boolean value indicating whether the search found relevant information.
- "documents": a list of documents that the agent has found that are relevant to the search query.

finisher: You can end the task by using this agent. By selecting this agent, you indicate that you have finished the task and the latest response generated by the answerer or reviser agent is the final response to the question.

input:

- "finished": a boolean value indicating that you have finished the task.

output: the agent will not provide any output.

Note: You should only provide this input to the agent in the given format and you don't need to provide the response to the agent.

question: {QUESTION}

To choose an agent, you need to provide the a valid json object in ``json`` block that contains the following fields:

- "agent": the name of the agent you want to choose.
- "input": the input you want to provide to the agent. This should be a valid json object that contains the required fields for the chosen agent.
- "reason": a brief explanation of why you chose the agent for the given input.

In response, the agent will provide you with the output in the specified format for the chosen agent. In selecting an agent, you should select the agent that you think is the most appropriate to take next. Using the same agent multiple times is allowed if you think it is necessary, but might not be useful always. In response, the agent will provide you with the necessary information to continue the conversation.

Figure 15: Prompt used for the Coordinator agent in the mRAG framework.

Searcher Agent
Prompt

You are a helpful and capable agent who's task is to help find information that can help the user answer the given question. The user provides you with a question and some information and suggestions about what aspect to search for. Your goal is to help the user find information that can help them answer the question. This is a multi-turn action. In this action, first, you need to provide a search query that will help you find relevant information to accurately answer the question. Note that search does not provide a direct answer to the question and should be used to gather useful information. In response, the user will provide you with information about the search results. Then, in the second turn, you need to analyze the provided information to verify the accuracy and relevance of the information. In response, the user will provide you with confirmation about the accuracy and relevance of the information. This continues until you no longer need more information and the provided information is sufficient and useful.

Step 1:

your input: The user provides you with the following information:

- "question": the question the user wants to answer.
- "information": a summary of the information the user has gathered so far. This can be empty if the user has not gathered any information yet.
- "suggestions": a set of suggestions about what aspect to search for.

your output: you need to provide a JSON object enclosed in ``json`` that contains the following fields:

- "search_query": the search query you suggest to find information that can help in answering the user's question.
- "search_query_explanation": an explanation of why you suggest this search query.

Step 2:

your input: The user provides you with the search results.

- "search_results": the search results using the query that you suggested.

your output: you need to provide a JSON object enclosed in ``json`` that contains the following fields:

- "query_id": the query ID of the search result.
- "relevance": a valid json list that each object in the list contains the following fields:
 - "doc_id": The document ID of the search result that you want to verify.
 - "is_relevant": a boolean value indicating whether the document is relevant to the query and contains useful information for collecting the information that can help answer the question.
 - "is_relevant_explanation": an explanation of why the document is relevant or not relevant.
- "change_search_query": a boolean value indicating whether you think we need to change the search query to find relevant information that can help answer the question. This is helpful when after multiple retrieval results from the same query is given to you and you couldn't find the relevant information. By setting this to true, you suggest a new search query to find relevant information. If you need to use the same query again, you can set this to false.
- "change_search_query_explanation": an explanation of why you need to change the search query.
- "new_search_query": the new search query you suggest to find relevant information that can help answer the question. This is only needed if you set "change_search_query" to true. Otherwise, you can leave this field empty.
- "end_search": a boolean that show that you are satisfied with the search results and you don't need to search anymore. This is helpful when you think you have found enough information and you don't need to search anymore. By setting this to true, you suggest to end the search. If you need to continue searching, you can set this to false. This should be false if you set "change_search_query" to true.
- "end_search_explanation": an explanation of why you need to end the search.

This is a multi-turn action. You will continue to provide search queries and analyze the search results until you no longer need more information and the provided information is sufficient and useful. Your output should be a valid JSON object enclosed in ``json`` that contains the fields mentioned above.

```
# question: {QUESTION}
# information: {INFORMATION}
# suggestions: {SUGGESTIONS}
```

Template for search results

This is the information resulted from your search query:

```
Query ID: {QUERY_ID}
Document 1 ID: {ID_1}
Document 1 text: {DOC_1}
Document 2 ID: {ID_2}
Document 2 text: {DOC_2}
```

Now I need you to verify the information. Be as objective as possible in your verification. Note you should strictly follow the given format and do not add any extra explanation or information to your output.

Figure 16: Prompt used for the Searcher agent in the mRAG framework.

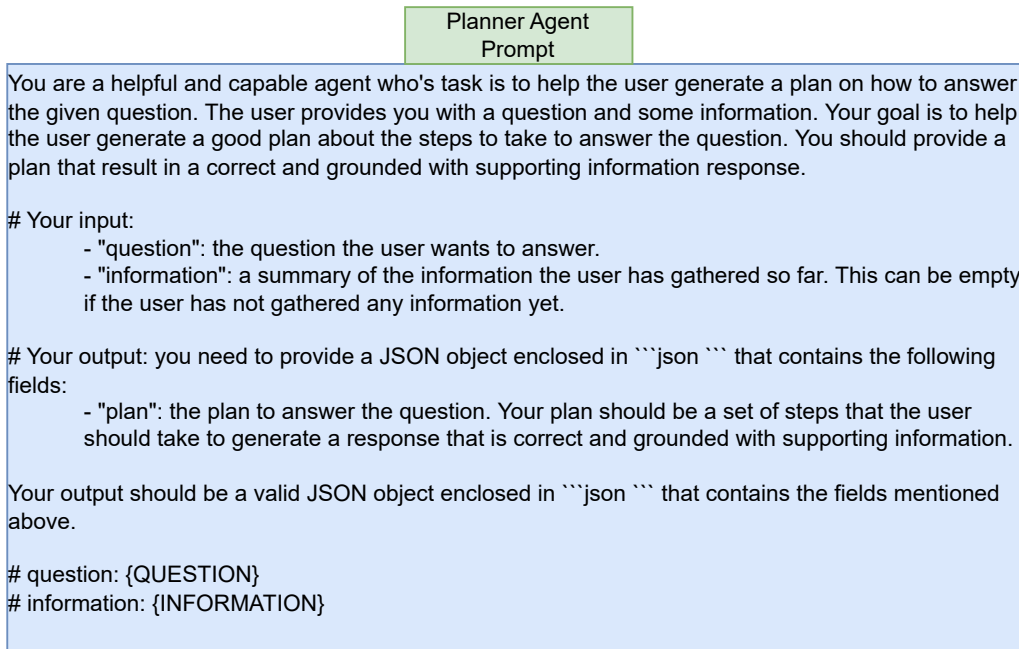


Figure 17: Prompt used for the Planner agent in the mRAG framework.

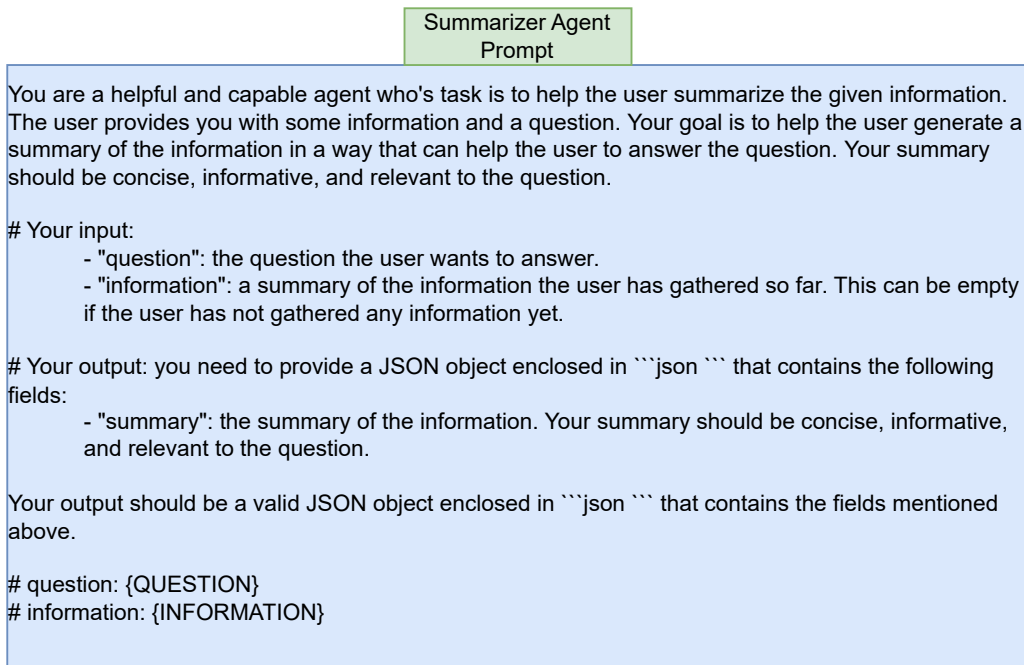


Figure 18: Prompt used for the Summarizer agent in the mRAG framework.

sampling using Nucleus Sampling [6], and the scores are averaged. Empirically, we observe that mRAG underperforms on the correctness dimension. To prioritize improvement in this area, we apply a weighting scheme in the final reward computation: a weight of 4 is

assigned to correctness, while faithfulness receives a weight of 1. The final reward is then computed as the weighted average of the two normalized scores, effectively emphasizing correctness in the optimization process to improve it more. We use instruction-tuned

Reasoner Agent Prompt

You are a helpful and capable agent who's task is to help the user analyze the given information from the given aspect by the user to help them answer the given question. In reasoning, you should provide a logical, coherent, and step by step analysis of the information in the requested aspect in a way to be helpful for answering the question. Your goal is to help the user understand the information from the given aspect. You should not answer the question but your analysis should be helpful for the user to answer the question.

Your input:

- "question": the question the user wants to answer.
- "information": a summary of the information the user has gathered so far. This can be empty if the user has not gathered any information yet.
- "aspect": the aspect the user wants to analyze the information from.

Your output: you need to provide a JSON object enclosed in ``json`` that contains the following fields:

- "analysis": a list of strings that containing step by step analysis of the information from the given aspect, where each string is a step in this analysis. Your analysis should be a logical, coherent, and step by step analysis of the information in the requested aspect.

Your output should be a valid JSON object enclosed in ``json`` that contains the fields mentioned above.

question: {QUESTION}
 # information: {INFORMATION}
 # aspect: {ASPECT}

Figure 19: Prompt used for the Reasoner agent in the mRAG framework.

Algorithm 1 Implementation of the Coordinator agent in the mRAG framework.

Input: question q , agents list A , LLM π

Output: response to the question r , supporting documents S

```

1:  $i = 0$                                 ▶ Counter of number of steps/calls to agents
2:  $finish = False$                         ▶ Initializing variables for when to end process
3:  $agents\_outputs = []$                    ▶ Initializing variables for saving the state of agents
4:  $S = \{\}$                                 ▶ Initializing variables for collecting supporting documents
5:  $r = ""$                                   ▶ Initializing variables for final response
6: while  $i < MAX\_STEPS$  and  $finish = False$  do
7:    $agent\_name, agent\_inputs = \pi(q, agents\_outputs, S)$ 
8:   if  $agent\_name = Finish$  then
9:     Break                                ▶ As long as the agent wants to continue and has budget
10:  end if
11:   $a = select\_agent(A, agent\_name)$         ▶ Select the agent from agents list
12:   $selected\_agent\_output = a(agent\_inputs)$   ▶ Call the agent with the given input parameters generated by coordinator
13:   $agents\_outputs = agents\_outputs + [selected\_agent\_output]$   ▶ Updating the state of the coordinator with the agent's output by
    appending
14:   $i = i + 1$                                 ▶ Updating the step
15:  if  $agent\_name = generator$  or  $agent\_name = reviser$  then ▶ If the selected agent generates a new response i.e., generator or reviser
16:     $r = agents\_outputs$                     ▶ Replace the response with the new response
17:  end if
18:  if  $agent\_name = searcher$  then        ▶ If the selected agent collects supporting information i.e., searcher agent
19:     $S = S \cup agents\_outputs$             ▶ Add the new supporting information to all supporting information
20:  end if
21: end while
22: return  $r, S$                             ▶ Returning the response and supporting documents

```

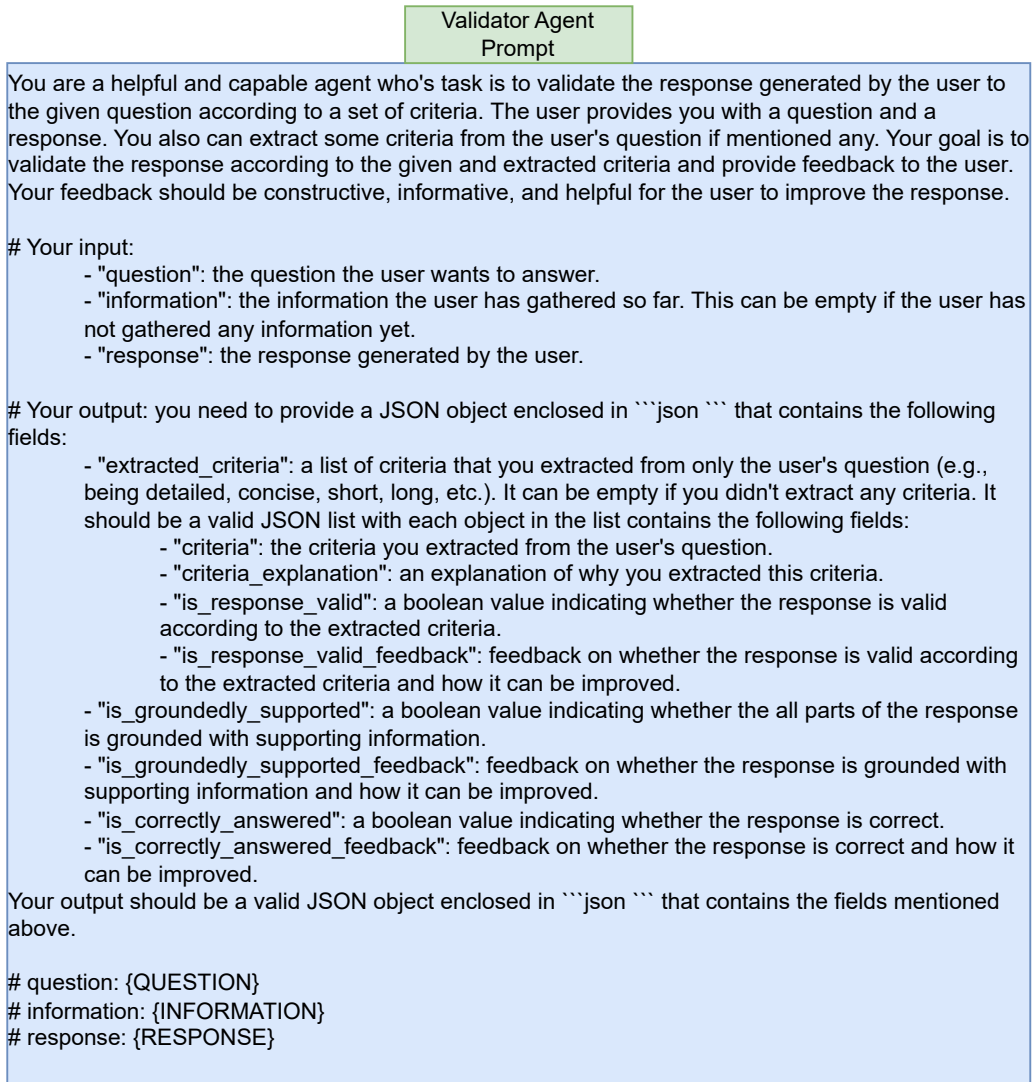


Figure 20: Prompt used for the Validator agent in the mRAG framework.

Qwen 2.5 with 14 billion parameters¹⁰ [12] as the backbone LLM for all the reward functions.

D.4 Inference Setup

For inference with LLMs, we utilize vLLM library¹¹ to serve the language models. A temperature of 0.1 with nucleus sampling is used for generation. The maximum combined input-output token length per agent is set to 32,000 tokens. The coordinator agent has a budget of up to 30 calls to other agents. The searcher agent can reuse the same query up to 5 times, with no additional budget constraints. Details of the retrieval model setup are provided in Appendix C.

¹⁰ Available at: <https://huggingface.co/Qwen/Qwen2.5-14B-Instruct>

¹¹ Available at: <https://github.com/vllm-project/vllm>

E CASE STUDIES

The proposed multi-agent system demonstrates notable and impressive behaviors, particularly in the decision-making processes of the coordinator and searcher agents, which are tasked with more complex responsibilities compared to others. To explore these behaviors in greater depth, we examine two representative case studies.

Multi-aspect questions. In cases where the input question encompasses multiple dimensions and necessitates the collection of information across diverse aspects, the system must effectively retrieve and integrate information from these various dimensions to produce a coherent and high-quality response. One example of such a query is “safety concerns in hydrogen steam reforming,” which requires the system to identify and synthesize information across

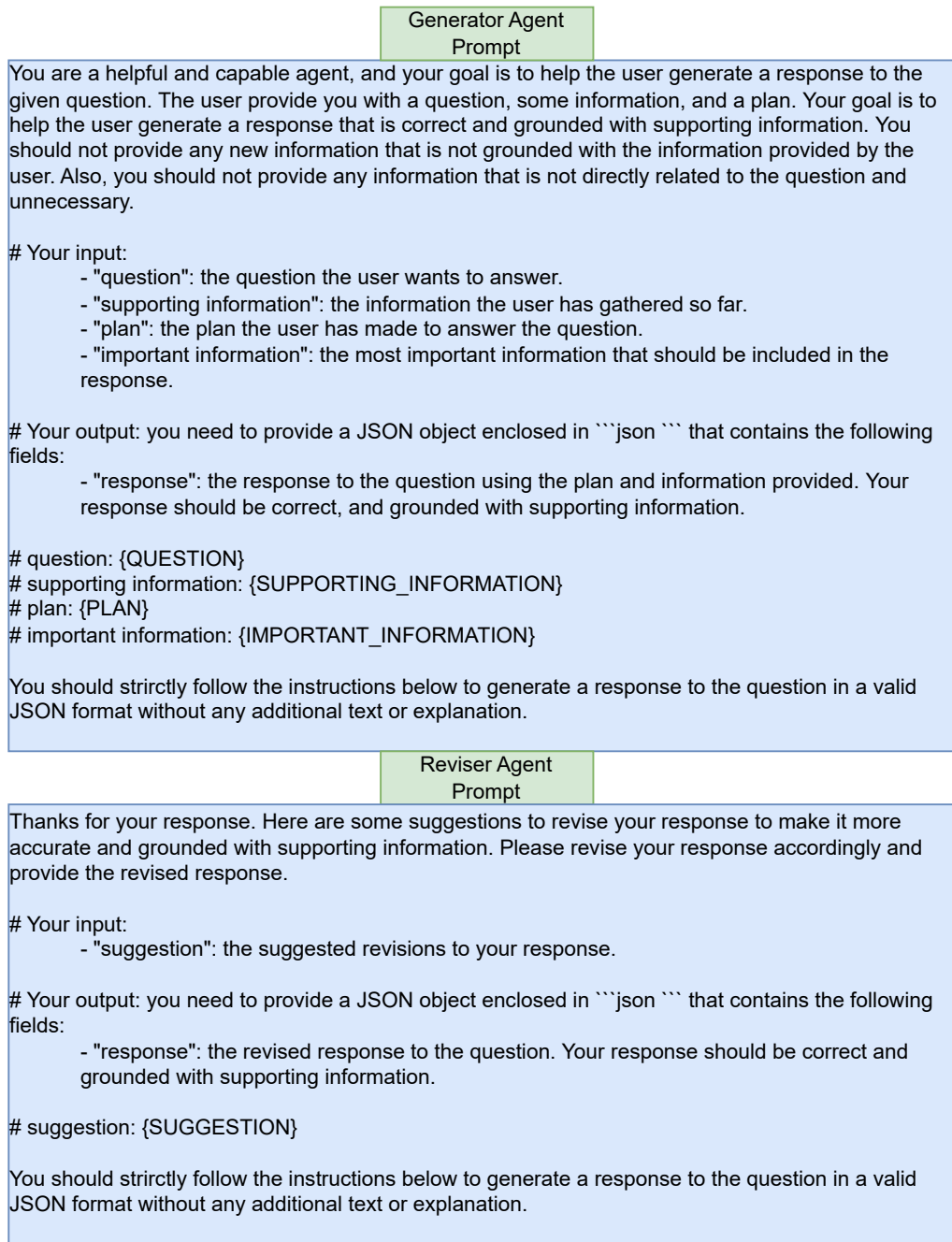


Figure 21: Prompt used for the Generator/Reviser agent in the mRAG framework.

multiple safety-related aspects of the process. The full interaction between agents in response to this question is presented in Table 2.

To address this question, the coordinator—serving as the entry point to the system—first invokes the planner agent to decompose the problem into a sequence of actionable steps. The planner breaks

the task down into identifying the key components of hydrogen steam reforming, gathering information on each component, searching for associated safety hazards, and synthesizing the collected information into a coherent response. Subsequently, the coordinator directs the searcher agent, providing suggestions to guide

Extracting Aspects for Coverage Reward

You are an impartial judge who has been asked to evaluate how the generated output answers the question. Your task is to given an expected output for a given question, extract all atomic pieces of information from the expected output. These atomic pieces of information are the smallest units of information that discuss a single unique aspect of the answer to the question.

Your input:

- "question": The question that was asked
- "expected_output": The expected output.

your output: you need to provide a JSON object enclosed in ``json`` that contains the following fields:

- "expected_output_aspects": a list of atomic pieces of information extracted from the expected output.

Each atomic piece of information is a json object with the following fields:

- "aspect": the atomic piece of information extracted from the expected output.
- "explanation": a brief explanation of why this atomic piece of information is important or relevant to the question.
- "evidence": the evidence from the expected output that supports this atomic piece of information.

question: {QUESTION}
expected_output: {EXPECTED_OUTPUT}

Your output should be a JSON object enclosed in ``json`` block in the given format.

Figure 22: Prompt used for extracting atomic aspects from the expected output for recall-oriented nugget correctness reward model.

Matching Aspects for Coverage Reward

You are an impartial judge who has been asked to evaluate how the generated output answers the question. You will be given a question, the expected output, a generated output, and a single aspect that you need to evaluate if the generated output contains the same information about from that aspect point of view. Your task is to compare the generated output and the expected output based on the given aspect. You need to provide a score between -1 and 2 that indicates the degree of match between the two outputs from the aspect point of view, using the following criteria:

scoring criteria:

- 2: Correct and relevant (no irrelevant information).
- 1: Correct but contains irrelevant information.
- 0: No answer provided (abstention).
- 1: Incorrect answer.

Your input:

- "question": The question that was asked
- "expected_output": The expected output.
- "generated_output": The generated output.
- "aspect": The aspect that you need to evaluate the generated output based on it, containing the following fields:
 - "aspect": the atomic piece of information extracted from the expected output.
 - "explanation": a brief explanation of why this atomic piece of information is important or relevant to the question.
 - "evidence": the evidence from the expected output that supports this atomic piece of information.

your output: you need to provide a JSON object enclosed in ``json`` that contains the following fields:

- "score": an int indicating the score you assign to the generated output based on the given aspect.
- "rationale": a str indicating the rationale behind your score.

question: {QUESTION}
expected_output: {EXPECTED_OUTPUT}
generated_output: {GENERATED_OUTPUT}
aspect: {ASPECT}

Your output should be a JSON object enclosed in ``json`` block in the given format.

Figure 23: Prompt used for matching atomic aspects between the expected output and generated response for recall-oriented nugget correctness reward model.

Extracting Aspects for Faithfulness Reward

You are an impartial judge who has been asked to evaluate how the generated output is faithful to the supporting documents provided. Your task is to given a generated output for a given question, extract all atomic pieces of information from the generated output. These atomic pieces of information are the smallest units of information that discuss a single unique aspect of the answer to the question.

Your input:

- "question": The question that was asked
- "answer": The generated answer.

your output: you need to provide a JSON object enclosed in ``json`` that contains the following fields:

- "answer_aspects": a list of atomic pieces of information extracted from the generated output. Each atomic piece of information is a json object with the following fields:
 - "id": the id of the atomic piece of information.
 - "aspect": the atomic piece of information extracted from the generated output.
 - "explanation": a brief explanation of why this atomic piece of information is important or relevant to the question.
 - "evidence": the evidence from the generated output that supports this atomic piece of information.

question: {QUESTION}
answer: {ANSWER}

Figure 24: Prompt used for extracting atomic aspects from the generated output for faithfulness reward.

Matching Aspects for Faithfulness Reward

You are an impartial judge who has been asked to evaluate how well the generated answer is grounded in the supporting knowledge provided based on the given criteria. You will be given a list of atomic aspects from the generated output and a list of supporting documents. Your task is to go through each extracted atomic aspect and check if its evidence from the generated output is grounded in the supporting documents. You need to provide a score between -1 and 1 that indicates the degree of match between the two outputs from the aspect point of view, using the following criteria:

1: Full support. All answer parts are grounded in the supporting documents.
0: Partial support. Not all answer parts are grounded in the supporting documents.
-1: No support. All answer parts are not grounded in the supporting documents.

Your input:

- "question": The question that was asked
- "answer_aspects": a list of atomic pieces of information extracted from the generated output. Each atomic piece of information is a json object with the following fields:
 - "id": the id of the atomic piece of information.
 - "aspect": the atomic piece of information extracted from the generated output.
 - "explanation": a brief explanation of why this atomic piece of information is important or relevant to the question.
 - "evidence": the evidence from the generated output that supports this atomic piece of information.
- "context": The a list of supporting documents that you should check the answer based on.

your output: you need to provide a JSON object enclosed in ``json`` that contains the following fields:

- "scores_list": a list of scores for each atomic aspect, indicating the score you assign to the generated answer based on the given criteria. Each object in the list should be a valid json object that contain the following fields:
 - "id": the id of the atomic piece of information.
 - "score": an int indicating the score you assign to the generated answer based on the given criteria.
 - "rationale": a str indicating the rationale behind your score.

question: {QUESTION}
answer_aspects: {ASPECT}
context: {CONTEXT}

Figure 25: Prompt used for matching aspects between the generated output and retrieved documents for faithfulness reward.

Algorithm 2 Implementation of the Searcher agent in the mRAG framework.**Input:** question q , context information c , suggested search aspects a , retrieval model R , LLM π **Output:** relevant retrieved documents set S

```

1:  $S = \{\}$ 
2:  $i = 0$ 
3:  $end\_search = False$ 
4:  $query = \pi(q, c, a)$ 
5: while  $i < MAX\_STEPS$  and  $end\_search = False$  do
6:    $this\_step\_docs = R(query, 2)$ 
7:   for  $d \in this\_step\_docs$  do
8:     if  $\pi(query, d) = relevant$  then
9:        $S = S \cup \{d\}$ 
10:    end if
11:  end for
12:   $i = i + 1$ 
13:   $change\_query = \pi(query, this\_step\_docs, S)$ 
14:   $end\_search = \pi(query, this\_step\_docs, S)$ 
15:  if  $change\_query = True$  then
16:     $query = \pi(query, this\_step\_docs, S)$ 
17:  end if
18: end while
19: return  $S$ 

```

▶ Initializing the relevant documents set
 ▶ Counter of number of retrieval steps
 ▶ retrieve two docs for query. If this query is issued before, return next 2 docs for query.
 ▶ For each document in the retrieved document for the query
 ▶ If document is relevant to the query
 ▶ Add the document to the relevant documents set
 ▶ Updating the step
 ▶ Check if we need to change the query
 ▶ Check if we need to end the search process
 ▶ Updating the search query if it needs update
 ▶ Returning the relevant retrieved documents

the retrieval of information related to the specified aspects. The searcher then performs targeted queries and returns relevant information accordingly. After analyzing the safety aspects of the process using the reasoner agent, the coordinator instructs the searcher to gather information about real-world historical safety incidents related to hydrogen steam reforming. The searcher employs the retrieval model to obtain relevant documents and returns the collected data to the coordinator. Given the volume of retrieved information, the coordinator then invokes the summarizer agent to condense the content into a coherent form. Subsequently, the coordinator calls the generator agent, providing the aggregated evidence and instructing it to generate a comprehensive response. The generator is specifically guided to include a detailed overview of the key components involved in hydrogen steam reforming, the associated safety hazards for each component, documented historical incidents, and current safety guidelines and regulations. Once the response is generated, the coordinator consults the validator agent to verify whether the response meets the defined criteria. The validator emphasizes the importance of completeness and depth, particularly due to the safety-critical nature of the query. After confirming that all criteria are satisfied, the coordinator ends the process and returns the response.

Retrieving more information to make sure enough information is collected. In addition to cases that require integrating multiple aspects, some queries present challenges in retrieving relevant information even for a single aspect, often due to limitations in the retrieval model. In such situations, a robust RAG system must be capable of detecting retrieval failures and proactively reformulating search queries to obtain more relevant content. To demonstrate this capability, we examine the system’s response to the query: “How did John Ball influence golf history at Hoylake, and what strategic

challenges does the course present regarding out of bounds?”, as illustrated in Table 3.

To address this question, the coordinator first invokes the planner agent to generate a structured plan. The resulting plan includes the following steps: identifying John Ball’s role in golf history, retrieving key events or milestones in his career at Hoylake, and examining the evolution and impact of out-of-bounds (OB) rules on gameplay through historical records and expert analysis. Guided by this plan, the coordinator then tasks the searcher agent with retrieving information about John Ball’s influence on golf history at Hoylake, which is successfully completed. Subsequently, the reasoner agent is called to analyze John Ball’s contributions. During this reasoning process, it identifies the need to gather additional information regarding the strategic challenges of the Hoylake course. However, the initial retrieval attempt fails to yield relevant content. To address this, the searcher reformulates the query to focus on the “strategic challenges of Hoylake Royal Liverpool Golf Club,” which successfully retrieves pertinent information. The coordinator then invokes the generator agent to produce a comprehensive response using the collected evidence. As in the previous case study, this response is passed to the validator agent for evaluation. Once all criteria are satisfied, the coordinator concludes the process and returns the final response.

Algorithm 3 Implementation of the correctness reward model.

Input: question q , generated response r , ground truth response gt , evaluator LLM π **Output:** correctness score s_f

- 1: $extracted_aspects = \pi(q, gt)$ ▷ Extracting the atomic aspects from the ground truth output
 - 2: $S_f = 0$
 - 3: **for** $aspect \in extracted_aspects$ **do** ▷ For each of the extracted aspects from the ground truth response
 - 4: $S_{aspect} = \frac{\pi(q, aspect, r, gt) + 1}{3}$ ▷ Score the aspect using the LLM in range of -1 and 2, then normalize it to 0 and 1
 - 5: $S_f = S_f + \frac{S_{aspect}}{|extracted_aspects|}$ ▷ Adding the score of aspect to the final score
 - 6: **end for**
 - 7: **return** s_f ▷ Returning the correctness score for the generated response
-

Algorithm 4 Implementation of the Faithfulness reward model.

Input: question q , generated response r , retrieved documents C , evaluator LLM π **Output:** faithfulness score s_f

- 1: $extracted_aspects = \pi(q, r)$ ▷ Extracting the atomic aspects from the generated output
 - 2: $S_f = 0$
 - 3: **for** $aspect \in extracted_aspects$ **do** ▷ For each of the extracted aspects from the generated response
 - 4: $S_{aspect} = \frac{\pi(q, aspect, r, C) + 1}{2}$ ▷ Score the aspect using the LLM in range of -1 and 1, then normalize it to 0 and 1
 - 5: $S_f = S_f + \frac{S_{aspect}}{|extracted_aspects|}$ ▷ Adding the score of aspect to the final score
 - 6: **end for**
 - 7: **return** s_f ▷ Returning the faithfulness score for the generated response
-

Table 2: First case study of mRAG on the question: safety concerns hydrogen steam reforming

Agent	Conversation
Coordinator	<p>User: {"question": "safety concerns hydrogen steam reforming"}</p> <hr/> <p>LLM: {"agent": "planner", <input": ""},<br="" "information":="" "safety="" concerns="" hydrogen="" reforming",="" steam="" {"question":=""></input":> "reason": "I will use the planner agent to create a strategy for gathering and processing information related to safety concerns associated with hydrogen steam reforming." }</p>
Planner	<p>User (Coordinator): {"question": "safety concerns hydrogen steam reforming";"information": ""}</p> <hr/> <p>LLM (Planner): { "plan": ["Identify the key components of hydrogen steam reforming process", "Research potential safety hazards associated with each component", "Gather data on accidents or incidents related to hydrogen steam reforming from reliable sources such as industry reports, academic journals, and government databases", "Review existing safety guidelines and regulations for hydrogen steam reforming from organizations like OSHA, ISO, or relevant national agencies", "Analyze the risk factors and propose mitigation strategies for identified safety concerns", "Summarize findings and present a comprehensive overview of safety concerns and recommended measures"] }</p>
Coordinator	<p>User(Planner): "Output from Planner [Avoiding Repetition]"</p> <hr/> <p>LLM: {"agent": "searcher", "input": { "question": "safety concerns hydrogen steam reforming", "information": "", "suggestions": ["key components of hydrogen steam reforming process";"potential safety hazards associated with each component";"accidents or incidents related to hydrogen steam reforming";"existing safety guidelines and regulations for hydrogen steam reforming"]}, "reason": "The searcher agent will help me find relevant information about the key components of hydrogen steam reforming, potential safety hazards, historical incidents, and current safety guidelines." }</p>

Searcher	<p>User (Coordinator): "Output from Coordinator [Avoiding Repetition]"</p> <hr/> <p>LLM (Searcher): {"search_query": "'key components of hydrogen steam reforming process", "search_query_explanation": "To understand the safety concerns of hydrogen steam reforming, it is important to first identify its key components as these components are likely to pose potential safety hazards."}</p> <hr/> <p>User (Retriever): This is the information resulted from your search query: Query ID: 0 Documnet 1 ID: 552 Document 1 text: Product gas from the steam reforming of the methane and naphtha contains equilibrium amounts of hydrogen,... Documnet 2 ID: 352 Document 2 text: A final process stream is produced by steam reforming of residual... Now I need you to verify the information. Be as objective as possible in your verification. Note you should strictly follow the given format and do not add any extra explanation or informati- on to your output.</p> <hr/> <p>LLM (Searcher): {"query_id": 0, "relevance": [{"doc_id": 552, "is_relevant": true, "is_relevant_explanation": "This document provides detailed information about the key components of the hydrogen steam reforming process, including the reformer, the temperature conditions, and the water-shift gas reactions. It is highly relevant to understanding the safety concerns of the process."}, {"doc_id": 352, "is_relevant": false, "is_relevant_explanation": "This document focuses on the advantages and economic benefits of the steam reforming process but does not provide specific details about the key components or potential safety hazards. Therefore, it is less relevant to the current query."}], "change_search_query": false, "change_search_query_explanation": "The initial search query yielded relevant documents that provide detailed information about the key components of the hydrogen steam reforming process. No immediate need to change the search query.", "new_search_query": "", "end_search": false, "end_search_explanation": "More information is needed to fully address the safety concerns of the hydrogen steam reforming process. Additional searches may be necessary to find details on potential safety hazards and existing safety guidelines."}</p> <hr/> <p>User (Retriever): This is the information resulted from your search query: Query ID: 0 Documnet 1 ID: 832 Document 1 text: Steam Methane Reforming one of the greatest challenges in today's 2019s industries... Documnet 2 ID: 915 Document 2 text: 5,082,481, Membrane Separation Process for Cracked Gases, describes... Now I need you to verify the information. Be as objective as possible in your verification. Note you should strictly follow the given format and do not add any extra explanation or information to your output.</p> <hr/> <p>LLM (Searcher): {"query_id": 0, "relevance": [{"doc_id": 832, "is_relevant": true, "is_relevant_explanation": "This document provides detailed information about the steam methane reforming process, including the temperature and pressure conditions, the role of reformer tubes, and the importance of the process in hydrogen production. It is highly relevant to understanding the safety concerns of the process."}, {"doc_id": 915, "is_relevant": false, "is_relevant_explanation": "This document discusses various membrane technologies for hydrogen recovery but does not provide specific details about the key components or potential safety hazards of the steam methane reforming process. Therefore, it is less relevant to the current query."}],</p>
-----------------	--

<p>Searcher (Contd.)</p>	<p>"change_search_query": false, "change_search_query_explanation": "The initial search query yielded relevant documents that provide detailed information about the steam methane reforming process. No immediate need to change the search query.", "new_search_query": "", "end_search": false, "end_search_explanation": "More information is needed to fully address the safety concerns of the steam methane reforming process. Additional searches may be necessary to find details on potential safety hazards and existing safety guidelines.}"</p> <hr/> <p>User (Retriever): This is the information resulted from your search query: Query ID: 0 Documnet 1 ID: 189 Document 1 text: The reformer furnace design alternatives will be discussed below. Feed gas 2014 usually a mixture of hydrogen, methane and other light hydrocarbons 2014 is first compressed to about 300 psig.... Documnet 2 ID: 254 Document 2 text: BRIEF DESCRIPTION OF THE DRAWINGS These and other objects, features and advantages of the present invention will become better understood with regard to the following description, pending claims and accompanying drawings where:.... Now I need you to verify the information. Be as objective as possible in your verification. Note you should strictly follow the given format and do not add any extra explanation or information to your output.</p> <hr/> <p>LLM (Searcher): { "query_id": 0, "relevance": [{"doc_id": 189, "is_relevant": true, "is_relevant_explanation": "This document provides detailed information about the key components of the hydrogen steam reforming process, including the preheating, compression, and heating stages, as well as the role of reformer tubes and furnace designs. It is highly relevant to understanding the safety concerns of the process."}, {"doc_id": 254, "is_relevant": false, "is_relevant_explanation": "This document focuses on a novel pre-reforming and reforming process configuration involving membranes for hydrogen gas separation and recovery. While it mentions the steam methane reforming process, it does not provide specific details about the key components or potential safety hazards. Therefore, it is less relevant to the current query."}], "change_search_query": false, "change_search_query_explanation": "The initial search query yielded relevant documents that provide detailed information about the key components of the hydrogen steam reforming process. No immediate need to change the search query.", "new_search_query": "", "end_search": false, "end_search_explanation": "More information is needed to fully address the safety concerns of the hydrogen steam reforming process. Additional searches may be necessary to find details on potential safety hazards and existing safety guidelines.}"</p> <hr/> <p>User (Retriever): This is the information resulted from your search query: Query ID: 0 Documnet 1 ID: 985 Document 1 text: Minimizing the hydrogen bled into the fuel gas can keep the hydrogen production levels manageable. However, the hydrogen plant feed usually includes some hydrogen that goes through for a 201cfree ride,... Documnet 2 ID: 243 Document 2 text: BRIEF DESCRIPTION OF THE DRAWINGS These and other objects, features and advantages of the present invention will become better understood with regard to the following description, pending claims and accompanying drawings where:....</p>
---------------------------------	---

Searcher (Contd.)	<p>Now I need you to verify the information. Be as objective as possible in your verification. Note you should strictly follow the given format and do not add any extra explanation or information to your output....</p> <hr/> <p>LLM (Searcher): { "query_id": 0, "relevance": [{"doc_id": 189, "is_relevant": true, "is_relevant_explanation": "This document provides detailed information about the key components of the hydrogen steam reforming process, including the preheating, compression, and heating stages, as well as the role of reformer tubes and furnace designs. It is highly relevant to understanding the safety concerns of the process."}, {"doc_id": 254, "is_relevant": false, "is_relevant_explanation": "This document focuses on a novel pre-reforming and reforming process configuration involving membranes for hydrogen gas separation and recovery. While it mentions the steam methane reforming process, it does not provide specific details about the key components or potential safety hazards. Therefore, it is less relevant to the current query."}], "change_search_query": false, "change_search_query_explanation": "The initial search query yielded relevant documents that provide detailed information about the key components of the hydrogen steam reforming process. No immediate need to change the search query.", "new_search_query": "", "end_search": false, "end_search_explanation": "More information is needed to fully address the safety concerns of the hydrogen steam reforming process. Additional searches may be necessary to find details on potential safety hazards and existing safety guidelines.}"</p> <hr/> <p>User (Retriever): This is the information resulted from your search query: Query ID: 0 Documnet 1 ID: 174 Document 1 text: Gas enters the bottom of the absorber, where it contacts lean solution. The carbon dioxide is absorbed from the gas, leaving the rest of the contaminants and hydrogen relatively untouched.... Documnet 2 ID: 769 Document 2 text: From our extensive range of gas production machinery, we manufacture, export and supply supreme quality Hydrogen Gas Generator.... Now I need you to verify the information. Be as objective as possible in your verification. Note you should strictly follow the given format and do not add any extra explanation or information to your output.</p> <hr/> <p>LLM (Searcher): { "query_id": 0, "relevance": [{"doc_id": 174, "is_relevant": false, "is_relevant_explanation": "This document focuses on the PSA (Pressure Swing Adsorption) process for hydrogen purification and does not provide specific details about the key components or potential safety hazards of the steam methane reforming process. Therefore, it is less relevant to the current query."}, {"doc_id": 769, "is_relevant": false, "is_relevant_explanation": "This document describes a hydrogen gas generator and its features but does not provide specific details about the key components or potential safety hazards of the steam methane reforming process. Therefore, it is less relevant to the current query."}], "change_search_query": true, "change_search_query_explanation": "The documents provided do not contain specific details about the key components or potential safety hazards of the steam methane reforming process. A new search query focusing on safety concerns and potential hazards is needed.", "new_search_query": "potential safety hazards associated with steam methane reforming", "end_search": false, "end_search_explanation": "More information is needed to fully address the safety concerns of the steam methane reforming process. A new search query focusing on potential safety hazards is necessary." }</p>
--------------------------	--

<p>Searcher (Contd.)</p>	<p>User (Retriever): This is the information resulted from your search query: Query ID: 0 Documnet 1 ID: 124 Document 1 text: Gas enters the bottom of the absorber, where it contacts lean solution. The carbon dioxide is absorbed from the gas, leaving the rest of the contaminants and hydrogen relatively untouched..... Documnet 2 ID: 832 Document 2 text: Steam Methane Reforming one of the greatest challenges in today's industries is an increased demand for hydrogen. Due to its ever-increasing industrial relevance, there is a need for efficient methods of hydrogen production, which has traditionally been done at a high temperature.... Now I need you to verify the information. Be as objective as possible in your verification. Note you should strictly follow the given format and do not add any extra explanation or information to your output.</p> <hr/> <p>LLM (Searcher): { "query_id": 1, "relevance": [{"doc_id": 124, "is_relevant": true, "is_relevant_explanation": "This document provides detailed information about the potential safety hazards during the startup procedure of steam methane reformers, including the risk of overheating and melting of reformer tubes. It is highly relevant to understanding the safety concerns of the process."}, {"doc_id": 832, "is_relevant": false, "is_relevant_explanation": "This document focuses on the general description of the steam methane reforming process and its importance but does not provide specific details about potential safety hazards. Therefore, it is less relevant to the current query:"}], "change_search_query": false, "change_search_query_explanation": "The initial search query yielded relevant documents that provide detailed information about the potential safety hazards during the steam methane reforming process. No immediate need to change the search query.", "new_search_query": "", "end_search": true, "end_search_explanation": "The provided information is sufficient to address the safety concerns of the steam methane reforming process. Further searches are not necessary." }</p>
<p>Coordinator</p>	<p>User (Searcher): "only the relevant documents found by searcher"</p> <hr/> <p>LLM : { "agent": "reasoner", "input": {"question": "safety concerns hydrogen steam reforming", "information": "Steam methane reforming is a chemical process by which methane is converted to hydrogen. It involves the reaction between natural gas, which contains methane and steam, at a temperature of 700-1000°C and under a pressure of 3-25 bar. The products of this chemical process are initially hydrogen and carbon monoxide. The carbon monoxide is further reacted with steam in a water-gas shift reaction to produce more hydrogen and carbon dioxide. Impurities are then removed from the system using pressure-swing adsorption. As a result, pure hydrogen is isolated and extracted. This process is mediated by reformer tubes also known as catalyst tubes used in steam reformer furnaces known as steam methane reformers. Minimizing the hydrogen bleed into the fuel gas can keep the hydrogen production levels manageable. However, the hydrogen plant feed usually includes some hydrogen that goes through for a free ride, except for the cost of heating it up to reformer temperature and ultimately cooling it back down to recovery level."}, "aspect": "Key components of hydrogen steam reforming process"}, "reason": "I will use the reasoner agent to analyze the key components of the hydrogen steam reforming process and identify potential safety concerns associated with each component." }</p>

Reasoner	<p>User (Coordinator): Output from Coordinator [Avoiding Repetition]</p> <hr/> <p>LLM (Reasoner): {"analysis": "Hydrogen steam reforming involves multiple key components including the chemical reaction between methane and steam, the use of reformer tubes or catalyst tubes, and the subsequent purification steps.", "The primary chemical reaction occurs in the steam reformer furnace, where methane reacts with steam at high temperatures (700-1000 u00b0C) and pressures (3-25 bar) to produce hydrogen and carbon monoxide.", "Reformer tubes or catalyst tubes play a crucial role in facilitating the chemical reaction. These tubes house the catalyst necessary for the steam reforming process, enhancing the rate of hydrogen production.", "After the initial reaction, a water-gas shift reaction takes place to further convert carbon monoxide into additional hydrogen and carbon dioxide, improving the purity of the hydrogen output.", "Purification steps such as pressure-swing adsorption are employed to remove impurities from the hydrogen stream, ensuring the final product is of high quality and suitable for various applications.", "Minimizing hydrogen bleed during the process helps maintain efficiency, but some hydrogen inevitably remains in the feed stream, requiring energy to heat and cool."}</p>
Coordinator	<p>User (Reasoner): "Output from Reasoner [Avoiding Repetition]"</p> <hr/> <p>LLM : { "agent": "reasoner", <input": "safety="" concerns="" hydrogen="" reforming",<br="" steam="" {"question":=""></input":> "information": "Hydrogen steam reforming involves multiple key components including the chemical reaction between methane and steam, the use of reformer tubes or catalyst tubes, and the subsequent purification steps. The primary chemical reaction occurs in the steam reformer furnace, where methane reacts with steam at high temperatures (700-1000u00b0C) and pressures (3-25 bar) to produce hydrogen and carbon monoxide. Reformer tubes or catalyst tubes play a crucial role in facilitating the chemical reaction. After the initial reaction, a water-gas shift reaction takes place to further convert carbon monoxide into additional hydrogen and carbon dioxide, improving the purity of the hydrogen output. Purification steps such as pressure-swing adsorption are employed to remove impurities from the hydrogen stream, ensuring the final product is of high quality and suitable for various applications. Minimizing hydrogen bleed during the process helps maintain efficiency, but some hydrogen inevitably remains in the feed stream, requiring energy to heat and cool."}, "aspect": "Potential safety hazards associated with each component"}, "reason": "I will use the reasoner agent to identify potential safety hazards associated with each key component of the hydrogen steam reforming process."}</p>

Reasoner	<p>User (Coordinator): Output from Coordinator [Avoiding Repetition]</p> <hr/> <p>LLM (Reasoner): {"analysis": "The steam reformer furnace is a critical component as it operates at extremely high temperatures (700-1000 u00b0C). High temperatures pose potential risks such as thermal stress, cracking, and potential leaks of flammable gases like methane, leading to fire or explosion hazards.", "Reformer tubes or catalyst tubes are another key component. These tubes contain reactive materials that can degrade over time due to exposure to high temperatures and corrosive environments, potentially leading to tube failure and gas leaks. Additionally, the high-pressure environment (3-25 bar) increases the severity of any leaks, posing significant safety risks.", "The water-gas shift reactor, although not explicitly mentioned in the provided information, is an important step in the process. It involves the reaction of carbon monoxide with steam at around 300-400 u00b0C to produce more hydrogen. High temperatures and the presence of reactive gases increase the risk of equipment failure and gas leaks, which could lead to inhalation of toxic gases or fires.", "Pressure-swing adsorption (PSA) is a purification step that uses adsorbents to remove impurities from the hydrogen stream. During the adsorption and desorption cycles, there is a risk of overpressure or vacuum conditions, which can damage the PSA equipment and lead to gas leaks. Furthermore, the handling of adsorbents can pose health risks if inhaled or contacted directly.", "Overall, the high temperatures, pressures, and flammable gases involved in the hydrogen steam reforming process mean that all components must be carefully designed, maintained, and monitored to ensure safety. Any failure in these systems can lead to serious safety hazards."}</p>
Coordinator	<p>User (Reasoner): Output from Reasoner [Avoiding Repetition]</p> <hr/> <p>LLM: {"agent": "searcher", <input": "safety="" concerns="" hydrogen="" reforming",<br="" steam="" {"question":=""></input":> "information": "The steam reformer furnace is a critical component as it operates at extremely high temperatures (700-1000u00b0C). Reformer tubes or catalyst tubes are another key component. The water-gas shift reactor involves high temperatures and the presence of reactive gases. Pressure-swing adsorption (PSA) equipment handles overpressure and vacuum conditions", "suggestions": ["Historical incidents related to hydrogen steam reforming","Existing safety guidelines and regulations for hydrogen steam reforming", "Mitigation strategies for identified safety concerns"], "reason": "The searcher agent will help me find historical incidents, existing safety guidelines, and mitigation strategies related to hydrogen steam reforming to provide a comprehensive overview of safety concerns and recommended measures."}</p>

Searcher	<p>LLM (Searcher): {"search_query": "historical incidents related to hydrogen steam reforming", "search_query_explanation": "Historical incidents can provide valuable insights into the types of safety issues that have occurred in the past, which can help identify current safety concerns and potential mitigation strategies."}</p> <hr/> <p>User (Retriever): This is the information resulted from your search query: Query ID: 2 Documnet 1 ID: 785 Document 1 text: Fuel refineries use a series of chemical reactions to arrive at the final product. Hydrogen fluoride is used in the oil refining process and causes a reaction, called alkylolation, to make fuel from unrefined oil.... Documnet 2 ID: 200 Document 2 text: Fig. 4: Mass fractions of the species (ethanol, CO2 and H2O) along the reformer axis. The mass fractions of the ethanol and steam decay along the reformer axis. The ethanol conversion is 80.3%..... Now I need you to verify the information. Be as objective as possible in your verification. Note you should strictly follow the given format and do not add any extra explanation or information to your output.</p> <hr/> <p>LLM (Searcher): { "query_id": 2, "relevance": [{"doc_id": 785, "is_relevant": true, "is_relevant_explanation": "This document provides a historical incident related to a refinery accident involving hydrofluoric acid, which can be linked to safety concerns in steam reforming processes. It highlights the potential for catastrophic impacts and the importance of safety protocols."}, {"doc_id": 200, "is_relevant": false, "is_relevant_explanation": "This document discusses the mass fractions of species in a steam reformer and safety concerns related to hydrogen integrity in steel tubes. While it provides technical details, it does not directly address historical incidents or safety guidelines specific to steam reforming. Therefore, it is less relevant to the current query."}], "change_search_query": false, "change_search_query_explanation": "The initial search query yielded relevant documents that provide a historical incident related to a refinery accident involving hydrofluoric acid, which can be linked to safety concerns in steam reforming processes. No immediate need to change the search query.", "new_search_query": "", "end_search": true, "end_search_explanation": "The provided information is sufficient to address the safety concerns of the steam methane reforming process. Further searches are not necessary." }</p>
-----------------	--

<p>Coordinator</p>	<p>User (Searcher): "only the relevant documents found by searcher"</p> <hr/> <p>LLM (Coordinator): {"agent": "summarizer", <input": "safety="" concerns="" hydrogen="" reforming",<br="" steam="" {"question":=""></input":> "information": ""The steam reformer furnace is a critical component as it operates at extremely high temperatures (700-10000b0C). Reformer tubes or catalyst tubes are another key component. The water-gas shift reactor involves high temperatures and the presence of reactive gases. Pressure-swing adsorption (PSA) equipment handles overpressure and vacuum conditions. Historical incidents related to hydrogen steam reforming include an explosion at an ExxonMobil refinery in Torrance, California, where an explosion occurred due to a leak of hydrocarbons into an electrostatic precipitator (ESP) device, causing sparks and igniting the hydrocarbons. This resulted in a dust cloud extending for miles and projected large debris around the site. The incident highlighted the importance of proper evaluation of process hazards, implementation of safeguards, and adherence to safety precautions."} "reason": "I will use the summarizer agent to create a concise summary of the key points regarding the safety concerns associated with hydrogen steam reforming based on the gathered information."}</p>
<p>Summarizer</p>	<p>User (Coordinator): Output from Coordinator and relevant retrieved documents [Avoiding Repetition]</p> <hr/> <p>LLM (Summarizer): {"summary": "Hydrogen steam reforming involves critical components like high-temperature furnaces (700-1000 u00b0C), reformer/tubes, and water-gas shift reactors. PSA equipment manages overpressure/vacuum conditions. Historical incidents, such as an explosion at ExxonMobil u2019s Torrance refinery, underscore the need for thorough hazard assessments, robust safety measures, and strict adherence to safety protocols."}</p>
<p>Coordinator</p>	<p>User (Answerer): Output from Summarizer [Avoiding Repetition]</p> <hr/> <p>LLM:{"agent": "Answerer", <input": "safety="" concerns="" hydrogen="" reforming",<br="" steam="" {"question":=""></input":> "guidance": "Provide a detailed overview of the key components of hydrogen steam reforming, potential safety hazards associated with each component, historical incidents, and existing safety guidelines and regulations.";important_information": "High-temperature furnaces (700-10000b0C), reformer/tubes, water-gas shift reactors, PSA equipment, historical incidents, and safety guidelines."} "reason": "I will use the answerer agent to generate a comprehensive response to the question about safety concerns in hydrogen steam reforming, incorporating the key components, potential hazards, historical incidents, and safety guidelines."},</p>
<p>Answerer</p>	<p>User (Coordinator): Output from Coordinator [Avoiding Repetition]</p> <hr/> <p>LLM: {"response": "Hydrogen steam reforming is a critical process in the production of hydrogen, involving several key components such as high-temperature furnaces, reformer/tubes, water-gas shift reactors, and PSA equipment. Each component presents specific safety concerns. High-temperature furnaces, operating at 700-1000 u00b0C, pose risks of thermal burns and potential tube failures due to excessive temperatures. Reformer/tubes are susceptible to coking and require careful monitoring to prevent catalyst degradation. Water-gas shift reactors, where the exothermic reaction occurs, need proper cooling systems to manage temperature increases. PSA equipment, used for hydrogen purification, must be handled with care to avoid pressure-related accidents. Historical incidents, such as the ExxonMobil refinery explosion involving hydrofluoric acid, highlight the importance of stringent safety measures. Safety guidelines emphasize the need for proper monitoring, regular maintenance, and adherence to operating procedures to mitigate risks. Regulations require thorough hazard assessments and the implementation of safeguards to prevent accidents. Understanding these components and their associated safety concerns is crucial for the safe operation of hydrogen steam reforming facilities." }</p>

Coordinator	<p>User (Answerer): Output from Answerer [Avoiding Repetition]</p> <hr/> <p>LLM: {"agent": "validator", "input": { "question": "safety concerns hydrogen steam reforming" "information": "Hydrogen steam reforming is a critical process in the production of hydrogen, involving several key components such as high-temperature furnaces, reformer/tubes, water-gas shift reactors, and PSA equipment. Each component presents specific safety concerns. High-temperature furnaces, operating at 700-100000b0C, pose risks of thermal burns and potential tube failures due to excessive temperatures. Reformer/tubes are susceptible to coking and require careful monitoring to prevent catalyst degradation. Water-gas shift reactors, where the exothermic reaction occurs, need proper cooling systems to manage temperature increases. PSA equipment, used for hydrogen purification, must be handled with care to avoid pressure-related accidents. Historical incidents, such as the ExxonMobil refinery explosion involving hydrofluoric acid, highlight the importance of stringent safety measures. Safety guidelines emphasize the need for proper monitoring, regular maintenance, and adherence to operating procedures to mitigate risks. Regulations require thorough hazard assessments and the implementation of safeguards to prevent accidents. Understanding these components and their associated safety concerns is crucial for the safe operation of hydrogen steam reforming facilities."}, "response": "Hydrogen steam reforming is a critical process in the production of hydrogen, involving several key components such as high-temperature furnaces, reformer/tubes, water-gas shift reactors, and PSA equipment. Each component presents specific safety concerns. High-temperature furnaces, operating at 700-100000b0C, pose risks of thermal burns and potential tube failures due to excessive temperatures. Reformer/tubes are susceptible to coking and require careful monitoring to prevent catalyst degradation. Water-gas shift reactors, where the exothermic reaction occurs, need proper cooling systems to manage temperature increases. PSA equipment, used for hydrogen purification, must be handled with care to avoid pressure-related accidents. Historical incidents, such as the ExxonMobil refinery explosion involving hydrofluoric acid, highlight the importance of stringent safety measures. Safety guidelines emphasize the need for proper monitoring, regular maintenance, and adherence to operating procedures to mitigate risks. Regulations require thorough hazard assessments and the implementation of safeguards to prevent accidents. Understanding these components and their associated safety concerns is crucial for the safe operation of hydrogen steam reforming facilities."}, "reason": "I will use the validator agent to check if the generated response meets the criteria for the given question."}</p>
Validator	<p>User (Coordinator): Output from Coordinator [Avoiding Repetition]</p> <hr/> <p>LLM: {"extracted_criteria": [{"criteria": "detailed", "criteria_explanation": "The question asks about safety concerns related to hydrogen steam reforming, which requires a detailed explanation.", "is_response_valid": true, "is_response_valid_feedback": "The response is detailed and covers various aspects of safety concerns in hydrogen steam reforming."}], "is_groundedly_supported": true, "is_groundedly_supported_feedback": "The response is well-grounded with supporting information, including specific components, potential risks, and historical incidents.", "is_correctly_answered": true, "is_correctly_answered_feedback": "The response correctly addresses the safety concerns of hydrogen steam reforming and provides relevant details."}</p>
Coordinator	<p>User (Validator): Output from Validator [Avoiding Repetition]</p> <hr/> <p>LLM: {"agent": "finisher", "input": {"finished": true}}</p>

Table 3: Second case study of mRAG on the question: How did John Ball influence golf history at Hoylake, and what strategic challenges does the course present regarding out of bounds?

Agent	Conversation
Coordinator	<p>User: {"question": "How did John Ball influence golf history at Hoylake, and what strategic challenges does the course present regarding out of bounds?"}</p> <hr/> <p>LLM: {"agent": "planner", "input": {"question": "How did John Ball influence golf history at Hoylake, and what strategic challenges does the course present regarding out of bounds?", "information": ""}, "reason": "I need to create a plan to gather and organize the necessary information to answer the question." }</p>
Planner	<p>User (Coordinator): {"question": "How did John Ball influence golf history at Hoylake, and what strategic challenges does the course present regarding out of bounds?" "information": "" }</p> <hr/> <p>LLM (Planner): { "plan": ["Research John Ball's role in golf history, focusing on his connection to Hoylake Golf Club.", "Identify key events or moments in John Ball's career at Hoylake that significantly impacted golf history.", "Analyze the current layout and characteristics of the Hoylake course, particularly its out of bounds (OB) rules and challenges it presents to players.", "Consult historical records and expert opinions to understand how these OB rules have evolved over time and their impact on gameplay.", "Compile findings into a coherent narrative that highlights John Ball's influence and the strategic challenges posed by the course."] }</p>
Coordinator	<p>User(Planner): "Output from Planner [Avoiding Repetition]"</p> <hr/> <p>LLM: {"agent": "searcher", "input": "question": "How did John Ball influence golf history at Hoylake, and what strategic challenges does the course present regarding out of bounds?", "information": "", "suggestions": ["John Ball", "Hoylake Golf Club", "history of golf at Hoylake", "out of bounds rules in golf"], "reason": "The next step is to find relevant information about John Ball's connection to Hoylake and the characteristics of the Hoylake course."}</p>

<p>Searcher</p>	<p>User (Coordinator): "Output from Coordinator [Avoiding Repetition]"</p> <hr/> <p>LLM (Searcher): {"search_query": "John Ball influence on Hoylake golf history", "search_query_explanation": "This search query aims to find information specifically related to how John Ball influenced golf history at Hoylake, which directly addresses part of the question."}</p> <hr/> <p>User (Retriever): This is the information resulted from your search query: Query ID: 0 Documnet 1 ID: 224 Document 1 text: United Kingdom: Grant Books, 2001. 81p. half leather, slipcase. Limited edition of 95 subscribers copies. Hoylake, otherwise known as The Royal Liverpool Golf Club is one of golf’s most storied venues.... Documnet 2 ID: 705 Document 2 text: Ball dominated amateur golf in Great Britain. He won all the important golf championships as well as the hearts and respect of his country.... Now I need you to verify the information. Be as objective as possible in your verification. Note you should strictly follow the given format and do not add any extra explanation or information to your output.</p> <hr/> <p>LLM (Searcher): {"query_id": 0, "relevance": [{"doc_id": 224, "is_relevant": true, "is_relevant_explanation": "This document mentions John Ball’s connection to Hoylake and his influence on golf history, making it relevant to the question."}, {"doc_id": 705, "is_relevant": true, "is_relevant_explanation": "This document also discusses John Ball’s influence on golf history and his relationship with Hoylake, making it relevant to the question."}], "change_search_query": false, "change_search_query_explanation": "The current documents are relevant and contain useful information about John Ball’s influence on golf at Hoylake.", "new_search_query": "", "end_search": true, "end_search_explanation": "The provided documents are sufficient to answer the question about John Ball’s influence on golf history at Hoylake."}</p>
<p>Coordinator</p>	<p>User (Searcher): "only the relevant documents found by searcher"</p> <hr/> <p>LLM : {"agent": "reasoner", "input": {"question": "How did John Ball influence golf history at Hoylake, and what strategic challenges does the course present regarding out of bounds?", "information": "'John Ball, Jr. was born in Hoylake, Cheshire and grew up playing golf on the Royal Liverpool course. He won several Amateur and Open Championships, including the 1890 Open Championship. He was known for his graceful swing and refusal to use a niblick. In 1878, he competed in his first Open Championship and finished fourth at Prestwick.'}, "aspect": "John Ball’s influence on golf history at Hoylake"}, "reason": "I need to analyze the information I’ve gathered to determine how John Ball influenced golf history at Hoylake."}</p>

<p>Reasoner</p>	<p>User (Coordinator): Output from Coordinator [Avoiding Repetition]</p> <hr/> <p>LLM (Reasoner): {"analysis": "John Ball, Jr. was born and raised in Hoylake, making him a local figure associated with the Royal Liverpool Golf Club, which is located there.", "He had significant success in golf competitions, winning multiple Amateur and Open Championships, with the most notable being the 1890 Open Championship. This highlights his skill and impact on the sport.", "His participation in the 1878 Open Championship at Prestwick, even before winning the major title, indicates his early prominence in professional golf and his potential influence on the tournament's legacy.", "Ball's graceful swing and refusal to use a niblick (an early type of iron club) reflect his unique style and contributed to his reputation as a skilled player, potentially influencing other golfers' techniques and preferences over time.", "Given his connection to Hoylake and the Royal Liverpool Golf Club, Ball likely played the course extensively, contributing to its historical significance and possibly offering insights or suggestions that shaped the venue's development and reputation."}</p>
<p>Coordinator</p>	<p>User (Reasoner): "Output from Reasoner [Avoiding Repetition]"</p> <hr/> <p>LLM : {"agent": "searcher", <input": "how="" and="" at="" ball="" bounds?",<br="" challenges="" course="" did="" does="" golf="" history="" hoylake,="" influence="" john="" of="" out="" present="" regarding="" strategic="" the="" what="" {"question":=""></input":> "information": "John Ball, Jr. was born and raised in Hoylake, making him a local figure associated with the Royal Liverpool Golf Club, which is located there. He had significant success in golf competitions, winning multiple Amateur and Open Championships, with the most notable being the 1890 Open Championship. His participation in the 1878 Open Championship at Prestwick, even before winning the major title, indicates his early prominence in professional golf and his potential influence on the tournament's legacy. Ball's graceful swing and refusal to use a niblick reflect his unique style and contributed to his reputation as a skilled player, potentially influencing other golfers' techniques and preferences over time. Given his connection to Hoylake and the Royal Liverpool Golf Club, Ball likely played the course extensively, contributing to its historical significance and possibly offering insights or suggestions that shaped the venue's development and reputation.", "suggestions": ["strategic challenges of Hoylake course", "out of bounds rules in golf", "Royal Liverpool Golf Club history"], "reason": "I need to find more information about the strategic challenges presented by the Hoylake course and the specific rules related to out of bounds to complete the analysis."}</p>

Searcher	User (Coordinator): Output from Coordinator [Avoiding Repetition]
	<hr/> <p>LLM (Searcher): {"search_query": "strategic challenges of Hoylake course", "search_query_explanation": "This search query aims to find information about the strategic challenges presented by the Hoylake course, which is crucial for understanding the second part of the question."}</p> <hr/> <p>User (Retriever): This is the information resulted from your search query: Query ID: 1 Documnet 1 ID: 399 Document 1 text: The old seventeenth, now the first, known as the Royal hole, was one of Harry Colt2019s most famous creations, with the green set right against Stanley Road; sadly, because it was impossible to get spectators round it, the green had to go to get the Open back to Hoylake.... Documnet 2 ID: 829 Document 2 text: The Open requires mental strength to overcome the obstacles thrown by Mother Nature. Having excellent control of ball flight is essential for any player to be successful when playing in windy, rainy and cold conditions.... Now I need you to verify the information. Be as objective as possible in your verification. Note you should strictly follow the given format and do not add any extra explanation or information to your output.</p> <hr/> <p>LLM (Searcher): { "query_id": 1, "relevance": [{"doc_id": 399, "is_relevant": true, "is_relevant_explanation": "This document discusses the strategic challenges of various golf courses, including Hoylake, particularly focusing on the use of out of bounds lines and bunkers to create strategic elements."} {"doc_id": 829, "is_relevant": false, "is_relevant_explanation": "This document focuses on the mental and physical preparation for playing in difficult weather conditions and does not discuss the strategic challenges of Hoylake or out of bounds rules."}], "change_search_query": false, "change_search_query_explanation": "The current documents are relevant and contain useful information about John Ball's influence on golf at Hoylake.", "new_search_query": "", "end_search": true, "end_search_explanation": "The provided documents are sufficient to answer the question about John Ball's influence on golf history at Hoylake."}</p> <hr/> <p>User (Retriever): This is the information resulted from your search query: Query ID: 2 Documnet 1 ID: 206 Document 1 text: But it is important to note that, when older courses were built, the penalty for going out of bounds was not necessarily quite as severe as it is today. Back in the day, the use of out of bounds as a strategic hazard was, if not exactly commonplace, far from unknown.... Documnet 2 ID: 399 Document 2 text: The old seventeenth, now the first, known as the Royal hole, was one of Harry Colt2019s most famous creations, with the green set right against Stanley Road; sadly, because it was impossible to get spectators round it, the green had to go to get the Open back to Hoylake...</p> <hr/> <p>LLM (Searcher): { "query_id": 2, "relevance": [{"doc_id": 206, "is_relevant": true, "is_relevant_explanation": "This document discusses the strategic use of out of bounds at Hoylake, providing specific examples and insights into how the course utilizes these areas."} {"doc_id": 399, "is_relevant": true, "is_relevant_explanation": "This document also provides detailed information about the strategic use of out of bounds at Hoylake, including examples like the Royal hole and the ninth hole (formerly the seventh)."}], "change_search_query": false, "change_search_query_explanation": "Both documents are relevant and contain useful information about the strategic challenges of Hoylake's course, particularly concerning out of bounds lines.", "new_search_query": ""}</p>

<p>Coordinator</p>	<p>User (Searcher): "only the relevant documents found by searcher"</p> <hr/> <p>LLM: {"agent": "answerer", "input": {"question": "How did John Ball influence golf history at Hoylake, and what strategic challenges does the course present regarding out of bounds?", "guidance": "Provide a detailed account of John Ball's influence on golf at Hoylake, highlighting his achievements and unique style. Also, describe the strategic challenges presented by the Hoylake course, particularly focusing on the out of bounds rules and how they affect gameplay." "important_information": "John Ball's connection to Hoylake, his golfing achievements, and the course's out of bounds rules." "reason": "Based on the gathered information, I will now craft a comprehensive response addressing both aspects of the question."}</p>
<p>Answerer</p>	<p>User (Coordinator): Output from Coordinator and relevant retrieved documents [Avoiding Repetition]</p> <hr/> <p>LLM (Answerer): {"response": "John Ball significantly influenced golf history at Hoylake, Cheshire, where he grew up playing on the Royal Liverpool Golf Club course. Born in Hoylake, Ball was the son of the Royal Hotel's prosperous owner, which placed him in the heart of the golfing community. His achievements include winning the Open Championship in 1890 and the Amateur Championship in 1888, making him the first English-born player to win both titles in the same year. Ball's unique style was characterized by his graceful swing and his refusal to use a niblick, which he considered 'another bloody spade.' This stubbornness and determination made him a formidable player in match play. Hoylake, known for its storied golfing history, presents strategic challenges, particularly concerning out of bounds rules. The course's design incorporates out of bounds lines as strategic hazards, encouraging players to take risks for better shot lines. For instance, the traditional opening hole features a drive over a turf 'cop,' with the favored strategic line being close to the cop. The use of out of bounds as a strategic element adds complexity to the game, requiring players to balance risk and reward. Overall, John Ball's influence on golf at Hoylake is marked by his exceptional achievements and his impact on the strategic design of the course, particularly through the thoughtful integration of out of bounds rules."}</p>

<p>Coordinator</p>	<p>User (Answerer): Output from Answerer [Avoiding Repetition]</p> <hr/> <p>LLM: {"agent": "validator", "input": { "question": "How did John Ball influence golf history at Hoylake, and what strategic challenges does the course present regarding out of bounds?" "information": "John Ball significantly influenced golf history at Hoylake, Cheshire, where he grew up playing on the Royal Liverpool Golf Club course. Born in Hoylake, Ball was the son of the Royal Hotel's prosperous owner, which placed him in the heart of the golfing community. His achievements include winning the Open Championship in 1890 and the Amateur Championship in 1888, making him the first English-born player to win both titles in the same year. Ball's unique style was characterized by his graceful swing and his refusal to use a niblick, which he considered 'another bloody spade.' This stubbornness and determination made him a formidable player in match play. Hoylake, known for its storied golfing history, presents strategic challenges, particularly concerning out of bounds rules. The course's design incorporates out of bounds lines as strategic hazards, encouraging players to take risks for better shot lines. For instance, the traditional opening hole features a drive over a turf 'cop,' with the favored strategic line being close to the cop. The use of out of bounds as a strategic element adds complexity to the game, requiring players to balance risk and reward. Overall, John Ball's influence on golf at Hoylake is marked by his exceptional achievements and his impact on the strategic design of the course, particularly through the thoughtful integration of out of bounds rules."}, "response": "John Ball significantly influenced golf history at Hoylake, Cheshire, where he grew up playing on the Royal Liverpool Golf Club course. Born in Hoylake, Ball was the son of the Royal Hotel's prosperous owner, which placed him in the heart of the golfing community. His achievements include winning the Open Championship in 1890 and the Amateur Championship in 1888, making him the first English-born player to win both titles in the same year. Ball's unique style was characterized by his graceful swing and his refusal to use a niblick, which he considered 'another bloody spade.' This stubbornness and determination made him a formidable player in match play. Hoylake, known for its storied golfing history, presents strategic challenges, particularly concerning out of bounds rules. The course's design incorporates out of bounds lines as strategic hazards, encouraging players to take risks for better shot lines. For instance, the traditional opening hole features a drive over a turf 'cop,' with the favored strategic line being close to the cop. The use of out of bounds as a strategic element adds complexity to the game, requiring players to balance risk and reward. Overall, John Ball's influence on golf at Hoylake is marked by his exceptional achievements and his impact on the strategic design of the course, particularly through the thoughtful integration of out of bounds rules."}, "reason": "I need to validate if the response accurately addresses the question and is grounded in the provided information."}}</p>
<p>Validator</p>	<p>User (Coordinator): Output from Coordinator [Avoiding Repetition]</p> <hr/> <p>LLM (Validator): {"extracted_criteria": [{"criteria": "detailed", "criteria_explanation": ""The question asks about John Ball's influence on golf history at Hoylake and the strategic challenges presented by the course. A detailed response would cover these aspects comprehensively.", "is_response_valid": true, "is_response_valid_feedback": "The response is detailed and covers both the historical influence of John Ball and the strategic challenges presented by the course."}], "is_groundedly_supported": true, "is_groundedly_supported_feedback": "The response is well-grounded in the provided information and includes specific examples such as the Open Championship wins and the strategic use of out of bounds lines.", "is_correctly_answered": true, "is_correctly_answered_feedback": "The response correctly answers the question by detailing John Ball's influence and explaining the strategic challenges presented by Hoylake."}}</p>

Coordinator	User (Validator): Output from Validator [Avoiding Repetition] <hr/> LLM: {"agent": "finisher", "input": "finished": true}
--------------------	--