

FACT, FETCH, AND REASON: A UNIFIED EVALUATION OF RETRIEVAL-AUGMENTED GENERATION

Satyapriya Krishna*

Harvard University

Kalpesh Krishna[†], Anhad Mohananey[†], Steven Schwarcz, Adam Stambler
Shyam Upadhyay & Manaal Faruqui

Google, Inc.

ABSTRACT

Large Language Models (LLMs) have demonstrated significant performance improvements across various cognitive tasks. An emerging application is using LLMs to enhance retrieval-augmented generation (RAG) capabilities. These systems require LLMs to understand user queries, retrieve relevant information, and synthesize coherent and accurate responses. Given the increasing real-world deployment of such systems, comprehensive evaluation becomes crucial. To this end, we propose **FRAMES** (**F**actuality, **R**etrieval, **A**nd reasoning **M**Easurement **S**et), a high-quality evaluation dataset designed to test LLMs' ability to provide factual responses, assess retrieval capabilities, and evaluate the reasoning required to generate final answers. While previous work has provided datasets and benchmarks to evaluate these abilities in isolation, **FRAMES** offers a unified framework that provides a clearer picture of LLM performance in end-to-end RAG scenarios. Our dataset comprises challenging multi-hop questions that require the integration of information from multiple sources. We present baseline results demonstrating that even state-of-the-art LLMs struggle with this task, achieving 0.40 accuracy with no retrieval. The accuracy is significantly improved with our proposed multi-step retrieval pipeline, achieving an accuracy of 0.66 (>50% improvement). We hope our work will help bridge evaluation gaps and assist in developing more robust and capable RAG systems.

1 INTRODUCTION

Recent advancements in Large Language Models (LLMs) have significantly enhanced their capabilities across various natural language processing tasks, especially in systems that demand both factual accuracy and sophisticated reasoning for complex queries (Zhao et al., 2023). Retrieval-augmented generation (RAG) techniques (Lewis et al., 2020; Fan et al., 2019; Guu et al., 2020) have become a powerful approach by leveraging the strengths of retrieval systems and the generative capabilities of LLMs. These techniques are particularly effective for tasks requiring multi-hop reasoning, factual grounding, and synthesizing information from diverse knowledge domains (Gao et al., 2023). However, despite this progress, the evaluation of RAG systems remains fragmented and insufficient, as existing benchmarks typically assess components like retrieval, factual correctness, and reasoning in isolation (Yu et al., 2024a). This piecemeal approach fails to capture the holistic performance of these systems in real-world applications (Yu et al., 2024b).

To bridge this gap, we introduce a novel evaluation framework, **FRAMES**¹ (**F**actuality, **R**etrieval, **A**nd reasoning **M**Easurement **S**et), designed to rigorously test LLMs on all three core capabilities—fact retrieval, reasoning across multiple constraints, and accurate synthesis of information into coherent responses. Unlike existing datasets such as TruthfulQA (Lin et al., 2021), HotpotQA (Yang et al., 2018b), or GSM8k (Cobbe et al., 2021), which focus on isolated aspects of LLM performance,

*Work done as an intern at Google. Corresponding Author: skrishna@g.harvard.edu

[†]Equal contribution

¹Dataset link : <https://huggingface.co/datasets/google/frames-benchmark>

Table 1: Comparison of **FRAMES** against other datasets. **FRAMES** provides a combination of evaluation samples to test the factuality, retrieval, and reasoning of RAG systems. The dataset also covers multi-hop/step questions along with temporal disambiguation.

| Dataset | Factuality | Retrieval | Reasoning | Multi-Hop/Step | Temporal Disambiguation |
|---|------------|-----------|-----------|----------------|-------------------------|
| FRAMES | ✓ | ✓ | ✓ | ✓ | ✓ |
| TruthfulQA (Lin et al., 2021) | ✓ | ✗ | ✗ | ✗ | ✗ |
| OpenbookQA (Mihaylov et al., 2018) | ✓ | ✗ | ✗ | ✗ | ✗ |
| HotpotQA (Yang et al., 2018b) | ✓ | ✗ | ✗ | ✓ | ✗ |
| HybridQA (Chen et al., 2020) | ✗ | ✗ | ✓ | ✓ | ✓ |
| GSM8k (Cobbe et al., 2021) | ✗ | ✗ | ✓ | ✓ | ✗ |
| Multihop-RAG(Tang & Yang, 2024) | ✓ | ✓ | ✗ | ✓ | ✗ |
| MoreHopQA (Schnitzler et al., 2024) | ✓ | ✗ | ✓ | ✓ | ✗ |
| MuSiQue (Trivedi et al., 2022) | ✓ | ✗ | ✓ | ✓ | ✗ |
| NaturalQuestions (Kwiatkowski et al., 2019) | ✗ | ✓ | ✗ | ✗ | ✓ |
| TriviaQA (Joshi et al., 2017) | ✓ | ✗ | ✗ | ✗ | ✗ |
| ELI5 (Fan et al., 2019) | ✗ | ✓ | ✓ | ✗ | ✗ |

FRAMES provides an integrated evaluation that challenges models across these dimensions simultaneously. This approach offers a more accurate reflection of how these systems perform as end-to-end reasoning solutions, especially in scenarios requiring multi-document retrieval and complex reasoning. For instance, a sample from our dataset is *"How many years earlier would Punxsutawney Phil have to be canonically alive to have made a Groundhog Day prediction in the same state as the US capitol?"* This requires the model to perform temporal and numerical reasoning after retrieving the critical articles needed to answer the question.

Our work addresses a critical void in the current landscape by offering a challenging evaluation benchmark that not only tests the individual components of LLMs but also evaluates their performance in an end-to-end context. Through our dataset, we simulate realistic, multi-document queries to assess the ability of LLMs to retrieve relevant facts, reason accurately, and synthesize information into coherent responses. Additionally, we present empirical results on the performance of state-of-the-art models, highlighting both their strengths and the limitations in their reasoning capabilities. These findings pave the way for further research and development of more robust and efficient retrieval-augmented generation systems. Our key contributions are as follows:

- We introduce **FRAMES**, a novel dataset of 824 test samples designed to evaluate LLMs’ ability to retrieve and reason across multiple documents in a unified framework.
- We provide a comprehensive evaluation of state-of-the-art LLMs, highlighting their performance on factuality, retrieval, and reasoning tasks across diverse domains.
- We present new empirical insights into the limitations of existing LLMs in handling multi-hop and temporal reasoning tasks, offering avenues for future research to improve these systems.
- We propose a multi-step retrieval and reasoning framework that compels models to iteratively retrieve and reason, significantly enhancing their performance on complex queries.

2 FRAMES

FRAMES (Factuality, Retrieval, And reasoning **ME**asurement **Set**) is an evaluation set of 824 questions designed to provide an end-to-end evaluation of Retrieval Augmented Generation (RAG) systems. It assesses three key components of a RAG system: Factuality, Retrieval, and Reasoning. Unlike most existing datasets and benchmarks that evaluate each of these RAG components in isolation, **FRAMES** offers a comprehensive test bed to gain a clear understanding of the overall quality of RAG systems(Lin et al., 2022; Yang et al., 2018a; Welbl et al., 2017). This holistic approach allows for a more accurate reflection of how these systems perform in real-world scenarios. In this section, we first detail our data collection process, which involved both synthetic data generation attempts and human annotation. Next, we present the dataset statistics, showcasing the diversity of topics and reasoning types covered. Finally, we outline the rigorous quality checks implemented to ensure

Task Instruction Prompt for Human Annotation

Task Description
 Create n factoid questions that demand multi-hop reasoning based on information found across multiple Wikipedia articles. These questions should ideally have a single, unambiguous answer and may optionally incorporate elements of challenging reasoning. Here's a breakdown of the key terms and their relationships:

- **Factoid Questions:** These are trivia-style questions with a single, clearly defined, and factually correct answer.
- **Multi-hop Reasoning:** This refers to the core requirement that answering the questions necessitates combining information from different sections within multiple Wikipedia articles. The example provided ("What is the name of the river that flows through the city where the Eiffel Tower is located?") highlights how this differs from a simple factoid question ("What is the capital of France?").
- **Challenging Reasoning:** This aspect encourages the inclusion of questions that go beyond simple information retrieval and demand critical thinking. This can be achieved through various question types like:
 - Numerical Reasoning: Involving counting, comparisons, or calculations. <examples>
 - Tabular Reasoning: Involving statistics found in tables / info boxes in wikipedia. <examples>
 - Multiple Constraints: Questions involving multiple constraints, whose intersection points towards a unique answer. <examples>
 - Temporal Reasoning : Questions involving reasoning through timelines. <examples>
 - Post processing: This involves requiring the answerer to perform specific post-processing steps after all necessary facts have been gathered. <examples>

Additional Requirements:

- Wikipedia Articles: The information used to answer the questions must be sourced from Wikipedia articles.
- Standalone and Context-Independent: Questions should be understandable without requiring additional information or context.
- Single, Unambiguous Answer: Each question should have only one correct answer, leaving no room for ambiguity.
- Avoid boolean questions (yes/no questions) that can be answered with a simple "yes" or "no." <examples>

Figure 1: Task instruction provided to human annotators to generate samples for **FRAMES**.

the dataset's reliability and challenging nature. By providing this end-to-end evaluation framework, **FRAMES** aims to bridge the gap in existing benchmarks and foster the development of more robust and efficient RAG systems.

Synthetic Data Generation Attempts. We start our data collection process with synthetic dataset generation to explore a potentially cost-effective alternative to expensive human annotation. We prompt a state-of-the-art LLM with instructions to use multiple articles to generate questions that would require information from multiple articles to answer. The prompt (shown in Figure 6 in Appendix A) takes as input the number of articles provided to generate questions. However, we observed significant issues with this approach. While the LLMs were able to generate coherent questions, there was a high proportion of hallucinated questions and answers (>30%). Additionally, the LLM struggled to generate questions that strictly required more than four articles. To evaluate the potential of this approach, we manually cleaned the hallucinated questions and answers from the obtained set. We then evaluated the same LLM on these cleaned questions and obtained an accuracy

Table 2: This table provides descriptions of the different reasoning types to which each question in **FRAMES** belongs. The distribution of samples belonging to each reasoning type is shown in Figure 2.

| Reasoning Type | Description |
|----------------------|--|
| Numerical Reasoning | This involves counting, comparisons, or calculations. For example, the question <i>"How many times faster is the second fastest bird in the Americas compared to the fastest human in the world? Round to the nearest integer."</i> asks for a calculation comparing the speeds of two objects. |
| Tabular Reasoning | This involves statistics found in tables or infoboxes in Wikipedia. For example, the question <i>"How many runs did the West Indies vice-captain score in the 1983 World Cup?"</i> requires the answerer to analyze tabular data of top run scorers and extract the relevant information. |
| Multiple Constraints | This involves questions with multiple constraints whose intersection points towards a unique answer. For example, <i>"I'm thinking of an airport near the intersection of US-52 and I-95. Can you remind me which one it is?"</i> This query has two constraints: first, to locate an airport, and second, that it should be near the intersection of US-52 and I-95. |
| Temporal Reasoning | This involves reasoning through timelines. For example, <i>"Leonardo DiCaprio once won an Oscar for best actor. Who won the award for best costume design sixteen years earlier?"</i> |
| Post-Processing | This requires the answerer to perform specific post-processing steps after all necessary facts have been gathered. For example, consider the question: <i>"What is five years after the founding of the largest country in North America in Roman numerals?"</i> . This question requires the following sub-instructions: (1) Numerical reasoning: Add five years to the founding date, and (2) Post-processing: Convert the resulting year into Roman numerals. |

of $\sim 32\%$, suggesting that the legitimate questions generated by LLMs were indeed challenging for state-of-the-art models. There are two key takeaways from our experimentation with synthetic data generation: (1) Synthetic test data requires heavy manual cleaning before usage, which suggests that we will need to rely on human annotations instead of LLMs to generate the final evaluation set; and (2) models performed significantly poorly on the correct test samples we tested on, suggesting that the instruction to create questions can be used to generate a challenging evaluation set.

Human annotation. Given these findings, we decided to use the core instruction for generating questions that combine information from multiple articles as a guide for human annotation, shown in Figure 1. This approach aimed to leverage the challenging nature of the synthetic questions while also mitigating the issues of hallucination present in LLM-generated content. Human annotators were tasked with creating questions that required information from multiple Wikipedia articles, following a similar structure to the synthetic prompts but with greater reliability and accuracy. The outcome of this human annotation resulted in 824 questions with their correct responses along with the list of Wikipedia articles needed to answer the questions. We also ask the human annotators to label each question based on five reasoning types, i.e, Numerical Reasoning, Tabular Reasoning, Multiple Constraints, Temporal Reasoning, and Post-Processing, described in more details in Table 2. Please note that a question can belong to multiple reasoning types. To ensure the highest quality annotations, we engaged a team of carefully vetted experts with extensive experience in question generation and complex reasoning tasks.

Dataset Statistics. The dataset comprises questions related to a diverse set of topics from Wikipedia, involving subjects such as history, sports, science, animals, health, etc. Each question in our dataset require 2-15 Wikipedia articles to answer, with the distribution of the percentage of dataset requiring different numbers of Wikipedia articles shown in Figure 2 (left). Approximately 36% of questions require two articles to answer, $\sim 35\%$ require three articles, $\sim 16\%$ require four articles, and so on. This distribution also represents the general trend of queries asked from LLMs in the real world (Liu et al., 2009), since the proportion of questions requiring two articles is higher than more complicated questions requiring a greater number of articles. Additionally, we have a healthy distribution of questions belonging to different reasoning types, shown in Figure 2 (right).

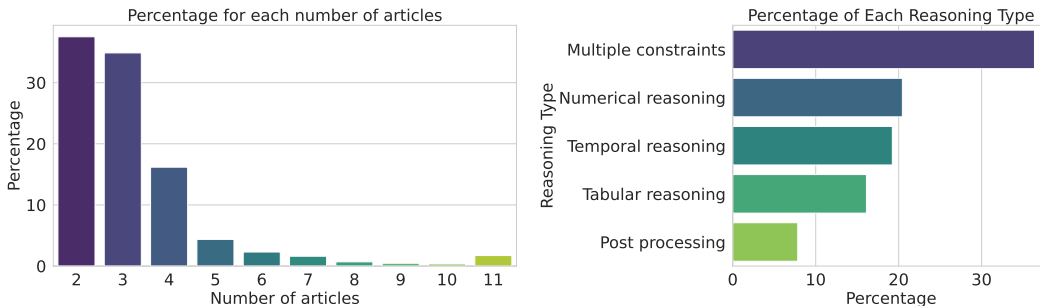


Figure 2: The figure shows the distribution of questions in the **FRAMES**, with the percentage of questions requiring different numbers of Wikipedia articles (left) and the percentage of the dataset belonging to each reasoning type (right). Please note that the percentage bar for 11 denotes the percentage of questions requiring 11 or more Wikipedia articles.

Questions requiring reasoning over multiple constraints hold the highest percentage of data samples in the test ($\sim 36\%$), followed by questions requiring numerical reasoning ($\sim 20\%$). Please note that many questions in the dataset also require a combination of different reasoning abilities to find an answer.

Quality Checks. Other than the data collection process described in the section above, human annotators also implemented several quality checks to ensure the dataset’s high quality and effectiveness in evaluating RAG capabilities:

- **Ensuring correctness and groundedness to Wikipedia:** We verified the correctness of questions and their corresponding answers by re-annotating the questions. Human annotators were asked to confirm if the provided answer was correct and could be answered using the associated Wikipedia pages. This annotation process was conducted three months after collecting the initial data, filtering out 5.5% of samples where the answer was no longer true after that period.
- **Removing ambiguity due to freshness (temporal disambiguation):** Annotators added extra context to disambiguate answers that could change over time. For example, a question like *"Which country were holders of the FIFA World Cup the last time the UEFA Champions League was won by a club from London?"* was revised to *"As of August 1, 2024, which country were holders of the FIFA World Cup the last time the UEFA Champions League was won by a club from London?"*. This approach mitigates issues with frequent manual updates required for maintaining previous datasets (Vu et al., 2023; Kasai et al., 2024).
- **Preventing guesswork by ensuring a large output space:** We removed questions with binary answers ("yes" or "no") to prevent LLMs from achieving 50% accuracy through random guessing. This ensures the dataset is challenging enough to clearly evaluate LLM capabilities.
- **Ensuring dataset interpretability and reliability:** We limited the articles to those from Wikipedia, which has a lower chance of containing unreliable information compared to other sources.
- **Addressing contamination issues:** To mitigate concerns about potential contamination due to Wikipedia articles being in LLM training sets, we designed questions that require additional reasoning and operations beyond simple fact retrieval. For instance, the question *"How many years earlier would Punxsutawney Phil have to be canonically alive to have made a Groundhog Day prediction in the same state as the US capitol?"* requires not only fact extraction but also additional calculations.

3 EMPIRICAL ANALYSIS

After obtaining a high-quality test set, we evaluate state-of-the-art LLMs on their ability to answer questions that require proficiency in factuality, retrieval, and reasoning. Our analysis is divided into

two sections: (1) Single-step Evaluations (Section 3.1): Here, we evaluate the LLMs based on a single-shot inference, where the idea is to ask the question and assess the response after a single inference call. This evaluation is further divided into cases with and without retrieval to analyze the impact of retrieval on performance. (2) Multi-Step Evaluations (Section 3.2): In this case, we evaluate the models after making more than a single inference step, focusing on scenarios where retrieval is explicitly required. The motivation for multi-step evaluations is to determine whether forcing the model to retrieve and reason across multiple steps could lead to performance improvements. Next, we describe the details of the two sets of experiments.

3.1 SINGLE-STEP EVALUATIONS

In this set of experiments, we evaluate the model using several baseline prompting methods on our test set to understand how well existing LLMs perform. Specifically, we experiment with three baseline approaches: **(1) Naive Prompt:** This is a straightforward approach where we simply ask the question to the model and evaluate if the model’s response without search retrieval contains the correct answer. **(2) BM25-Retrieved Prompt (n_docs):** This approach augments the question with the top n_docs documents having the highest BM25 score (Robertson et al., 1995) retrieved from a Wikipedia data dump². The BM25 score is computed between the question and every article in the Wikipedia dump, after which the top n_docs with the highest scores are added to the prompt. The motivation behind this approach is to observe improvements in model performance when relevant articles are added to the context. This is denoted as BM25-R (n_doc) in the results. **(3) Oracle Prompt:** This prompt includes the question along with all the ground truth Wikipedia articles used by the human annotators to generate the question. The performance of the Oracle Prompt provides the upper bound of model performance in the case of a perfect retrieval system that is able to extract all the relevant articles.

Experiment Setup. For the experiments, we use Gemini-Pro-1.5-0514 (Google, 2024b), Gemini-Flash-1.5-0514 (Google, 2024a), Gemma2-27b (Gemma et al., 2024), and Gemma2-9b (Gemma et al., 2024) as the state-of-the-art LLM since they have shown great performance on several public benchmarks. Since the gold answers to questions in the dataset are free-form tokens instead of choices from multiple-choice answers, we use an LLM to evaluate if the outcome from the LLM under evaluation matches the gold answer, using the prompt shown in Figure 7 in Appendix B. This auto-rating mechanism was tested against human evaluations, in which the LLM-based evaluation showed strong alignment with human annotations (accuracy: 0.96 and Cohen’s Kappa: 0.889 for Gemini-Pro-1.5-0514 as autorating LLM), making LLM-based evaluation a suitable approach to evaluate the correctness of model responses.

LLMs perform poorly in single-step evaluations. Based on results shown in Table 3, we observe that naive prompting attains a performance of $\sim 40\%$ with gradual increases when including BM25 retrieved articles for Gemini-Pro-1.5-0514. The model achieves an accuracy of $\sim 45\%$ when the number of documents in the context is 2, and $\sim 47\%$ when double the number of articles are added to the context. These improvements demonstrate the room for enhancement when the model is able to retrieve relevant articles required to answer the question. The core reason behind these improvements is the improvement in recall in the articles present in context which increased from 0.12 (BM25-R(n_docs = 2)) to 0.15 (BM25-R (n_docs = 4)). In addition to these approaches, we observe an accuracy of $\sim 72\%$ for Gemini-Pro-1.5-0514 when all the gold Wikipedia articles are provided in the context, which we call Oracle Prompt. Out of $\sim 28\%$ samples where the model made errors, $\sim 80\%$ of those misclassifications belong to numerical, tabular, and post-processing categories. Hence, these misclassifications show the reasoning gaps in model performance where even after providing all the relevant facts, the model failed to reason through the different facts to provide a correct answer to the question. The accuracies obtained by the Naive Prompt and Oracle Prompt can be considered as the lower bound (when no relevant articles were provided to the model) and upper bound (when all relevant articles were provided to the model) of model performances on **FRAMES**. This pattern can also be seen in Figure 3 where we plotted accuracy for each reasoning type and observe that the model performed the lowest in numerical, post-processing, and tabular reasoning tasks. We also observe that adding BM25 retrieved articles primarily helped with questions requiring reasoning through multiple constraints ($\sim 8\%$ improvement) and post-processing ($\sim 10\%$

²https://www.tensorflow.org/datasets/catalog/wikipedia#wikipedia20230601en_default_config

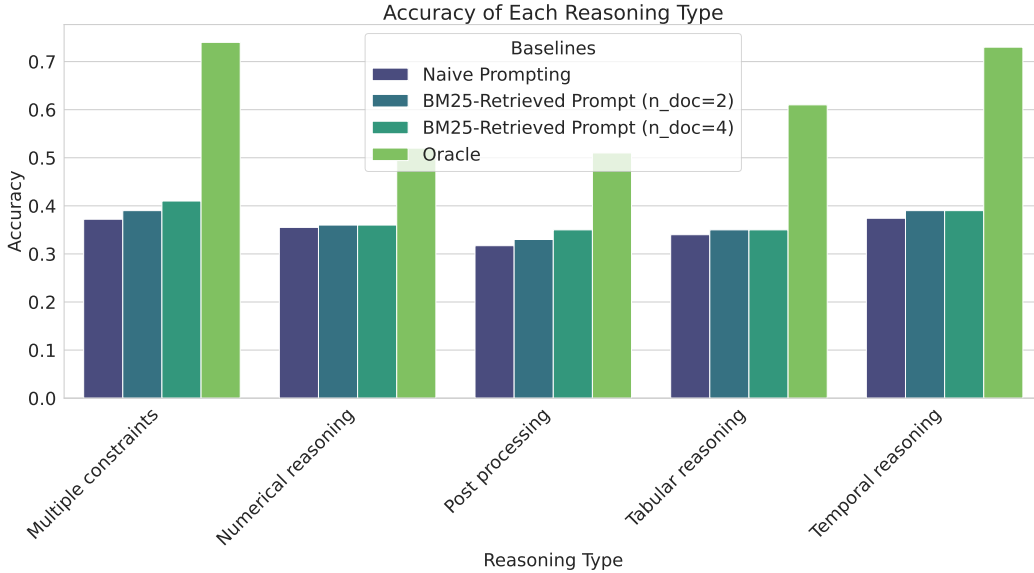


Figure 3: Accuracy of Gemini-Pro-1.5-0514 across different reasoning types in our test set. The results indicate superior performance on logical and temporal reasoning tasks, with notable deficiencies in numerical, tabular, and post-processing reasoning. The substantial performance improvements observed with oracle information underscore the critical role of relevant contextual information in enhancing model accuracy across all reasoning categories.

Table 3: This table presents the accuracy performance of Gemini-Pro-1.5-0514 (G-Pro-1.5), Gemini-Flash-1.5-0514 (G-Flash-1.5), Gemma2-27b, and Gemma2-9b on our proposed evaluation dataset. Please note that the performance of Gemma models is not reported for cases requiring longer context due to the small maximum context length of the model.

| Baselines | G-Pro-1.5 | G-Flash-1.5 | Gemma2-27b | Gemma2-9b |
|------------------|-----------|-------------|------------|-----------|
| Naive Prompt | 0.408 | 0.263 | 0.308 | 0.051 |
| BM25-R (n_doc=2) | 0.452 | 0.288 | - | - |
| BM25-R (n_doc=4) | 0.474 | 0.315 | - | - |
| Oracle Prompt | 0.729 | 0.665 | - | - |

improvement). This aligns well with the fact that providing more relevant articles helps in obtaining facts for each constraint, leading to improvements in performance. We take these learnings and experiment with a more complicated setup where the model is asked to find answer to questions through multiple iterations instead of a single step.

3.2 MULTI-STEP EVALUATIONS

Based on the findings from the previous experiment with single-step evaluations, where we observed an increase in performance when related articles are added to the context, we were led to explore a setting where the model is compelled to plan its search for relevant Wikipedia articles in order to find answers. More specifically, we design a pipeline where the model is asked a question along with the instruction to generate k search queries which are then used to extract the top- n_docs Wikipedia articles with the highest BM25 scores. These documents are then appended to the context. This process of query generation and retrieved article augmentation is carried forward for n steps. Once the n steps of retrieval are completed, the model is asked to answer the question based on the articles appended in the context, as shown in Algorithm 1. We conduct two sets of experiments here: (1) Vanilla with no explicit planning instructions, and (2) With search planning instructions to help the model navigate the search process efficiently. To implement this pipeline, we used the simplest document retrieval component which is essentially an index of Wikipedia pages, where the articles with the highest BM25 scores for each query are returned to the LLM and added to the

Algorithm 1 Multi-Step Evaluation with BM25 Retrieval

- 1: **Input:** Initial question Q , number of iterations n , number of queries k , number of documents n_docs
- 2: **Output:** Final response R
- 3: Initialize context $C \leftarrow \{Q\}$
- 4: **for** iteration $i = 1$ to n **do**
- 5: **for** query $j = 1$ to k **do**
- 6: Generate search query Q_{ij} based on context C
- 7: Retrieve top n_docs documents D_{ij} using BM25 based on Q_{ij}
- 8: $C \leftarrow C \cup (D_{ij} \setminus C)$ {Add only new documents to context}
- 9: **end for**
- 10: **end for**
- 11: Generate final response R using context C
- 12: **return** R

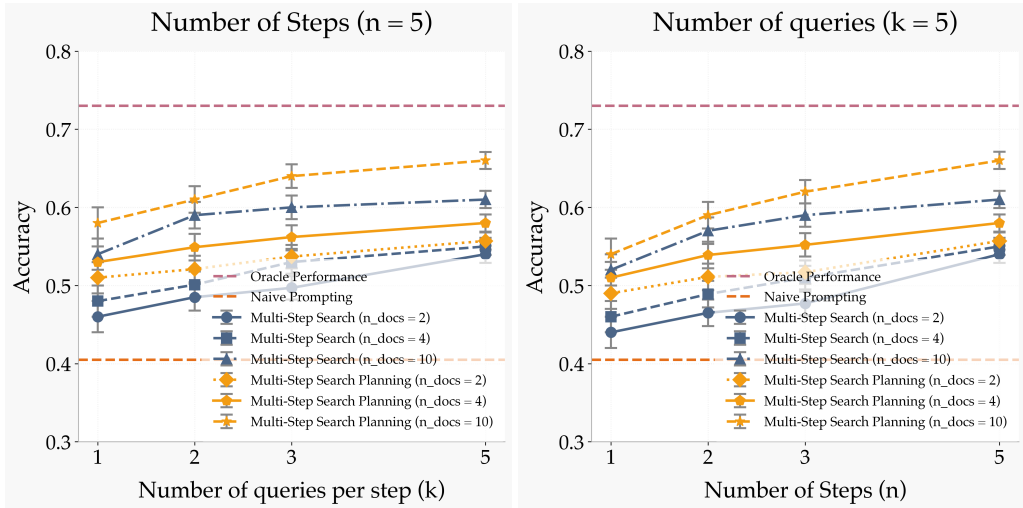


Figure 4: The figure shows the performance improvements when the number of steps (n) and number of queries per step (k) is changed. We achieved the best performance of 0.66 with the combination of $(k, n, n_docs) = (5, 5, 10)$

context. This retrieval component is used instead of making direct calls to an online search engine for two reasons: (1) We would like to keep the retrieval system constant to clearly evaluate the search planning capability of the LLMs instead of the retrieval system’s capability in returning the most relevant articles, and (2) The BM25-based retrieval system makes our pipeline reproducible and limits the search space to Wikipedia pages only, as the questions were generated from Wikipedia articles only.

Multi-Step retrievals significantly improve model performance. We plot model performance on different combinations of (k, n, n_docs) in Figure 5. Based on these results, we observe a steady increase in performance as the number of steps and queries are increased with accuracy increasing from ~ 0.45 to ~ 0.52 for the case of $(k, n, n_docs) = (5, 5, 2)$ for vanilla setting where the model is not provided with any specific planning instructions. This is expected, as more steps and queries allow the model to add more relevant documents to its context, leading the model to improve recall, which translates to better accuracy. However, the performance still remains quite low even after five iterations of search retrievals, which is computationally expensive since this requires six non-parallelizable inference calls (five for retrieval + one for final answering) to answer each question. One of the reasons we found behind this slow progress in performance is the lack of diversity in

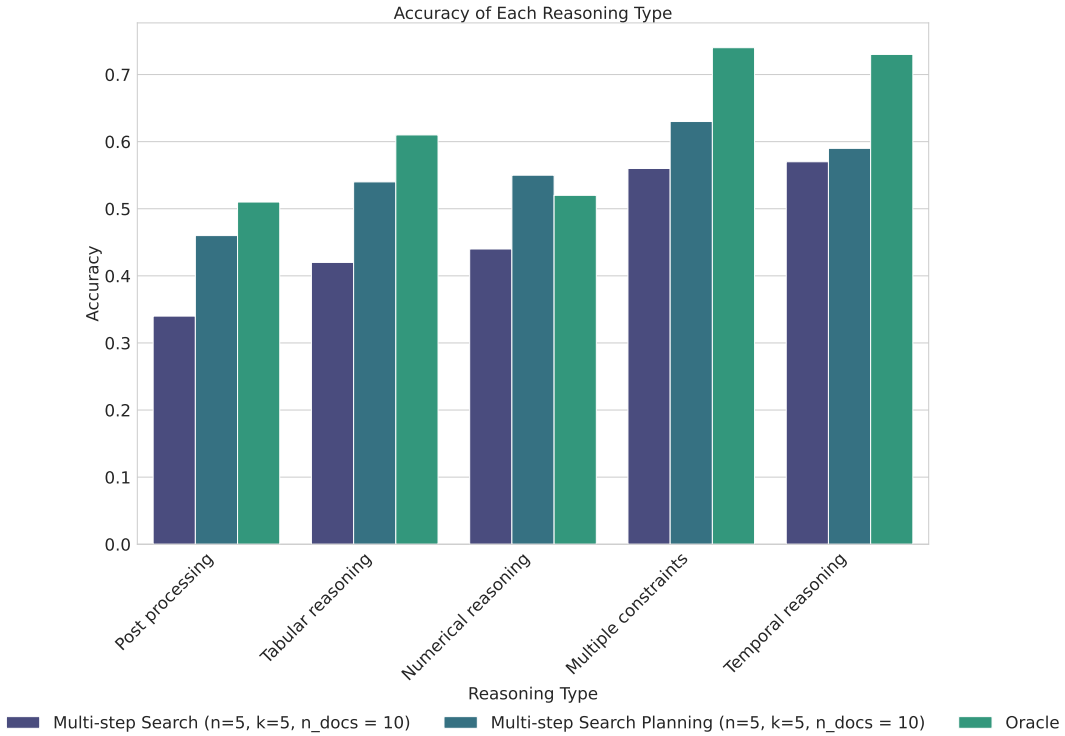


Figure 5: This plot shows the accuracy of Gemini-Pro-1.5-0514 on each reasoning type in our test set. We observe a significant increase in performance for all reasoning types when we use multi-step search planning, with the performance on numerical reasoning even exceeding oracle performance.

the queries generated by the model; it seemed like the model goes in the wrong direction in search retrievals and never corrects itself. To mitigate this problem, we experiment with two changes to the instructions: (1) We provide a few examples of how an ideal best-case search query sequence should look, and (2) We provide instructions not to repeat queries and force the model to "think step-by-step" (Kojima et al., 2022). We observe a very promising trend with these changes, where the model performance (0.66) through iterations reaches close to the oracle performance (0.73) by the end of five iterations of retrievals. We hope our benchmark will be useful for the community to further reduce the number of search calls and improve model accuracy.

4 RELATED WORKS

Evaluating Retrieval-Augmented Generation (RAG) systems has become increasingly important as these models integrate retrieval mechanisms with generative capabilities to enhance factual accuracy and reasoning (Yu et al., 2024b). Existing benchmarks, such as NaturalQuestions (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), and ELI5 (Fan et al., 2019), have been used to evaluate RAG models, but they often focus on specific aspects like retrieval accuracy or single-turn question answering without considering the full complexity of real-world applications. For instance, NaturalQuestions primarily tests retrieval precision, while TriviaQA emphasizes factual correctness in trivia-style questions. ELI5, on the other hand, is designed for explainability but does not rigorously assess the multi-hop reasoning necessary for synthesizing information from multiple sources. These benchmarks, while valuable, tend to evaluate RAG systems in a piecemeal fashion, missing the comprehensive assessment needed to truly measure their end-to-end capabilities. We provide additional comparisons against other datasets in Table 1.

FRAMES addresses these limitations by offering a unified and more holistic evaluation framework for RAG systems. Unlike existing datasets, **FRAMES** tests models across three critical dimensions: factual retrieval, reasoning, and synthesis. It incorporates complex multi-hop queries that require

models to retrieve and integrate information from various sources while also handling temporal disambiguation—a challenge not adequately covered by benchmarks like NaturalQuestions or ELI5. Additionally, **FRAMES** includes tasks that assess the synthesis of information into coherent and contextually accurate responses, ensuring that RAG systems are evaluated on their ability to perform in realistic, multifaceted scenarios. This makes **FRAMES** a more rigorous and comprehensive benchmark, well suited for guiding the development of next-generation RAG systems.

5 CONCLUSION

In this work, we introduced **FRAMES**, a comprehensive evaluation dataset designed to test the capabilities of Retrieval-Augmented Generation (RAG) systems across factuality, retrieval accuracy, and reasoning. Our experiments with state-of-the-art LLMs highlight the existing gaps in their ability to handle complex, multi-hop reasoning tasks. The baseline results showed that even advanced models struggle significantly with the challenging scenarios presented in **FRAMES**, achieving only moderate improvements when multi-step retrieval and reasoning strategies were employed. The **FRAMES** dataset addresses a critical need in the evaluation of RAG systems by offering an integrated framework that tests these systems in a more holistic manner compared to existing benchmarks. By simulating realistic, multi-document queries, **FRAMES** provides a clearer picture of the current capabilities and limitations of LLMs in real-world applications. Our findings underscore the importance of further enhancing both the retrieval mechanisms and the reasoning capabilities of these models to improve their overall performance.

Future Work. Moving forward, there are several promising avenues for future research. First, the development of more sophisticated retrieval strategies is essential. This includes exploring dense retrievers trained directly on the multihop retrieval task, such as those based on ColBERT (Khattab & Zaharia, 2020), or SimCSE (Gao et al., 2021) architectures. These approaches could better handle diverse and complex queries by adapting to the context iteratively. Second, improving the reasoning capabilities of LLMs remains a significant challenge. We can explore process supervision methods like those used in PRM-800K (Lightman et al., 2023), or investigate distillation techniques on successful trajectories, similar to approaches in ToolFormer (Schick et al., 2024) and DSPy (Khattab et al., 2023). These methods could enhance numerical, temporal, and post-processing reasoning. Additionally, we can explore modeling approaches such as context reduction of wiki articles to improve planning capabilities and training query generators for more effective information retrieval. Lastly, expanding the **FRAMES** dataset to include more diverse and domain-specific questions, as well as incorporating more dynamic elements such as real-time information retrieval, could further enhance its utility as a benchmark for next-generation RAG systems. It is important to note that future work should also address the potential limitations of our current approach, including the risk of pretraining data contamination, which may affect the generalizability and reliability of the results, particularly when using Wikipedia articles that could overlap with LLM training data.

Limitations. While **FRAMES** provides a comprehensive evaluation framework for RAG systems, it is important to acknowledge certain limitations. One significant concern is the potential for pretraining data contamination. As large language models are trained on vast amounts of internet data, there is a risk that some of the information in our dataset may have been seen by these models during their pretraining phase. This could lead to artificially inflated performance metrics and reduce the dataset’s effectiveness in measuring true generalization capabilities. Future iterations of **FRAMES** should explore techniques to mitigate this issue, such as using more recent or synthetic data, or developing methods to quantify and account for potential contamination. Additionally, while we have strived for diversity in our dataset, it may not fully represent the entire spectrum of real-world queries and scenarios, potentially limiting its applicability to certain domains or use cases. Addressing these limitations will be crucial for improving the robustness and reliability of RAG system evaluations.

REFERENCES

Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Wang. Hybridqa: A dataset of multi-hop question answering over tabular and textual data. *arXiv preprint arXiv:2004.07347*, 2020.

- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.
- Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. Eli5: Long form question answering. *arXiv preprint arXiv:1907.09190*, 2019.
- Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*, 2021.
- Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2023.
- Team Gemma, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.
- Google. Gemini 1.5 flash. <https://deepmind.google/technologies/gemini/flash/>, 2024a.
- Google. Gemini 1.5 pro. <https://blog.google/technology/ai/google-gemini-next-generation-model-february-2024/>, 2024b.
- Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval augmented language model pre-training. In *International conference on machine learning*, pp. 3929–3938. PMLR, 2020.
- Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*, 2017.
- Jungo Kasai, Keisuke Sakaguchi, Ronan Le Bras, Akari Asai, Xinyan Yu, Dragomir Radev, Noah A Smith, Yejin Choi, Kentaro Inui, et al. Realtime qa: what’s the answer right now? *Advances in Neural Information Processing Systems*, 36, 2024.
- Omar Khattab and Matei Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pp. 39–48, 2020.
- Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. Dspy: Compiling declarative language model calls into self-improving pipelines. *arXiv preprint arXiv:2310.03714*, 2023.
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35: 22199–22213, 2022.
- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020.
- Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let’s verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

- Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*, 2021.
- Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pp. 3214–3252. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.ACL-LONG.229. URL <https://doi.org/10.18653/v1/2022.acl-long.229>.
- Tie-Yan Liu et al. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval*, 3(3):225–331, 2009.
- Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*, 2018.
- Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. Okapi at trec-3. *Nist Special Publication Sp*, 109:109, 1995.
- Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36, 2024.
- Julian Schnitzler, Xanh Ho, Jiahao Huang, Florian Boudin, Saku Sugawara, and Akiko Aizawa. Morehopqa: More than multi-hop reasoning. *arXiv preprint arXiv:2406.13397*, 2024.
- Yixuan Tang and Yi Yang. Multihop-rag: Benchmarking retrieval-augmented generation for multi-hop queries. *arXiv preprint arXiv:2401.15391*, 2024.
- Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. *Trans. Assoc. Comput. Linguistics*, 10:539–554, 2022. doi: 10.1162/TACL_A_00475. URL https://doi.org/10.1162/tacL_a_00475.
- Tu Vu, Mohit Iyyer, Xuezhi Wang, Noah Constant, Jerry Wei, Jason Wei, Chris Tar, Yun-Hsuan Sung, Denny Zhou, Quoc Le, et al. Freshllms: Refreshing large language models with search engine augmentation. *arXiv preprint arXiv:2310.03214*, 2023.
- Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. Constructing datasets for multi-hop reading comprehension across documents. *CoRR*, abs/1710.06481, 2017. URL <http://arxiv.org/abs/1710.06481>.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun’ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pp. 2369–2380. Association for Computational Linguistics, 2018a. doi: 10.18653/V1/D18-1259. URL <https://doi.org/10.18653/v1/d18-1259>.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*, 2018b.
- Hao Yu, Aoran Gan, Kai Zhang, Shiwei Tong, Qi Liu, and Zhaofeng Liu. Evaluation of retrieval-augmented generation: A survey. *ArXiv*, abs/2405.07437, 2024a. URL <https://api.semanticscholar.org/CorpusID:269758033>.
- Hao Yu, Aoran Gan, Kai Zhang, Shiwei Tong, Qi Liu, and Zhaofeng Liu. Evaluation of retrieval-augmented generation: A survey. *arXiv preprint arXiv:2405.07437*, 2024b.
- Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

A SYNTHETIC DATA GENERATION PROMPT

Synthetic Data Generation

System: You are a helpful assistant.

User: ""**TASK:** You will be provided with {k_context} Wikipedia article extracts. Based on these extracts, generate {n_questions} challenging factoid questions that meet the following criteria:

1. **Standalone & Context-Independent:** Questions should not contain any references to "Article 1", "Article 2", etc. They should be understandable without any additional context.
2. **Unambiguous Answer:** Each question should have a single, clear, and factual answer.
3. **Multi-hop Reasoning:** Answering each question should require combining information from ALL {k_context} provided Wikipedia articles.
4. **Grounded in Context & Conceptual Format:** Each question must conceptually follow this format, seamlessly integrating information from each article:
 Start with a clear question word (What/How/Where/When).
 Introduce information from each article step-by-step, using connectors to link them logically.
 **Example connectors: 'in relation to', 'compared to', 'as a result of', 'which also', 'in addition to'.
 ** For each question: *
 Provide the single-word answer in parentheses after the question mark. *
 On a new line, clearly explain the reasoning process. *
 For each article, bullet point the specific piece of information used to formulate the question.

Example:
 Question: What type of bird, belonging to the Ardeidae family, went extinct around 1690 and was known for its terrestrial abilities? (Dodo)
 Reasoning: * **Article 1:** Provides information about the Dodo belonging to the Ardeidae family. * **Article 2:** Mentions the extinction of the Dodo around 1690. *
 Article 3: Highlights the Dodo's adaptation to terrestrial life.

{WIKI ARTICLES} ""

Figure 6: Prompt used to generate questions synthetically using Gemini-Pro-1.5-0514. k_context and n_questions are placeholders for the number of articles provided and the number of questions to generate per inference.

B AUTORATER PROMPT

Auto-rating Prompt

System: You are a helpful assistant.
User: ""===Task===

I need your help in evaluating an answer provided by an LLM against a ground truth answer. Your task is to determine if the ground truth answer is present in the LLM's response. Please analyze the provided data and make a decision.

===Instructions===

1. Carefully compare the "Predicted Answer" with the "Ground Truth Answer".
2. Consider the substance of the answers – look for equivalent information or correct answers. Do not focus on exact wording unless the exact wording is crucial to the meaning.
3. Your final decision should be based on whether the meaning and the vital facts of the "Ground Truth Answer" are present in the "Predicted Answer:"

===Input Data===

- Question: «question»
- Predicted Answer: «LLM_response»
- Ground Truth Answer: «ground_truth_answer»

===Output Format===

Provide your final evaluation in the following format:
"Explanation:" (How you made the decision?)
"Decision:" ("TRUE" or "FALSE")

Please proceed with the evaluation. "" ""

Figure 7: Prompt used to auto-rate the responses of LLM in the experiments. The LLM is provided with questions, model responses, and ground truth answers, along with instructions to check if the model response contains the gold answer.