2024 KDD Cup CRAG Workshop: UM6P Team Technical Report

Firdawse Guerbouzi
College of Computing - UM6P
firdawse.guerbouzi1@gmail.com

Rida Lefdali
College of Computing - UM6P
rida.lefdali@um6p.ma

Hasnae Zerouaoui
College of Computing - UM6P
hasnae.zerouaoui@um6p.ma

Karima Echihabi
College of Computing - UM6P
karima.echihabi@um6p.ma

ABSTRACT

Large Language Models (LLMs) have shown impressive capabilities in understanding and generating coherent natural language, but they still suffer from hallucinations, i.e. answers that seem coherent but are incorrect. Retrieval-Augmented Generation (RAG) aims at reducing hallucinations by grounding the LLMs with external relevant data sources. The Meta KDD Cup 2024 introduced the Comprehensive RAG (CRAG) challenge, which evaluates Question-Answering (QA) systems across five domains and eight question types. The challenge consists of three tasks: Web-Based Retrieval Summarization, Knowledge Graph and Web Augmentation, and End-to-End RAG. This paper summarizes the UM6P Team's participation in the first task. We describe our experimental framework including hyperparameter tuning, sampling strategy selection (to best align the offline and online results), and an extensive evaluation of various RAG pipelines. We also share key insights about the contribution of each RAG component to the overall performance, covering chunking, retrieval, and enhancement techniques. The pipelines were assessed offline using Llama 3 and online using GPT-4 based on the number of correct answers, missing answers, and hallucinations. Our experiments indicate that the best-performing pipeline consists of Facebook AI Similarity Search (FAISS), sentence chunking, re-ranking, and Hypothetical Document Embedding for input enhancement (HyDE), achieving a competitive accuracy score of 0.339 compared to the top score of 0.393, despite a lower overall CRAG score (0.05 vs. 0.204) due to hallucinations (0.288 vs. 0.189). We conclude with a discussion of the main technical and performance challenges encountered during the competition, and some pointers on future research directions.

CCS CONCEPTS

• **Computing methodologies**; • **Information systems**; • **Applied computing**;

KEYWORDS

Large Language Models, Retrieval Augmented Generation, Web-Based, Summarization.

2024 KDD Cup Workshop for Retrieval Augmented Generation, August 25-29, Barcelona, Spain

ACM Reference Format:
Firdawse Guerbouzi, Rida Lefdali, Hasnae Zerouaoui, and Karima Echihabi. 2024. 2024 KDD Cup CRAG Workshop: UM6P Team Technical Report. In *Proceedings of 2024 KDD Cup Workshop for Retrieval Augmented Generation.* ACM, New York, NY, USA, 9 pages.

1 INTRODUCTION

Large Language Models (LLMs) have made significant strides in understanding and generating natural language [15]. Despite these advancements, challenges such as hallucinations—where the model generates information not supported by its training data or context—remain prevalent [3]. As illustrated in Figure 1, using an LLM alone returns the wrong answer to the question "When was the first robot used in surgery", but the correct answer when the LLM is augmented with a search engine. To address these issues, Retrieval-Augmented Generation (RAG) systems have been proposed to enhance LLMs by integrating retrieval mechanisms to fetch relevant information before generating responses [19] [15]. This is particularly important when queries require details that are not included in the model's training data, for example when information extends beyond the model's cut-off date or needs to be retrieved from private data sources. However, RAG systems still face many challenges, including the persistence of hallucinations. This is due in a large part to the extensive tuning and experimentation needed to optimize their performance and minimize errors [3]. Addressing these hallucinations and ensuring the fidelity of retrieved information remain critical areas requiring ongoing research and evaluation [35].

Considerable efforts have been made to advance this field through the development of various benchmarks [6, 9, 10, 23, 26, 30]. For example, the benchmarks in [6, 22] assess LLMs' performance on several critical aspects, including integrative queries that require combining information from multiple documents to generate a coherent response, noisy information where irrelevant or extraneous details can mislead the model, and counterfactual information where the model might generate a correct answer based on internal knowledge but be influenced by incorrect external data. However, they primarily focus on the generation aspect without specifically evaluating the accuracy of retrieval mechanisms. In contrast, [30] introduces a dataset designed for queries that require retrieving and reasoning from multiple pieces of evidence. The study in [9] evaluates the impact of different RAG methods on *Retrieval Precision*, which measures the proportion of relevant content retrieved, and *Answer Similarity*, which rates how closely generated answers match reference answers, providing insights into the effectiveness of various retrieval strategies.

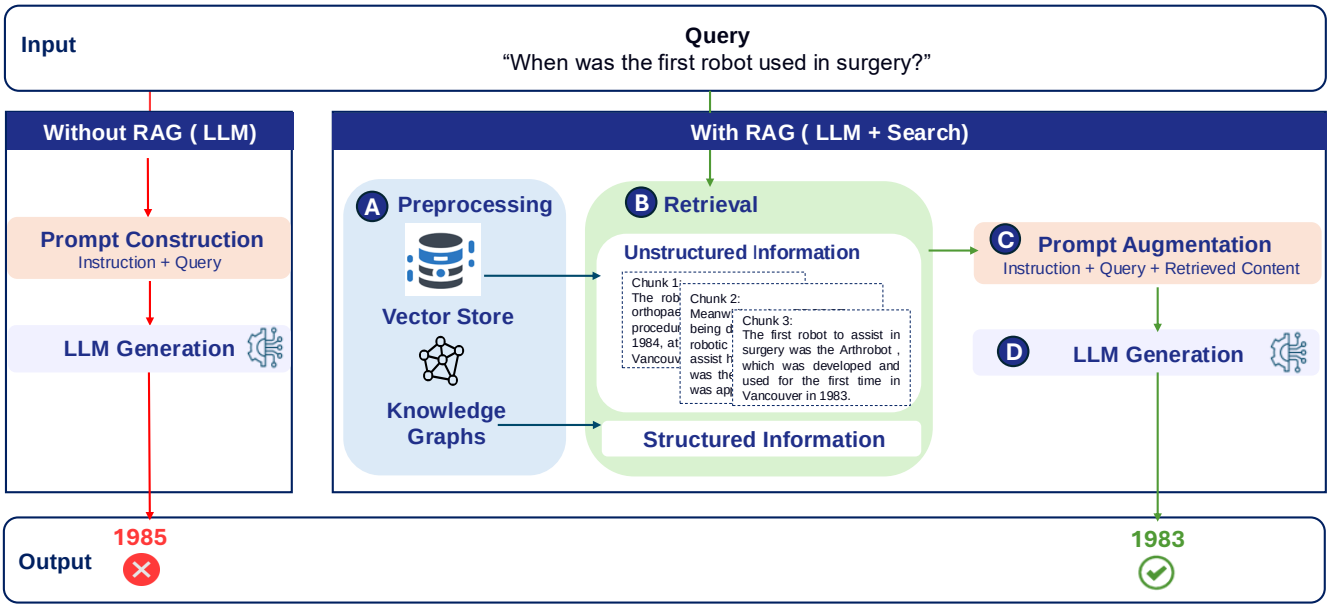

Figure 1: A representative CRAG pipeline that consists of: (A) a pre-processing step which (i) transforms the data (web pages) into chunks, represents each chunk with an embedding, and loads all embeddings into a vector store, and (ii) prepares the pre-built Knowledge Graphs; (B) a retrieval step that returns the top K chunks using a similarity search from web pages and/or structured information from the Knowledge Graphs; (C) a prompt augmentation step that provides the LLM an input that consists of the query and the retrieved information; (D) an LLM Generation step which generates the answer to the query.

Despite these advancements, there are still many open research questions in this field [33]: 1) evaluating the contribution of the retrieval content to the final generated text since current evaluation methods focus primarily on the generation aspect; 2) selecting the most relevant information to ensure accuracy and relevance; 3) reducing QA latency for real-time applications; 4) improving the synthesis of information for complex queries; and 5) developing benchmarks that can adapt to the evolving nature of real-world knowledge sources and assess different parts of RAG systems.

To meet these needs, the CRAG Challenge was initiated providing clear metrics and evaluation protocols to thoroughly assess RAG systems, focusing on two key areas: Web-based summarization pipelines and Knowledge Graph (KG) extraction. Web-based summarization involves distilling vast amounts of unstructured web information into concise, and relevant content tailored to the query, while KG extraction provides structured data (where nodes represent entities and labeled edges the relationships between entities) to enhance the accuracy and relevance of generated content. The CRAG Challenge comprises three tasks: Web-Based Retrieval Summarization (Task 1), KG and Web-Augmentation (Task 2), and End-to-End RAG (Task 3). The tasks are based on the same set of Question-Answering (QA) pairs but differ in the size of the external data available for retrieval, and the nature of the data structures used. In Task 1, the goal is to extract and summarize information from five web pages into accurate answers. Task 2 adds a second data source organized as a KG, and mock Application Programming Interfaces (APIs) to access it. Each domain in this task has its own knowledge graph, requiring retrieval of specific entities to access the appropriate KGs. Task 3 is similar to Task 2, except for the scale of the web pages which is increased to 50 rather than 5 web pages, many of which are irrelevant to the query and considered noise.

To explore these tasks, we investigated different pipelines. We examined the impact of three chunking techniques: Sentence Chunking (SeC), Semantic Chunking (SmC), and Recursive Chunking (RC); three retrieval techniques: BestMatch 25 (BM25), Facebook AI Similarity Search (FAISS), and a combination of both; and other enhancement techniques such as Hypothetical Document Embedding (HyDE) for input enhancement, in addition to re-ranking and Maximal Marginal Relevance (MMR) for retrieval enhancement. We also focused on sampling strategy selection to best align the offline and online results. Additionally, we experimented with LLM agents. Evaluations were conducted using LLM-specific metrics that we describe in more detail in Section 3.3.3

In this technical report, we make the following contributions:

- we present a brief survey of the state-of-the-art RAG benchmarks and describe the main concepts of chunking, retrieval and input enhancement techniques, which led to the best performance as we will demonstrate in Section 2.
- we conduct an experimental evaluation of different RAG pipelines, and study the impact of chunking, retrieval, sampling and enhancement techniques on QA accuracy over the CRAG dataset [1]. We also show how this impact changes depending on the type of the question.
- we share the key insights gained in this study, highlight the main challenges encountered during our participation, and pinpoint some interesting research directions.

- we share our code publicly in [11].

2 PRELIMINARIES & RELATED WORK

Existing surveys [14, 15, 35] provide an overview of the current RAG landscape, categorizing various models based on their architecture and enhancement approaches, and highlighting the importance of robust evaluation methodologies in advancing RAG systems. Evaluating RAG systems is particularly challenging since it requires assessing all the RAG components which are interdependent. These include the embedding model, the chunking, the retrieval, the generation, and the prompt. Despite these contributions, examining all the RAG components and their interactions remains a challenging research problem. Therefore, in this section we start by presenting the RAG pipeline and the different techniques used during our participation in the CRAG competition.

2.1 RAG Systems

RAG marks a notable advancement in Generative AI because it enhances the quality and relevance of the generated answers with the integration of information retrieval techniques [14, 19]. It tackles a key issue of standalone generative models, which is hallucination [3], i.e., generating responses that seem coherent but are not factual. By retrieving relevant information from external sources, RAG systems significantly reduce the occurrence of hallucinations; thus improving the reliability and richness of the generated content. Figure 1 illustrates the three main tasks in the CRAG process, beginning with: the pre-processing step (A) which (i) prepares the retrieval corpus by extracting relevant content from the web pages, dividing them into chunks, representing them as embeddings and loading them into a a vector store, and (ii) connects to the set of pre-built Knowledge Graphs (KGs) provided by the organizers. Then, the retrieval process (B) exploits search to return the top K chunks based from the vector store and/or the KGs according to the query's key entities [35]. Following this, prompt augmentation (C) enhances the prompt with the information retrieved in (B) and passes it to the LLM. Finally, the LLM generates the answer (D).

2.2 Chunking techniques

To enhance the ability of the RAG systems to retrieve accurately the relevant documents, chunking techniques, which segment documents into smaller parts, are important [20]. Applying the appropriate chunking technique can significantly improve the quality of the generated answer [32], as we will demonstrate in Section 4.4. During the CRAG competition, we used three chunking techniques.

(1) *Sentence Chunking (SeC)* [24] divides the text to individual sentences defined by punctuation marks (e.g., periods, exclamation marks, and question marks).
(2) *Semantic Chunking (SmC)* [12] segments the text into semantically meaningful chunks. This is achieved by first splitting the text into sentences, converting these sentences into vector embeddings, and calculating the cosine similarity between them. If the similarity between consecutive segments exceeds a predefined threshold, a split is performed.
(3) *Recursive Chunking (RC)* [28] enables multi-level analysis by hierarchically segmenting text into progressively smaller chunks. This process involves initially splitting the text with a primary separator, such as a paragraph break, and then recursively applying different separators, such as line breaks or commas, until the desired chunk size is reached. This technique is implemented in tools like Langchain's RecursiveCharacterTextSplitter class [18].

Both SeC and RC where implemented using the LangChain framework [18].

2.3 Retrieval Techniques

The retrieval part of a RAG architecture represents a fundamental building block that grounds the LLM with factural information from an external reliable data source. Given a text query, the retrieval returns the embeddings in the data stores that are similar to the query embedding, typically using a distance measure such as the Euclidean distance or cosine similarity [35]. During the competition, we used three retrieval techniques:

(1) *BestMatch 25 (BM25)* [25] is a ranking algorithm that builds on TF-IDF [27] by incorporating term frequency saturation and adjusting for document length, allowing it to rank documents based on query term frequency and rarity in the overall corpus.
(2) *Facebook AI Similarity Search (FAISS)* [8] is a library provided by Meta containing a set of similarity search techniques including inverted indexes, graph-based indexes and scans, for efficient similarity search.
(3) *Hybrid Search* combines different retrieval techniques. During the competition, we combine both FAISS [8] and BM25 [25] through an ensemble retriever. Initially, each method ranks the documents based on its specific criteria: FAISS based on vector similarity, while BM25 based on term frequency and document length. The ensemble retriever utilizes a method called Reciprocal Rank Fusion (RRF) [7] to aggregate these two sets of rankings and re-rank them by applying a weighted sum of the reciprocal ranks.

2.4 Enhancement Techniques

We now explore methods that enhance the performance of RAG systems through two key techniques: input query transformation and retriever enhancement. We first examine an input query transformation technique designed to improve the quality of the query itself [35]. Then, we discuss retriever enhancement methods, such as re-rankers and Maximal Marginal Relevance (MMR), which refine the results to achieve improved diversity and overall performance.

(1) *Hypothetical Document Embedding (HyDE)* [13] is used in NLP and information retrieval as an input query transformation enhancement technique, by creating embeddings based on hypothetical retrieval scenarios that capture the context and meaning of relevant documents.
(2) *Re-rankers* are machine learning models, often based on cross-encoders, that serve as a secondary filter. After the initial retrieval system has collected a set of potentially relevant documents for the user's query, a re-ranker refines this selection based on similarity scores. During the competition, we evaluated the ms-marco-MiniLM-L-2-v2 [16] re-ranker.
(3) *Maximal Marginal Relevanc (MMR)* [5] is an information retrieval technique that ensures the retrieved documents are

both relevant to the user's query and diverse. It achieves relevance by evaluating how closely each document matches the query, and it reduces redundancy by penalizing documents that are too similar to those already selected.

2.5 Generation

The generation component of the RAG architecture is also a key building block. LLMs are advanced AI systems designed to understand and produce human language [14]. At their core, LLMs rely on a transformer architecture, which processes sequential data more efficiently [31]. They use an encoder-decoder framework to uncover statistical relationships between text tokens, which are individual units of text (not necessarily individual words), represented as high-dimensional vectors through embeddings. Following the competition guidelines, predictions were generated using Llama 3 following the prompt in Figure 2. Then we evaluated the generated answers offline with Llama 3 [34], and online with GPT-4.

Prompt

You are a Q&A assistant. Your goal is to answer questions as accurately as possible based on the instructions and context provided without using prior knowledge.
Analyze the context carefully and provide a direct, concise, and as short as possible answer without explanation.
If the question is based on a false premise or assumption, respond with 'invalid question".
If the answer is not in the context, just say 'I don't know.' Don't make up an answer.
Question: {question}
Context: {context}

Figure 2: Sample prompt used to guide the LLM

3 THE META KDD CUP CRAG CHALLENGE: UM6P TEAM'S APPROACH

This section describes the experimental framework we followed during the CRAG competition, focusing on Task 1 which uses 5 web pages as a data source.

3.1 Empirical Design

Figure 3 summarizes our approach, which involved tuning hyperparameters such as temperature, chunk size, and the number of retrieved documents; identifying the best sampling strategy to align with online results; and conducting progressive studies to determine the most effective pipeline. Our evaluation process included comparing three chunking techniques—SeC, SmC, and RC—and assessing three retrieval techniques: BM25, FAISS, and hybrid methods. Additionally, we explored three enhancement techniques: HyDE for input enhancement, and MMR and re-ranker for retrieval enhancement. The results were evaluated based on the number of correct answers, missing answers, hallucinations, and the CRAG score [33].

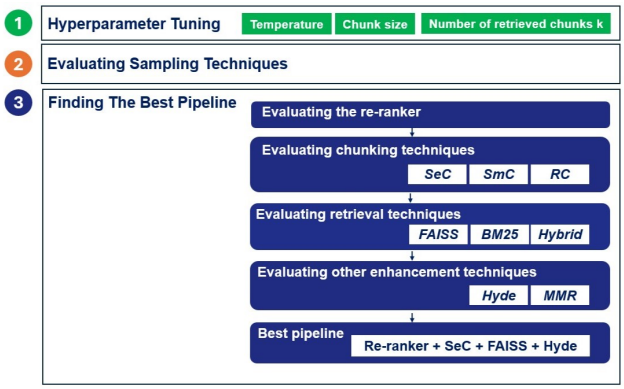

Figure 3: Experimental framework

3.2 Environment

Experiments were first conducted locally using one NVIDIA A100-SXM4 GPU with 80GB of RAM. Once the local results were satisfactory, the solution was submitted for online evaluation on a setup with four NVIDIA T4 GPUs, each equipped with 16GB of GPU memory. Each solution consists of an end-to-end RAG pipeline.

3.3 Experimental Setup

3.3.1 ***Dataset.*** The CRAG challenge [33] provides several datasets designed for evaluating RAG systems on three different tasks, spanning five domains: Finance, Sports, Music, Movies, and Open Data. To support retrieval tasks, the dataset incorporates:

- Web Search Results: Each question is used to retrieve up to 50 HTML pages from the Brave search API. The proportion of relevant answers found among these pages was 84% for web-based questions and 63% for KG-based questions [33].
- KGs: Includes mock KGs created from publicly available KG data, featuring randomly selected entities of the same type, and also "hard negative" entities with similar names.
- Mock APIs: Predefined mock APIs are provided to facilitate structured searches within the mock KGs, such as a `get_price_history(ticker)` API for stock price queries.

The dataset encompasses 220,000 web pages, a KG with 2.6 million entities, and 38 mock APIs. While the dataset supports all tasks, this report concentrates on the first task as we encountered challenges in submitting an agent-based solution for the other two tasks.

3.3.2 ***Queries.*** The CRAG challenge proposes a diverse set of queries to evaluate RAG systems. It covers simple queries, queries with conditions, set-based queries, comparison queries, aggregation queries, multi-hop queries, post-processing heavy queries, and queries with false premises. The variety in query types allows a comprehensive assessment of retrieval systems, providing insights into the strengths and weaknesses of various methods in real-world applications. The queries consist of 2,425 QA pairs sourced from web content and 1,984 QA pairs derived from KGs.

3.3.3 ***Measures.*** Evaluating LLMs is an ongoing research challenge and a crucial component of the RAG pipeline. The CRAG

challenge suggests several metrics to assess the factual accuracy and reliability of the generated text [33]:

- *Hallucination* refers to instances where the LLM produces text containing incorrect or fabricated information that is neither supported by the context nor by external sources.
- *Accuracy* measures how closely the generated answer aligns with the ground truth in content, structure, and intent.
- *Exact Accuracy* refers to the situation where the generated answer exactly matches the reference answer.
- *Missing* refers to a response that is vague or unhelpful (e.g., "I do not know," "I am sorry, I cannot find the answer"), an empty response due to a system error, or a request for clarification of the original question.
- *CRAG score* is an aggregated score that considers accuracy, missing values, and hallucinations. It applies a penalty for the latter as they can undermine user trust significantly. The CRAG score is computed using the following formula, n being the number of queries:

$$\text{CRAG Score} = \frac{2 \times \text{n}_{\text{correct}} + \text{n}_{\text{miss}}}{n} - 1$$

The quality of generated answers (accurate, missing, or hallucinated) is evaluated using an LLM. Although the online evaluation uses GPT-4, we opted for Llama 3 in the offline evaluation for cost considerations.

3.3.4 ***Preprocessing.*** The web pages selected for the retrieval corpus required parsing to enable effective chunking and information retrieval. We utilized the Python library *BeautifulSoup4* [4] for this purpose. Beautiful Soup parses HTML pages and constructs a traversable tree structure. To extract text content efficiently, we concatenated the text from each web page and removed any extraneous whitespaces.

4 RESULTS & DISCUSSION

In this section, we report our experimental results. We began the competition by evaluating the impact of the initial pipeline, which used the FAISS retriever and the RC technique, considered as our baseline. We then assessed the impact of incorporating a retrieval enhancement technique using a re-ranker. Once the best pipeline is selected, we continued our evaluation by implementing various chunking techniques to identify the most effective approach and explored different retrieval techniques to refine the pipeline. Additionally, the effects of introducing supplementary inputs and enhancing the retriever were examined to optimize overall performance.

4.1 Hyperparameter Tuning

We consistently used the Llama 3 model, fine-tuning various parameters, including the temperature, which we ultimately set to 0.3. In our experiments with chunking techniques, we first focused on RC to determine the optimal chunk size and number of retrieved chunks. As illustrated in Figure 4, the best CRAG score was achieved with 5 retrieved documents and a chunk size of 256. Increasing the chunk size or the number of retrieved documents beyond these values caused the CRAG to decline. This suggests that the LLM benefits from a more concise context to minimize hallucinations.

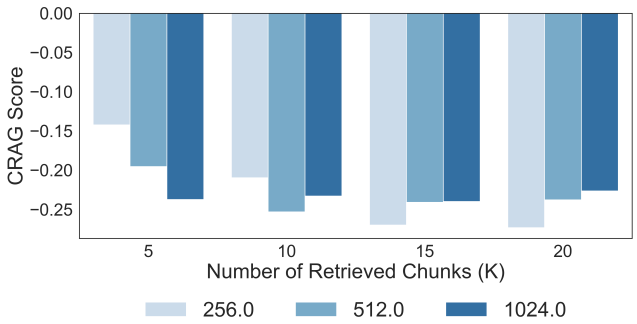

Figure 4: Impact of the number of retrieved documents

Following our analysis of chunking techniques, we turned to evaluate the impact of the number of retrieved documents per question type, as shown in Figure 5. We maintained a chunk size of 256, which had previously yielded the best CRAG score. Our findings indicate that increasing the number of retrieved documents generally reduces the CRAG score for most question types, with the exception of "set" and "false premise." Specifically, question types such as "simple," "simple with condition," and "aggregation" showed significant performance drops when the number of retrieved documents K was very high. We believe that this may be attributed to the lost-in-the-middle [21] problem, where the LLM struggles to maintain focus and perform accurate reasoning when confronted with excessive information. As the volume of retrieved documents increases, the model may become overwhelmed and lose track of relevant details, leading to diminished performance.

Next, we examined the effect of chunk size on performance, fixing K to 5. Figure 6 reveals that performance decreases as chunk size increases for most question types, except for "false premise" and "set". This indicates that the chunk size parameter should also be adapted to the question type to achieve the best results.

Additionally, for "false premise" cases, we observed that the score improves with more retrieved documents and larger chunk size. This might be because the LLM, already prompted to handle false assumptions as shown in Figure 2, becomes more confident in incorrectly identifying false premises when presented with additional, potentially confusing information. Overall, we chose K to be 5 with a chunk size of 256 for RC as it provided the best results for most cases and fit within our time constraints; however, a more detailed analysis by query type and a query-adaptive selection of these parameters could have further enhanced performance.

We employed a similar methodology to assess various chunking techniques and determine the optimal hyperparameters. For SmC, the size of chunks depends on the semantic relationship between sentences. This approach has some disadvantages, as larger chunks can exceed the model's context size, leading to submission failures. Based on our experiments, using 10 sentences proved to be the most reliable. In the case of SeC, the best performance was achieved with 20 sentences. These configurations were chosen for their ability to enhance the CRAG score. For the hybrid retriever, which combines keyword and semantic retrieval approaches, we assigned a weight of 0.5 to each type.

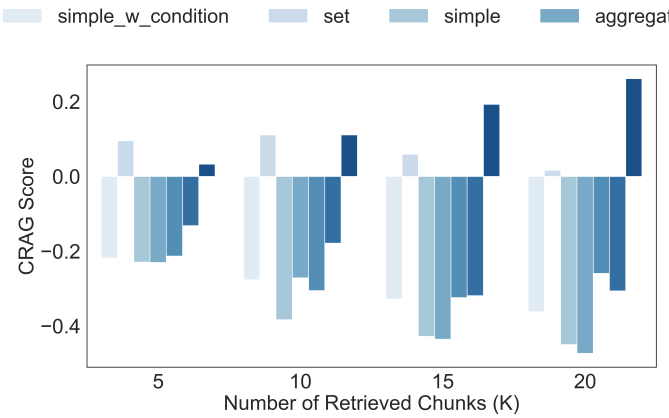

Figure 5: Impact of K per question type

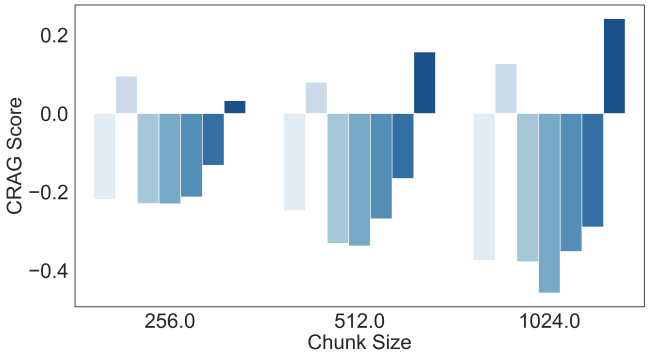

Figure 6: Impact of chunk size per question type

4.2 Evaluating Sampling Techniques

A significant challenge encountered during the competition was the limited number of online submissions allowed per week, which made it essential to align our internal evaluations with the online results as accurately as possible. To address this, three sampling techniques were tested for selecting 333 queries, consistent with the number used in the online evaluation. (1) The first technique selects the top 333 queries. (2) The second selects 25% of queries from each question type. (3) The third creates three distinct non-overlapping subsets of 333 queries, and averages results across these subsets. The absolute error between the offline and online results was calculated for each method. As shown in Table 1, the second technique yielded the lowest absolute error of 0.042, compared to 0.087 for the first and 0.094 for the third. This led to the adoption of the second technique for subsequent experiments. However, some inconsistencies between the offline and online results persisted. This discrepancy is explored in detail in the challenges section, where we examine potential factors contributing to these differences.

Table 1: Evaluation of Different Sampling Strategies

Pipeline	Sampling Strategy	Offline Result	Online Result	Abs. Err.
BM25 + SeC + Re-ranker	(1)	0.051	-0.036	0.087
	(2)	0.006	-0.036	0.042
	(3)	0.058	-0.036	0.094

4.3 Evaluating the Re-ranker

To evaluate the re-ranker, we fix the semantic retriever to be the FAISS flat index, and the chunking approach to be RC per [29]. As shown in Table 2, this initial setup achieved a CRAG score of -0.05, with an accuracy of 0.224 and a hallucination rate of 0.272. These metrics are crucial as they are the main components of the CRAG score, reflecting the performance of a robust RAG system. Building on existing research indicating the pivotal role of re-rankers in enhancing RAG systems by refining retrieved documents [2, 9], we incorporate a re-ranker as a second-stage retriever. This modification resulted in a slight improvement in the CRAG score, bringing it to -0.0118. This improvement was primarily due to a reduction in the hallucination rate from 0.272 to 0.259.

4.4 Evaluating Chunking Techniques

Following the evaluation of the re-ranker, we focus on refining chunking techniques, which are critical for enhancing the efficiency and accuracy of the retrieval process. Effective chunking significantly impacts the overall performance of RAG models in several areas, including retrieval efficiency, accuracy, relevance, scalability, manageability, and balanced information distribution [9, 20, 32].

We compare three chunking methods: RC, SeC, and SmC. The results, presented in Table 3, reveal that while RC achieves the highest accuracy of 0.249, it also incurs the highest hallucination score of 0.333, resulting in a CRAG score of -0.008. In contrast, SeC provides the best balance between accuracy and hallucination, leading to an improved CRAG score of 0.009. On the other hand, SmC leads to the lowest CRAG score, at -0.093. These results were somewhat unexpected. Despite its simplicity, SeC outperformed the more complex methods. This suggests that SeC's straightforward approach maintains clarity and relevance better, reducing model confusion. RC's high accuracy came at the cost of increased hallucinations, while SmC's attempt to combine related information might have reduced its effectiveness. This suggests that either simpler chunking methods are satisfactory or that more effective chunking techniques should be devised.

4.5 Evaluating Retriever Techniques

Building on insights gained from previous experiments, which highlight improvements in the CRAG scores with the re-ranker and SeC, we now investigate the impact of retriever techniques, which are fundamental building blocks for RAG architectures. We evaluate three types of retrievers: a semantic retriever using FAISS, a keyword-based retriever using BM25, and a hybrid retriever combining both FAISS and BM25. As detailed in Table 4, the performance of each retriever varies. BM25 achieves the highest accuracy at 0.297 but has a high hallucination rate of 0.333, resulting in a

Table 2: Impact of The re-ranker

Pipeline	Exact Accuracy		Accuracy		Hallucination		Missing		CRAG score	
	Offline	Online	Offline	Online	Offline	Online	Offline	Online	Offline	Online
FAISS + RC (baseline)	0.0	0.047	0.348	0.224	0.308	**0.272**	0.344	0.5	0.04	-0.0472
FAISS + RC + Re-ranker	**0.028**	**0.039**	**0.276**	**0.224**	**0.248**	**0.259**	**0.476**	**0.5**	**0.028**	**-0.0118**

Table 3: Impact of The Chunking approaches

Pipeline	Exact Accuracy		Accuracy		Hallucination		Missing		CRAG score	
	Offline	Online	Offline	Online	Offline	Online	Offline	Online	Offline	Online
FAISS + RC + re-ranker	0.024	0.015	0.34	0.249	0.25	0.333	0.402	0.417	0.087	-0.00841
FAISS + SmC + re-ranker	0.04	0.03	0.268	0.234	0.327	0.248	0.448	0.438	0.162	-0.093
FAISS + Sec+ re-ranker	**0.003**	**0.006**	**0.273**	**0.243**	**0.225**	**0.234**	**0.501**	**0.523**	**0.048**	**0.009**

CRAG score of -0.036. This suggests that BM25, while effective at retrieving relevant information, may generate more hallucinations due to its focus on keyword matching, which can lead to less contextually relevant results and increased errors. In contrast, the FAISS retriever achieves a CRAG score of 0.009, demonstrating a better balance between accuracy and hallucination. Surprisingly, the hybrid retriever, which combines both FAISS and BM25, results in the lowest CRAG score of -0.042. This could suggest that combining the two methods might introduce conflicting information or fail to leverage the strengths of either approach effectively. The hybrid approach may be diluting the advantages of each individual retriever or complicating the retrieval process, leading to decreased overall performance. These results highlight the need for further investigation into how combining retrieval methods impacts performance and suggest that optimizing individual retrievers or refining their integration could improve results. Overall, the FAISS retriever provides the most robust performance, offering an effective compromise between accuracy and hallucination.

4.6 Evaluating Other Enhancement Techniques

Amongst the different pipelines we experimented with, the one that leads to the best online CRAG score is FAISS combined with SeC and the ms-marco-MiniLM-L-2-v2 re-ranker. To further enhance the pipeline performance, we explore additional techniques, including Hypothetical Document Embedding (HyDE) [13]. This method generates hypothetical answers, using an LLM without external knowledge, and then extracts similar answers from the retrieval corpus. The underlying assumption is that answers are more closely related to each other in semantic space than the relationship between queries and their corresponding answers. As shown in Table 5, integrating HyDE significantly improves accuracy from 0.243 to 0.339, leading to a better CRAG score of 0.05. This indicates that generating hypothetical answers can effectively capture relevant semantic similarities, enhancing retrieval quality. We also investigate MMR, a technique used to select documents that are both relevant and diverse [5]. However, in our online evaluation, MMR does not yield positive results. Instead, it increases the hallucination rate from 0.288 to 0.339, suggesting that while MMR aims to diversify retrieved results, it may have introduced more noise rather than improving overall performance [9]. One additional observation is the misalignment between offline evaluations with Llama 3 and online evaluations with GPT models across all tables. This discrepancy underscores a significant challenge in aligning open model evaluations with closed models like GPT. To address these differences more accurately, incorporating human evaluation could provide a more precise analysis of the discrepancies and help bridge the gap between different evaluation methodologies.

4.7 Summary

In this report, we summarize the results of an experimental evaluation that we conducted in the context of the 2024 KDD Cup CRAG competition. We evaluate the various components of RAG architectures to assess their individual and collective impacts on system performance. We cover the retriever, the chunking, the re-ranking and other enhancement techniques. Our findings underscore the critical role of each of these components in shaping the overall efficacy of the RAG system. The re-ranker, in particular, is highly effective, as evidenced by its ability to improve performance metrics and reduce the hallucination rate, thereby enhancing the CRAG score. This success prompted further exploration of additional re-ranker models to optimize performance even further. We observe that the HyDE Enhancement significantly boosts accuracy by 39%, improving it from 0.243 to 0.339. On the other hand, other methods, like MMR, can have an adverse effect, increasing the hallucination rate. These results highlight that the effectiveness of enhancement techniques can vary considerably based on factors such as the type, size, and format of the retrieval corpus and the nature of the queries. During the competition, the best-performing pipeline utilized is composed of the FAISS retriever, the SeC, the re-ranker, and HyDE, achieving the highest CRAG score of 0.05 with improved accuracy

Table 4: Impact of The Retriever approaches

Pipeline	Exact Accuracy		Accuracy		Hallucination		Missing		CRAG score	
	Offline	Online	Offline	Online	Offline	Online	Offline	Online	Offline	Online
FAISS + SeC+ re-ranker	**0.003**	**0.006**	**0.273**	**0.243**	**0.225**	**0.234**	**0.501**	**0.523**	**0.048**	**0.009**
BM25 +SeC+ re-ranker	0.012	0.009	0.369	0.297	0.264	0.333	0.366	0.369	0.105	-0.036
Hybrid +SeC+ re-ranker	0.0	0.015	0.368	0.213	0.288	0.255	0.344	0.532	0.08	-0.042

Table 5: Impact of The Enhancement techniques

Pipeline	Exact Accuracy		Accuracy		Hallucination		Missing		CRAG score	
	Offline	Online	Offline	Online	Offline	Online	Offline	Online	Offline	Online
FAISS +SeC+ re-ranker	0.003	0.006	0.273	0.243	0.225	0.234	0.501	0.523	0.048	0.009
FAISS + SeC+ re-ranker + HyDE	**0.02**	**0.024**	**0.342**	**0.339**	**0.255**	**0.288**	**0.40**	**0.372**	**0.08**	**0.0511**
FAISS +SeC+ re-ranker + HyDE + MMR	0.0	0.024	0.428	0.336	0.232	0.339	0.32	0.324	0.19	-0.003

and reduced hallucination rates. Consequently, while certain techniques and enhancements can lead to improved RAG performance, their impact is highly context-dependent. This necessitates a nuanced evaluation within the specific application scenario to fully understand and leverage their potential benefits.

5 CHALLENGES

In this report, we analyze the main experimental results of different RAG pipelines evaluated internally to select the best solution for submission to the 2024 KDD Cup CRAG competition. Throughout this process, we encountered several challenges, categorized into technical difficulties related to the submission process and performance issues with the methods we tested.

Technical challenges include: (1) The limitation on the number of submissions allowed per week, which restricted our ability to iteratively refine and submit multiple pipelines. (2) Submission issues with Task 2, where we could not submit a competitive pipeline that uses an agent-based solution on KGs: an LLM agent identifies suitable APIs for specific queries and extracts relevant keywords, while a supervisor agent classifies queries by domain and directs them to the appropriate sub-agent. This solution required a context length exceeding the 8K tokens allowed by Llama 3 [34]. This constraint limited Llama 3's capacity to generate a comprehensive chain of thought and perform multiple reasoning steps. However, our offline evaluation with Mistral [17], which has a context limit of 32k tokens, showed improved performance. (3) Time and resource constraints, which prevented us from evaluating different embedding models and re-ranking techniques to further enhance the retrieval process and overall performance.

Performance challenges include: (4) The need for a detailed analysis based on query types, as specific enhancement techniques and RAG components may be closely related to query complexity. (5) Discrepancies between the scores reported by the online evaluation using GPT-4 and the offline evaluation based on Llama 3. We believe these inconsistencies could stem from differences in model reasoning capabilities and the inherent randomness of the evaluation sets.

6 CONCLUSION

Participation in this competition has been a highly rewarding experience, despite not securing a win. It provided a valuable opportunity to contribute to and delve deeply into one of the most complex and impactful areas of AI. Significant insights were gained into the key components of RAG systems and their impact on performance. We plan to leverage these insights to propose solutions that address the challenges faced, and improve the performance of RAG systems. In particular, we plan to (i) extend our experimental evaluation to cover additional datasets, queries, LLMs, embedding models and enhancement techniques to generalize our findings; and (ii) develop a scalable hybrid retriever technique that fuses vector search and keyword search into a single operation, dynamically adapting the weight of each type of search in the final ranking to the dataset and query workload.

7 ACKNOWLEDGMENTS

We extend our thanks to Meta and the ACM for organizing the KDD Cup 2024 CRAG Challenge, which offered a stimulating and insightful experience. We also acknowledge the UM6P Toubkal Supercomputer and the NVIDIA AI Technology Center (NVAITC) EMEA for their valuable access to GPUs.

REFERENCES

[1] AIcrowd. 2024. Meta Comprehensive RAG Benchmark KDD Cup 2024. https://www.aicrowd.com/challenges/meta-comprehensive-rag-benchmark-kdd-cup-2024. Accessed: 2024-06-07.

[2] Nicholas Ampazis. 2024. Improving RAG Quality for Large Language Models with Topic-Enhanced Reranking. In *Artificial Intelligence Applications and Innovations*, Ilias Maglogiannis, Lazaros Iliadis, John Macintyre, Markos Avlonitis, and Antonios Papaleonidas (Eds.). Springer Nature Switzerland, Cham, 74–87.

[3] Zechen Bai, Pichao Wang, Tianjun Xiao, Tong He, Zongbo Han, Zheng Zhang, and Mike Zheng Shou. 2024. Hallucination of Multimodal Large Language Models: A Survey. arXiv:2404.18930 [cs.CV]

[4] Beautiful Soup. 2024. *Beautiful Soup 4 Documentation.* https://pypi.org/project/beautifulsoup4/ Accessed: 2024-08-19.

[5] Jaime Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Melbourne, Australia) *(SIGIR '98)*. Association for Computing Machinery, New York, NY, USA, 335–336. https://doi.org/10.1145/290941.291025

[6] Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2023. Benchmarking Large Language Models in Retrieval-Augmented Generation. arXiv:2309.01431 [cs.CL]

[7] Gordon Cormack, Charles Clarke, and Stefan Büttcher. 2009. Reciprocal Rank Fusion Outperforms Condorcet and Individual Rank Learning Methods. In *International Conference on Research and Development in Information Retrieval (SIGIR)*. https://doi.org/10.1145/1571941.1572114

[8] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The Faiss library. arXiv:2401.08281 [cs.LG] https://arxiv.org/abs/2401.08281

[9] Matouš Eibich, Shivay Nagpal, and Alexander Fred-Ojala. 2024. ARAGOG: Advanced RAG Output Grading. arXiv:2404.01037 [cs.CL]

[10] Shahul Es, Jithin James, Luis Espinosa-Anke, and Steven Schockaert. 2023. RAGAS: Automated Evaluation of Retrieval Augmented Generation. arXiv:2309.15217 [cs.CL]

[11] Firdawse. 2024. *UM6P Team Meta KDD 2024 CUP.* https://github.com/firdawse21/UM6P-Team-Meta-KDD-2024-CUP.git Accessed: 2024-08-09.

[12] FullStackRetrieval. 2024. 5 Levels of Text Splitting. https://github.com/FullStackRetrieval-com/RetrievalTutorials/blob/main/tutorials/LevelsOfTextSplitting/5_Levels_Of_Text_Splitting.ipynb Accessed: 2024-08-21.

[13] Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2022. Precise Zero-Shot Dense Retrieval without Relevance Labels. arXiv:2212.10496 [cs.IR]

[14] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-Augmented Generation for Large Language Models: A Survey. arXiv:2312.10997 [cs.CL]

[15] Yizheng Huang and Jimmy Huang. 2024. A Survey on Retrieval-Augmented Text Generation for Large Language Models. arXiv:2404.10981 [cs.IR]

[16] Hugging Face. 2024. Cross-encoder/ms-marco-MiniLM-L-2-v2. https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-2-v2. Accessed: 2024-08-09.

[17] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. arXiv:2310.06825 [cs.CL] https://arxiv.org/abs/2310.06825

[18] Aarushi Kansal. 2024. LangChain: Your Swiss Army Knife. In *Building Generative AI-Powered Apps: A Hands-on Guide for Developers*. Springer, 17–40.

[19] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. arXiv:2005.11401 [cs.CL] https://arxiv.org/abs/2005.11401

[20] Demiao Lin. 2024. Revolutionizing Retrieval-Augmented Generation with Enhanced PDF Structure Recognition. arXiv:2401.12599 [cs.AI] https://arxiv.org/abs/2401.12599

[21] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the Middle: How Language Models Use Long Contexts. arXiv:2307.03172 [cs.CL] https://arxiv.org/abs/2307.03172

[22] Yi Liu, Lianzhe Huang, Shicheng Li, Sishuo Chen, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. 2023. RECALL: A Benchmark for LLMs Robustness against External Counterfactual Knowledge. arXiv:2311.08147 [cs.CL]

[23] Yuanjie Lyu, Zhiyu Li, Simin Niu, Feiyu Xiong, Bo Tang, Wenjin Wang, Hao Wu, Huanyong Liu, Tong Xu, Enhong Chen, Yi Luo, Peng Cheng, Haiying Deng, Zhonghao Wang, and Zijia Lu. 2024. CRUD-RAG: A Comprehensive Chinese Benchmark for Retrieval-Augmented Generation of Large Language Models. arXiv:2401.17043 [cs.CL]

[24] J. Mehul. 2023. RAG Part 2: Chunking. https://medium.com/@j13mehul/rag-part-2-chunking-8b68006eefc1 Accessed: 2024-08-22.

[25] Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval* 3, 4 (2009), 333–389.

[26] Jon Saad-Falcon, Omar Khattab, Christopher Potts, and Matei Zaharia. 2024. ARES: An Automated Evaluation Framework for Retrieval-Augmented Generation Systems. arXiv:2311.09476 [cs.CL]

[27] Claude Sammut and Geoffrey I. Webb (Eds.). 2010. *TF–IDF*. Springer US, Boston, MA, 986–987. https://doi.org/10.1007/978-0-387-30164-8_832

[28] Spurthi Setty, Katherine Jijo, Eden Chung, and Natan Vidra. 2024. Improving Retrieval for RAG based Question Answering Models on Financial Documents. arXiv:2404.07221 [cs.IR]

[29] Spurthi Setty, Harsh Thakkar, Alyssa Lee, Eden Chung, and Natan Vidra. 2024. Improving Retrieval for RAG based Question Answering Models on Financial Documents. arXiv:2404.07221 [cs.IR] https://arxiv.org/abs/2404.07221

[30] Yixuan Tang and Yi Yang. 2024. MultiHop-RAG: Benchmarking Retrieval-Augmented Generation for Multi-Hop Queries. arXiv:2401.15391 [cs.CL]

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention Is All You Need. arXiv:1706.03762 [cs.CL]

[32] Xiaohua Wang, Zhenghua Wang, Xuan Gao, Feiran Zhang, Yixin Wu, Zhibo Xu, Tianyuan Shi, Zhengyuan Wang, Shizheng Li, Qi Qian, Ruicheng Yin, Changze Lv, Xiaoqing Zheng, and Xuanjing Huang. 2024. Searching for Best Practices in Retrieval-Augmented Generation. arXiv:2407.01219 [cs.CL] https://arxiv.org/abs/2407.01219

[33] Xiao Yang, Kai Sun, Hao Xin, Yushi Sun, Nikita Bhalla, Xiangsen Chen, Sajal Choudhary, Rongze Daniel Gui, Ziran Will Jiang, Ziyu Jiang, Lingkun Kong, Brian Moran, Jiaqi Wang, Yifan Ethan Xu, An Yan, Chenyu Yang, Eting Yuan, Hanwen Zha, Nan Tang, Lei Chen, Nicolas Scheffer, Yue Liu, Nirav Shah, Rakesh Wanga, Anuj Kumar, Wen tau Yih, and Xin Luna Dong. 2024. CRAG – Comprehensive RAG Benchmark. arXiv:2406.04744 [cs.CL] https://arxiv.org/abs/2406.04744

[34] Peitian Zhang, Ninglu Shao, Zheng Liu, Shitao Xiao, Hongjin Qian, Qiwei Ye, and Zhicheng Dou. 2024. Extending Llama-3's Context Ten-Fold Overnight. arXiv:2404.19553 [cs.CL]

[35] Penghao Zhao, Hailin Zhang, Qinhan Yu, Zhengren Wang, Yunteng Geng, Fangcheng Fu, Ling Yang, Wentao Zhang, Jie Jiang, and Bin Cui. 2024. Retrieval-Augmented Generation for AI-Generated Content: A Survey. arXiv:2402.19473 [cs.CV]

Received ; revised ; accepted