Enhancing Large Language Models with Retrieval-Augmented Generation: A Case Study on Movie Data Beyond the Training Cutoff

Marcel Mikołajczyk [1]

[1]Warsaw University of Technology, Faculty of Electrical Engineering

Introduction

Large Language Models (LLMs) have transformed Natural Language Processing (NLP) with their ability to generate fluent, human-like responses and reason across complex tasks. Despite their strength, LLMs face some limitations. They cannot access knowledge beyond their training cutoff, and they may hallucinate, generating incorrect information. These issues are especially problematic in domains that evolve rapidly or require precise facts.

Retrieval-Augmented Generation (RAG) addresses these challenges by equipping LLMs with non-parametric memory. It retrieves relevant external data and provides it to the LLM, improving the accuracy and reliability of the response. Various RAG paradigms exist, from basic keyword-based retrieval to advanced approaches using dense embeddings, modular pipelines, knowledge graphs, and adaptive agent-based strategies.

This study explores how RAG enhances LLM performance in answering questions about movies released in 2024, which fall beyond the model's knowledge cutoff.

Methodology

The RAG system uses a dataset of 14.736 movies and TV shows released in 2024, collected from the OMDb API, enabling the Llama 3.2 3B model to handle queries about movies beyond its knowledge. The data is stored in JSON format and loaded into ChromaDB, with plot embeddings generated using the all-MiniLM-L6-v2 model. Metadata supports attribute-based searches, while embeddings enable semantic search based on plot similarities.

The system follows a similar approach to that proposed by Jeong et al.[2], in which an LLM classifies the complexity of the query and determines the appropriate retrieval strategy. Additionally, another LLM that acts as a critic[1] is implemented in order to reduce hallucinations in certain areas. The workflow of RAG system is presented in Figure 1.

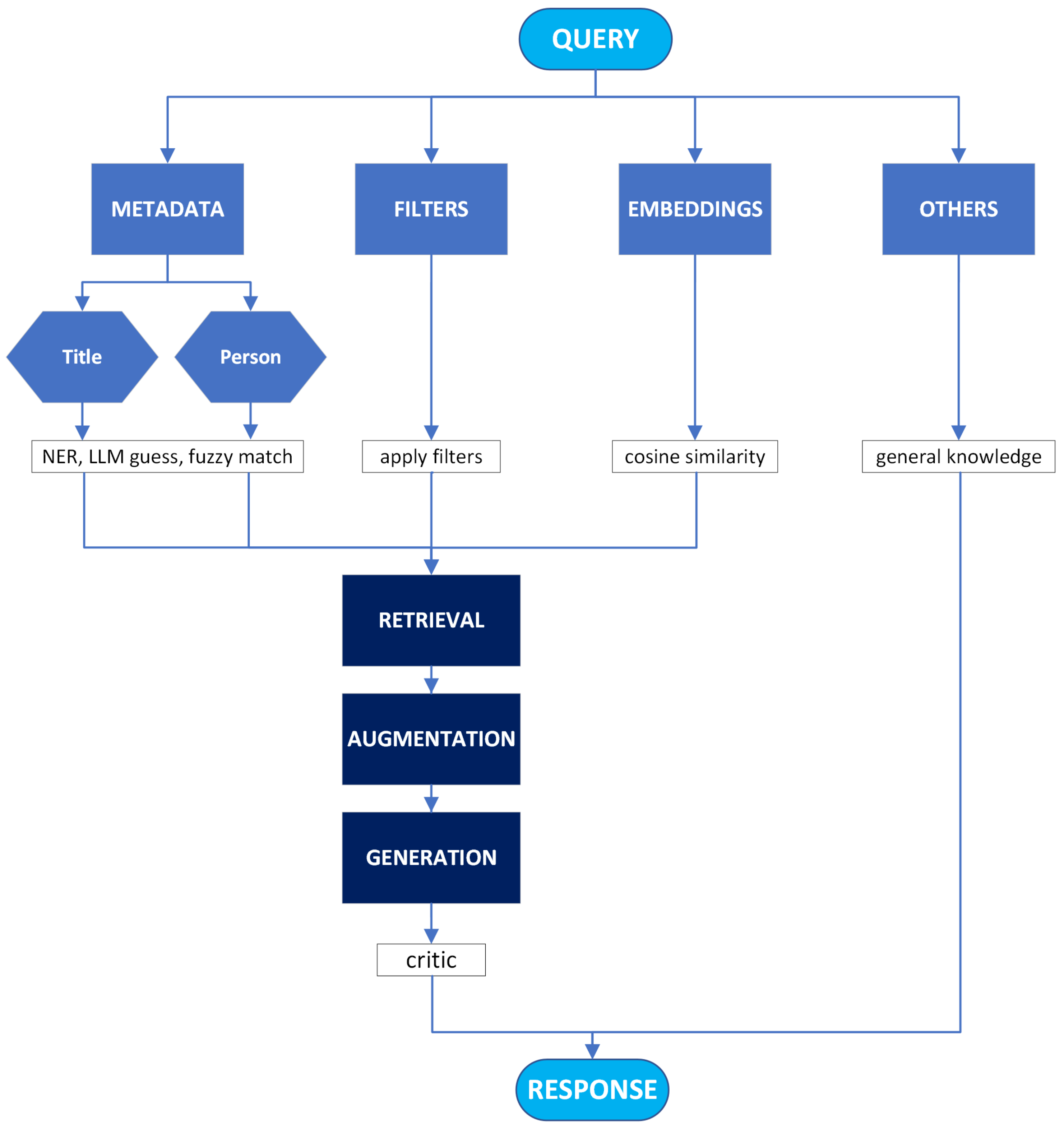
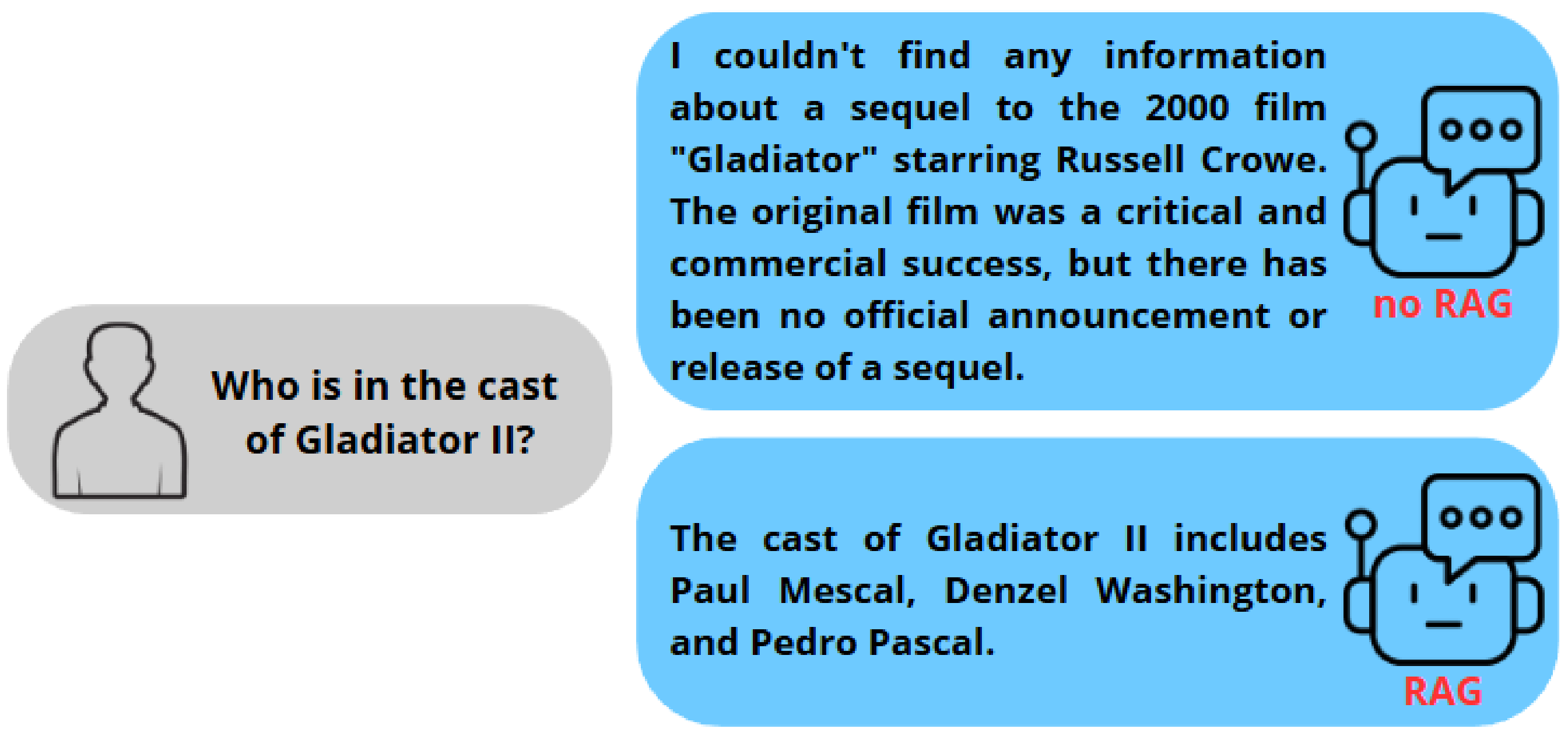


Figure 1. RAG Workflow

The pipeline consists of query classification, retrieval, augmentation, generation, and evaluation by critic before final response. Queries are classified into four types: metadata (specific attributes), filters (conditions on attributes), embeddings (semantic similarity), and others (LLM base knowledge). An example of the LLM's response with and without RAG is shown in Figure 2.



Who is in the cast of Gladiator II?

I couldn't find any information about a sequel to the 2000 film "Gladiator" starring Russell Crowe. The original film was a critical and commercial success, but there has been no official announcement or release of a sequel.

The cast of Gladiator II includes Paul Mescal, Denzel Washington, and Pedro Pascal.

Figure 2. LLM response with and without RAG

Experiments & Results

To evaluate system performance, experiments were conducted across different query types using subsets of the entire movie dataset. All results were manually evaluated based on correctness and absence of hallucinations:

- **Metadata, title-based queries:** Ten batches of 100 randomly selected movie titles were used to assess accuracy in answering prompts like "Who directed ... ?" or "What is the main theme of ... ?".
- **Metadata, actor-based queries:** Similar testing methodology as title-based queries, with prompts like "Can you name movies with ...?" or "What are some movies featuring ...?".
- **Filters-based queries:** 100 prompts requiring multi-attribute filtering (e.g., "Recommend me 3 action movies with Rotten Tomatoes rating above 6") were tested.
- **Embeddings-based queries:** 100 prompts focused on semantic similarity (e.g., "Can you list a few movies similar to Gladiator II?").

Figure 3 summarizes the system's accuracy across the tested queries. While metadata-based queries yielded high accuracy, more complex reasoning in filter-based and embedding-based queries resulted in reduced performance due to hallucinations.

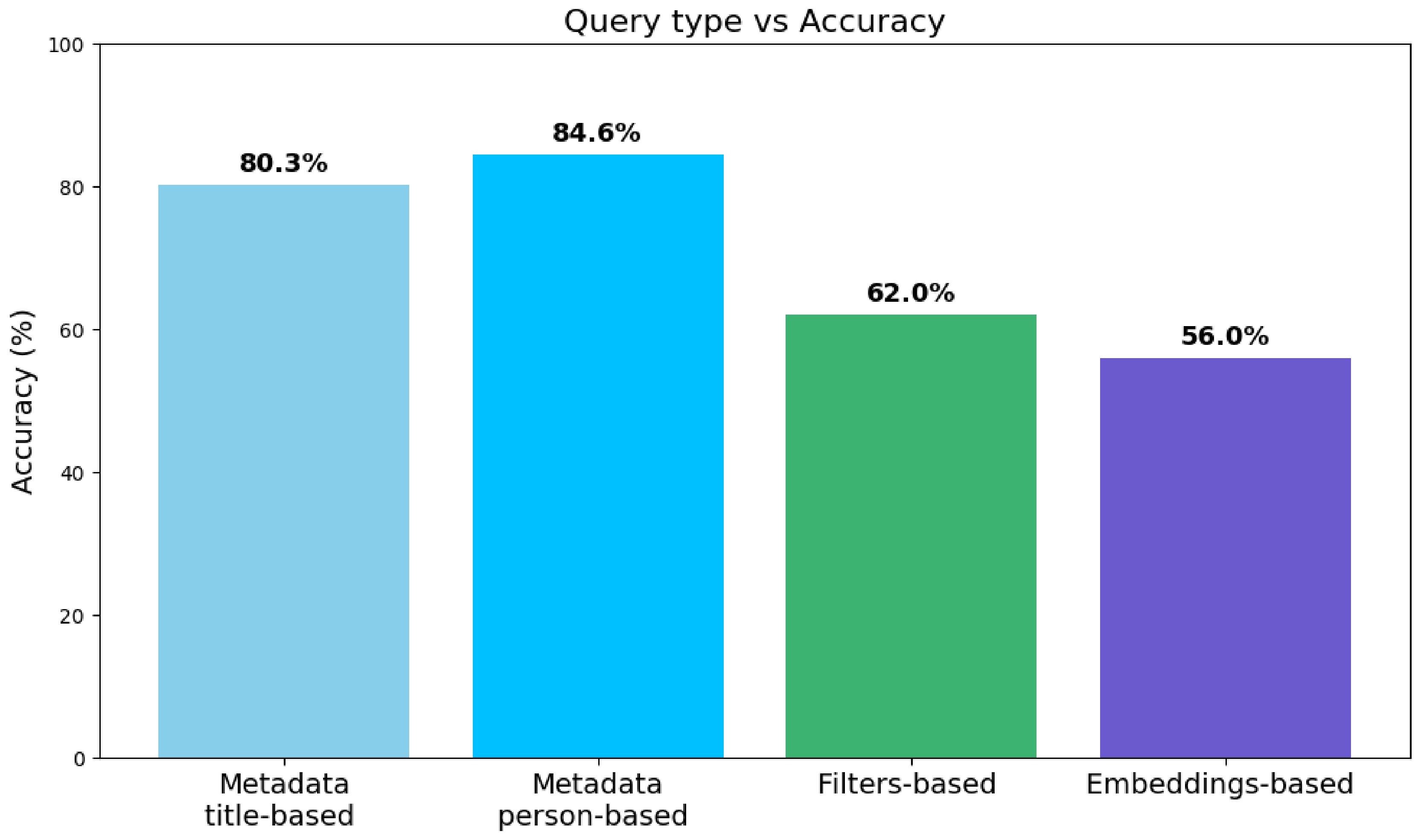


Figure 3. Overall accuracy

Overall, the system demonstrated satisfactory performance across all query types, although several challenges were identified. In title-based searches, partial matches caused errors, particularly with longer or more complex movie titles where the system incorrectly interpreted incomplete overlaps as correct results. For filter-based and embedding-based queries, relevant documents were generally retrieved, however, hallucinations frequently occurred as the LLM attempted to generate answers based on the extracted information. The results show that integration of RAG improves the LLM's ability to retrieve external knowledge and generate responses based on it, allowing the model to provide more accurate, context-aware answers even for information beyond its training data.

Conclusions

- The utilization of a similar approach to Adaptive-RAG[2], where the LLM classifies the query type based solely on structured instructions (without fine-tuning), and chooses an appropriate retrieval approach, yielded satisfactory results and shows promise for enhancing LLMs with access to external knowledge sources.
- Partial title matching was a challenge, leading to occasional misidentifications in metadata-based search.
- Filter and embedding based queries generally retrieved the correct documents, however, the increased retrieved contextual information induced hallucinations in the LLM's generated responses.
- A critic model[1], instructed to validate numerical facts and filter alignment (e.g., "88 min is greater than 1 hour"), helped improve factual correctness. However, due to LLM response variability, qualitative assessment of the critic's performance remains challenging.
- Manual evaluation on selected data subsets introduces a degree of subjectivity. Broader testing and incorporation of user feedback loops would provide a more objective and reliable measure of RAG performance.

References

[1] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi.
Self-rag: Learning to retrieve, generate, and critique through self-reflection, 2023.

[2] Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C. Park.
Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity, 2024.