Hallucination Mitigation Strategies in Medical Question-Answering (MQA)

Columbia University & Elsevier

Team: Rithika Lakshmi Shankar (rls2250), Prajna Girish (pg2785), Tushar Bura (tb3077), Kanisha Shah (ks4175)

Mentors: Dr. Elia Lima-Walton, Harsh Sindhwa

Abstract

Hallucinations in Medical Question Answering (MQA) systems—where large language models (LLMs) generate inaccurate or fabricated responses—pose significant risks to patient safety and healthcare decision-making. This project, Hallucination Mitigation Strategies in Medical Question Answering (MQA), addresses this critical challenge by developing a robust framework to detect and minimize hallucinations in LLM-based medical QA systems. Our approach integrates a multi-stage architecture that combines Retrieval-Augmented Generation (RAG) with advanced hallucination detection and answer refinement techniques. The system retrieves relevant medical knowledge using Elasticsearch and FAISS, embeds context through Dense Passage Retrieval (DPR), and generates responses using LLMs like Claude. To ensure accuracy, a cosine similarity-based validation module cross-checks generated answers against trusted medical knowledge bases, such as MedQuAD. This architecture ensures that answers are not only accurate but also grounded in reliable medical information. The project utilized the MedQuAD dataset, a comprehensive resource of 47,457 medical question-answer pairs sourced from 12 NIH websites. This dataset provided the foundation for training and evaluating our system, offering rich annotations such as question types, UMLS Concept Unique Identifiers (CUIs), and semantic types. Key findings demonstrate that our framework effectively reduces hallucination rates by grounding responses in verified knowledge while maintaining relevance and accuracy. Future improvements include fine-tuning LLMs with domain-specific datasets and enhancing hallucination detection through additional similarity metrics and external validation against resources like PubMed. This work contributes to the development of trustworthy AI systems for healthcare, ensuring safer and more reliable medical information retrieval for patients and practitioners alike.

1 Introduction

The rapid advancement of large language models (LLMs) has significantly transformed the landscape of artificial intelligence, particularly in specialized domains such as healthcare. Medical Question Answering (MQA) systems, powered by LLMs, have emerged as valuable tools for retrieving medical information and assisting in clinical decision-making. However, these systems face a critical challenge: the phenomenon of hallucination. Hallucination refers to the generation of inaccurate or fabricated responses that are not grounded in reliable medical knowledge. In the context of healthcare, such errors can have severe implications, including compromised patient safety and misinformed medical decisions. This project, Hallucination Mitigation Strategies in Medical Question-Answering (MQA), addresses these challenges by developing robust methodologies to detect and minimize hallucinations in MQA systems. Leveraging state-of-the-art techniques such as Retrieval-Augmented Generation (RAG) and advanced similarity-based validation mechanisms, this work aims to enhance the reliability and trustworthiness of AI-assisted medical information retrieval. By combining retrieval-based knowledge grounding with sophisticated answer verification strategies, the project seeks to ensure that LLM outputs are both accurate and contextually relevant. This report outlines the problem scope, technical approach, findings, and future directions for improving hallucination mitigation in MQA systems.

2 Problem Definition

Medical Question-Answering (MQA) systems are designed to provide accurate and contextually relevant answers to user queries based on vast repositories of medical knowledge. However, despite their potential, these systems often generate hallucinations—responses that are factually incorrect, irrelevant, or fabricated. This issue arises due to inherent limitations in LLMs, such as over-reliance on statistical patterns rather than factual grounding and insufficient domain-specific training.

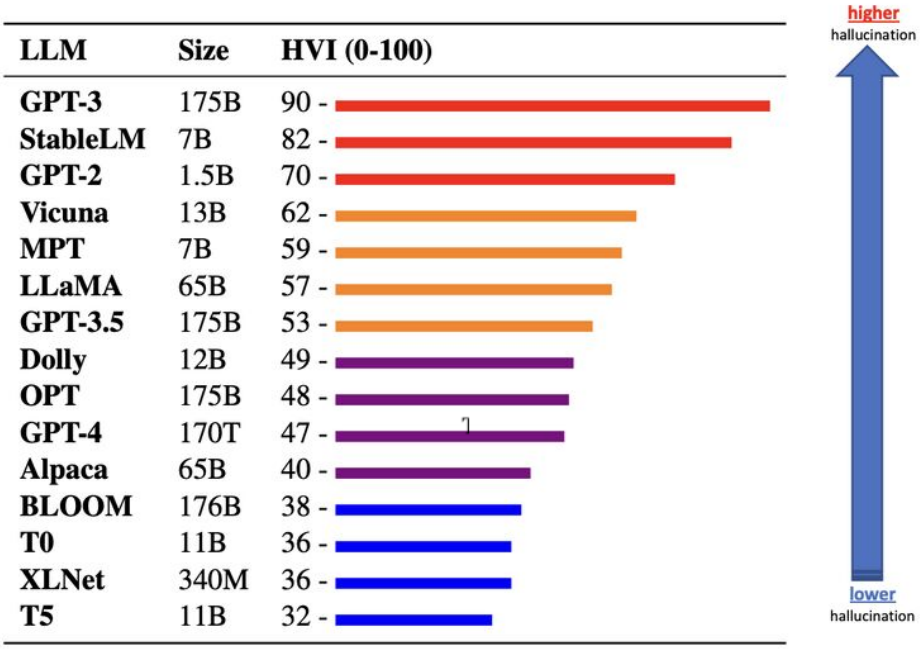


Figure 1: The HVI scale illustrates the hallucination tendencies exhibited by various LLMs.

In the medical domain, where precision is paramount, hallucinations pose significant risks:

- **Patient Safety:** Incorrect or misleading information can lead to inappropriate treatments or delayed medical interventions.
- **Healthcare Decision-Making:** Hallucinated outputs can erode trust in AI systems among healthcare providers and patients.
- **Ethical Concerns:** Dissemination of false information undermines the ethical responsibility of ensuring reliable and trustworthy AI applications.

The complexity of this problem is further compounded by the diversity and specificity of medical queries. Existing Medical Question Answering (MQA) systems struggle to balance generalizability with domain-specific accuracy. Additionally, traditional validation methods often fail to detect subtle inaccuracies in generated responses.

To address these challenges, this project proposes a multi-stage framework that integrates retrieval-based knowledge grounding with advanced hallucination detection and answer refinement techniques. By employing a Retrieval-Augmented Generation (RAG) architecture combined with similarity-based validation metrics, the system ensures that generated answers are anchored in verified medical knowledge while minimizing the risk of hallucination. This approach not only enhances answer accuracy but also builds trust in AI-driven medical applications.

3 Existing Frameworks and Their Drawbacks

Retrieval-Augmented Generation (RAG) Frameworks

RAG-based systems combine retrieval mechanisms with generative models to ground responses in external knowledge bases. These frameworks retrieve relevant context from sources like PubMed or MedQuAD and use it to generate answers. While RAG has shown promise in reducing hallucinations, it has notable limitations:

- **Dependency on Retrieval Quality:** Retrieval quality is highly dependent on the robustness of the underlying search mechanism (e.g., Elasticsearch or FAISS). Inadequate retrieval can lead to incomplete or irrelevant context being provided to the model, increasing the risk of hallucination.
- **Lack of Rigorous Validation:** These systems often fail to validate the generated response against the retrieved context rigorously, leading to subtle inaccuracies in answers.
- **Limited Domain Adaptation:** Adaptation to domain-specific nuances, such as medical terminology and guidelines, remains limited without extensive fine-tuning.

HALO Framework

HALO enhances medical QA systems by generating multiple query variations and retrieving enriched context from external knowledge bases using techniques like maximum marginal relevance scoring. It utilizes advanced prompt engineering strategies, such as few-shot prompting and chain-of-thought reasoning.

- **Computational Overhead:** The reliance on multiple query variations and iterative refinement increases computational overhead, making real-time applications challenging.
- **Latency Issues:** HALO's dependency on external knowledge bases introduces potential latency issues and risks inaccuracies if the knowledge base is outdated or incomplete.
- **Specialized Query Challenges:** While effective for general medical queries, HALO struggles with highly specialized or rare medical topics due to limited training data diversity.

Self-Reflection Loop Methodology

This framework incorporates iterative feedback loops where LLMs evaluate their own outputs for factual consistency and refine them over multiple iterations. It leverages multitasking capabilities and interactivity to improve response accuracy progressively. However:

- **Increased Response Time:** The iterative nature of self-reflection increases response time significantly, making it unsuitable for time-sensitive medical applications.
- **Reliance on Self-Assessment:** The approach relies heavily on the model's inherent ability to self-assess, which may not always align with external factual standards, leading to persistent inaccuracies.
- **Lack of Grounding:** It lacks robust mechanisms for grounding responses in verified medical knowledge bases, which limits its effectiveness in mitigating hallucinations specific to high-stakes medical contexts.

4 How Our Framework Overcomes These Drawbacks - An Overall Approach

Our proposed framework integrates a multi-stage architecture that combines the strengths of existing systems while addressing their limitations:

Enhanced Retrieval Layer

We employ Elasticsearch and FAISS for similarity-based retrieval but enhance this layer using Dense Passage Retrieval (DPR) encoders tailored for medical datasets like MedQUAD. This ensures that retrieved context is both relevant and comprehensive, overcoming regular RAG's dependency on generic retrieval mechanisms.

Rigorous Validation Mechanisms

Our system incorporates a cosine similarity-based validation module that compares generated responses against retrieved knowledge. This step ensures alignment between the output and verified medical information, addressing the validation gaps in existing frameworks like RAG and HALO.

Efficient Answer Refinement

Instead of computationally expensive iterative feedback loops, we use a single-pass answer refinement process that integrates similarity metrics with external validation against trusted medical databases (e.g., PubMed). This reduces latency while maintaining accuracy, overcoming the inefficiencies of self-reflection methods.

Domain-Specific Adaptation

Our framework is fine-tuned on domain-specific datasets and incorporates structured prompts tailored for medical queries. This improves performance on specialized topics, addressing HALO's limitations with rare or complex medical scenarios.

By combining retrieval-augmented generation with robust validation and refinement techniques, our framework ensures accurate, reliable, and efficient hallucination mitigation tailored specifically for Medical Question-Answering (MQA) systems.

5 About the Data

For this project, we utilized the MedQuAD (Medical Question Answering Dataset) from GitHub, a comprehensive dataset specifically designed to support research in medical question-answering systems. MedQuAD consists of 47,457 medical question-answer pairs curated from 12 reputable NIH websites, including sources such as *cancer.gov*, *niddk.nih.gov*, GARD, and MedlinePlus Health Topics. The dataset spans 37 question types related to diseases, drugs, and other medical entities such as tests, covering a wide range of topics critical to healthcare.

Key Features of the Dataset

- **Diversity of Question Types:** The dataset includes questions on various medical topics such as treatment options, diagnosis methods, side effects, and more. This diversity enables the dataset to represent real-world medical queries comprehensively.
- **Annotations for NLP Tasks:** Each question-answer pair is enriched with additional metadata in XML format, such as:
  - Question type and focus
  - Category of the question focus (e.g., Disease, Drug, or Other)
- **Domain-Specific Focus:** While most collections in MedQuAD focus on diseases, certain subsets also include drug-related and other medical topics. This specialization ensures relevance to healthcare applications.

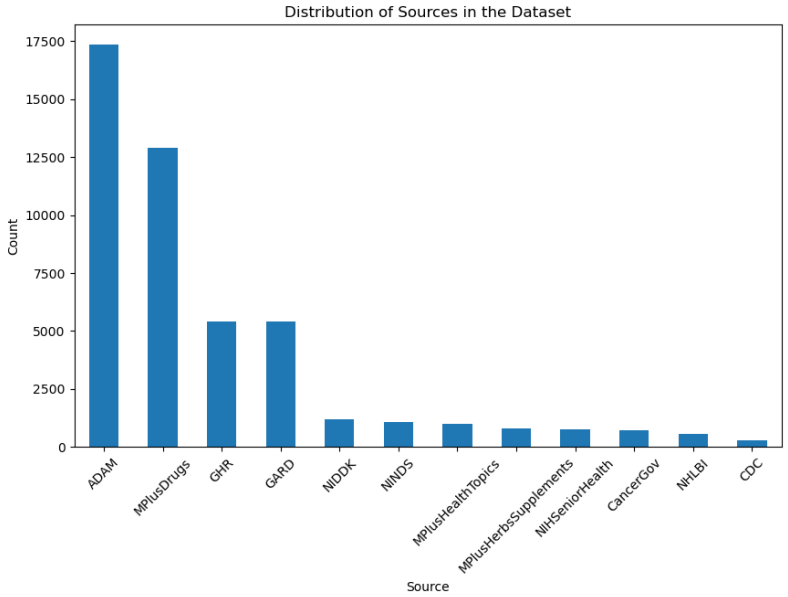


Figure 2: Distribution of Sources in the Dataset

The MedQuAD dataset was instrumental in training and evaluating our hallucination mitigation strategies. By combining the structured nature of MedQuAD with our multi-stage architecture (retrieval-augmented generation and validation), we ensured that our system could handle a wide range of medical queries while minimizing hallucinations effectively.

6 Analytic Methods

What is RAG?

Retrieval-Augmented Generation (RAG) is an approach that enhances the generative capabilities of Large Language Models (LLMs) by incorporating context retrieved from external knowledge sources. Instead of relying solely on pre-trained knowledge within the model, RAG combines retrieval-based and generative techniques to generate contextually relevant and factually accurate responses. This makes it especially suited for domains like healthcare, where accuracy is paramount.

How is Data Stored in ElasticSearch?

In our project, ElasticSearch serves as the backbone for efficient storage and retrieval of the MedQuAD dataset, enabling the RAG model to generate contextually accurate responses. The dataset is preprocessed and indexed into ElasticSearch, with each medical document represented as an individual document in the index. Key attributes such as question type, answers, contextual metadata, and embedding vectors (generated using a pre-trained language model) are stored alongside the raw text.

The embedding vectors are crucial as they enable similarity-based retrieval during querying. ElasticSearch's support for vector search with plugins like k-NN allows us to store these high-dimensional embeddings for real-time similarity computations. Additionally, fields are structured to optimize search, such as tokenizing and normalizing medical terms to ensure consistent retrieval performance.

By indexing the dataset this way, ElasticSearch acts as a scalable, fault-tolerant, and highly performant data store tailored for high-accuracy information retrieval, which is fundamental to our project's architecture.

Query Processing in ElasticSearch

When a user poses a question to the system, the query processing in ElasticSearch occurs as follows:

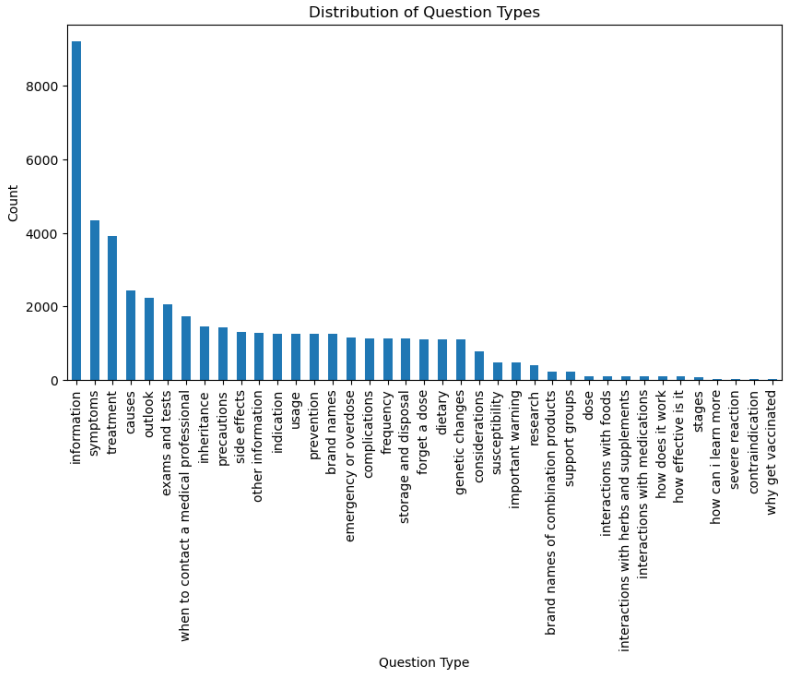


Figure 3: Distribution of Question Types in the Dataset

1. **Query Transformation:** The question is first embedded using a pre-trained language model into a high-dimensional vector. This embedding captures the semantic meaning of the question, making it suitable for similarity-based retrieval.

2. **Similarity Search:** ElasticSearch's vector search capabilities, using the stored embedding vectors, are employed to retrieve the most contextually relevant documents. The similarity computation (e.g., cosine similarity or dot product) ensures that retrieved results are semantically aligned with the query, even if exact keywords do not match.

3. **Ranking and Filtering:** The retrieved documents are ranked based on their relevance scores, calculated during similarity search. Additional filtering criteria, such as document metadata (e.g., specific medical terms or categories), can be applied to further refine the results.

4. **Context Extraction for Model Input:** The top-ranked document(s) are extracted and passed as context to the RAG model (Claude 3.5). This context serves as the foundation for generating the final answer, ensuring the response is not only contextually relevant but also factually grounded in the indexed dataset.

Why Elasticsearch Fits This Application?

- **Scalable Retrieval:** Elasticsearch can handle the large volume of medical data in the MedQuAD dataset, ensuring consistent performance even as the dataset grows.

- **Low-Latency Search:** Real-time response is critical for interactive systems like ours. ElasticSearch's optimized indexing and retrieval mechanisms allow for sub-second query responses.

- **Vector-Based Similarity:** The ability to perform efficient vector similarity searches directly in ElasticSearch aligns perfectly with the semantic nature of medical question-answering, bridging the gap between user queries and relevant dataset content.

- **Customizability:** With its rich querying capabilities, we can fine-tune search behavior (e.g., boosting specific fields or applying domain-specific filters) to optimize results for our specific use case.

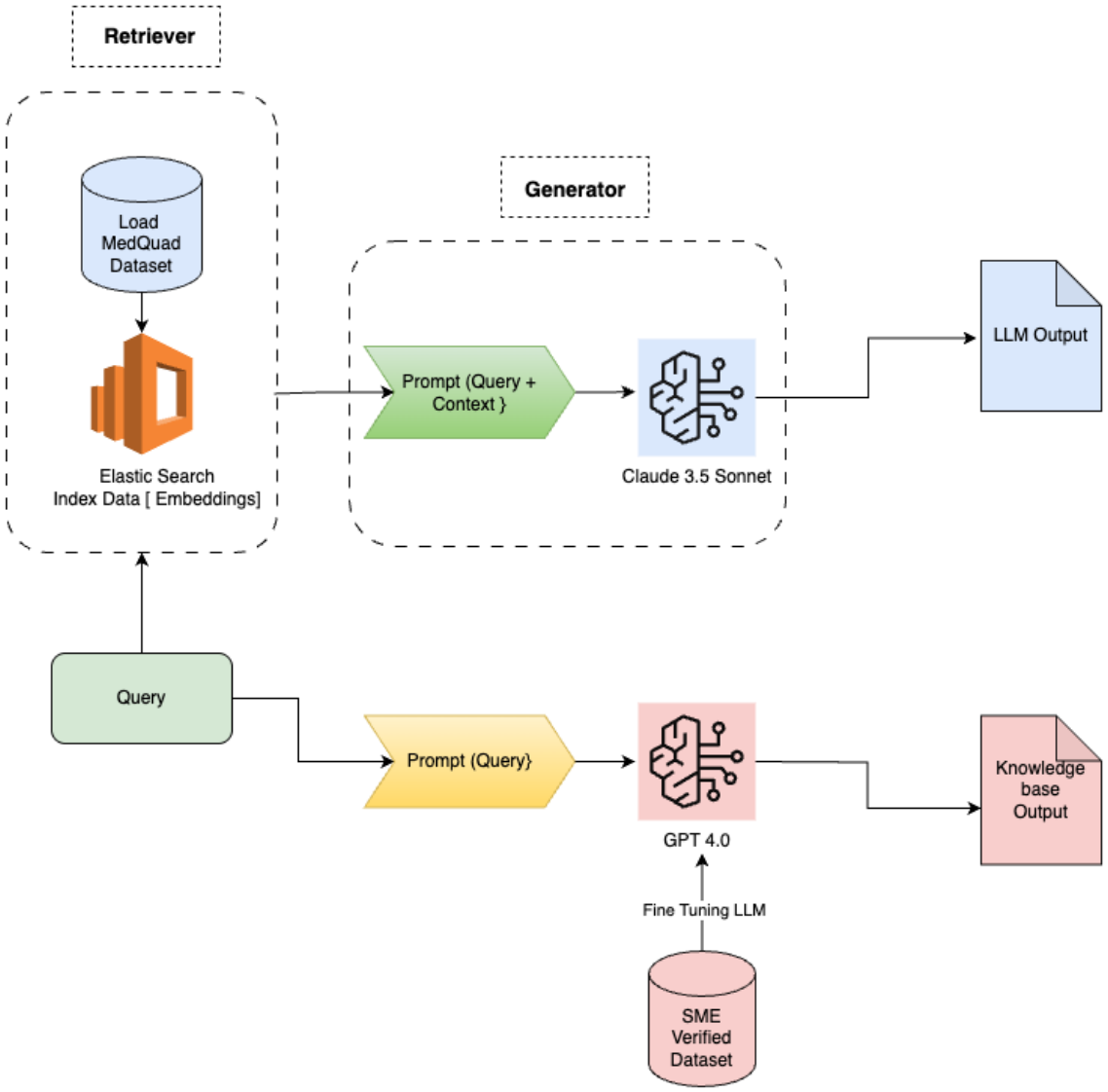


Figure 4: System Architecture Diagram

Why Fine-Tune the LLM on the Dataset Instead of Directly Using It?

- **Domain-Specific Knowledge:** General-purpose LLMs lack the domain-specific context required for accurate medical question-answering. Fine-tuning on the MedQuAD dataset ensures that the model is better aligned with the nuances of medical terminology and queries.
- **Improved Retrieval Relevance:** By fine-tuning, the model leverages retrieval contexts more effectively, producing more relevant and factually accurate outputs.
- **Reduction in Hallucination:** Fine-tuning aligns the generative capabilities of Claude 3.5 with real-world medical datasets, reducing the likelihood of generating hallucinated or irrelevant responses.

Why Use a Different LLM than Claude 3.5 for the Hallucination Pipeline?

- Using a different model ensures diverse generative pathways, making it less likely that both models exhibit the same hallucinations.
- GPT 4.0 was fine-tuned with an SME-verified dataset sourced from official medical websites, ensuring high-quality, accurate answers for validation.

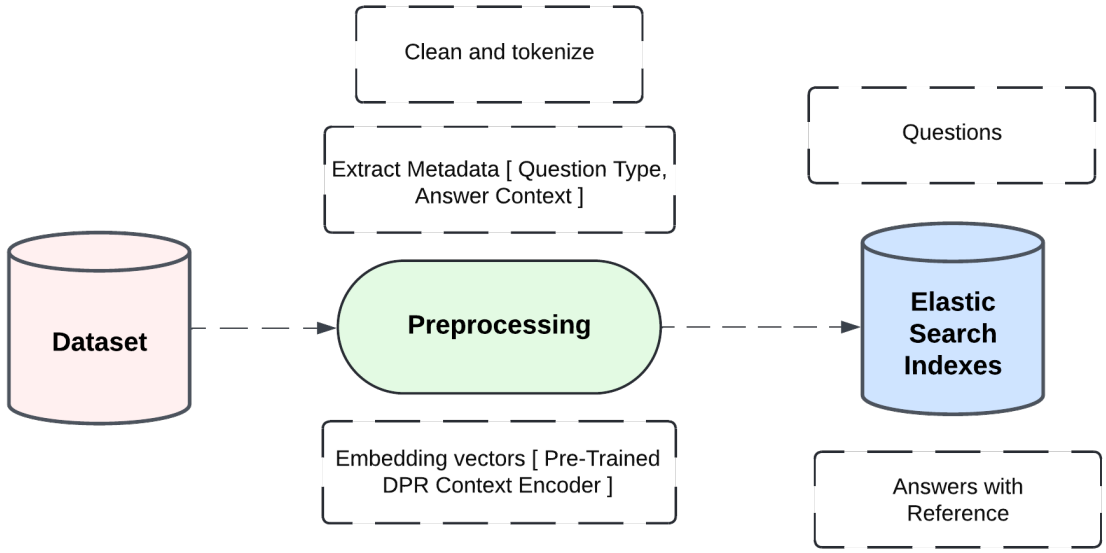


Figure 5: Data Storage in Elastic Search

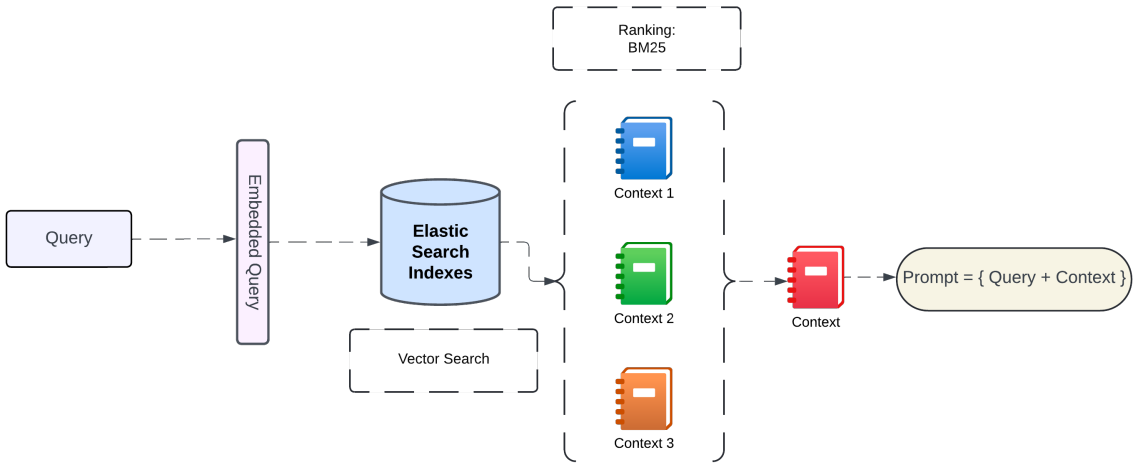


Figure 6: Query Process in Elastic Search

- While Claude 3.5 excels in context-based generation using RAG, GPT 4.0 provides a purely generative approach aligned with verified data, acting as a control for the output comparison process.
- Incorporating multiple models with different architectures increases the system's robustness and ensures a more comprehensive output evaluation.

Hallucination Detection Pipeline

To address hallucination, we employed a dual-model validation approach. The first model (RAG with Claude 3.5) generates answers by combining retrieval-based context with generative capabilities, while the second model (GPT 4.0) produces responses based on SME-verified datasets.

Each generated response is validated by calculating the cosine similarity of the answer embeddings. A threshold of 0.7 was chosen based on domain expert recommendations, ensuring a balance between accuracy and false-positive detection. Responses with similarity scores below this threshold are flagged as hallucinated, prompting further refinement of inputs or retrieval mechanisms.

Question Routing

Process:

1. **Input Handling:** Each user question is first routed to Model 1 (RAG with Claude 3.5), which retrieves relevant context from ElasticSearch and generates a response.
2. **Secondary Validation:** The same question is then passed to Model 2 (GPT 4.0), which generates an independent response based solely on the SME-verified dataset.

Why This Architecture?

- **Two-Layer Validation:** By generating outputs from two models with distinct architectures and datasets, the system minimizes the risk of shared errors or biases.
- **Reduced Dependency:** Relying on multiple models ensures that no single model disproportionately influences the output, making the system more reliable.
- **Iterative Improvement:** The dual-output approach allows for iterative refinement in cases where hallucinations are detected, providing a fallback mechanism for uncertain responses.

Answer Comparison - How It Works:

- The embeddings of both responses are generated and compared using cosine similarity.
- The similarity score quantifies how contextually aligned the two outputs are:
  - **Score** ≥ 0.7**:** Outputs are considered similar and accurate.
  - **Score** < 0.7**:** Responses are flagged as hallucinated, triggering re-evaluation with refined prompts or additional retrieval context.

Why Cosine Similarity?

- Provides a mathematically sound way to measure semantic alignment between the two answers.
- Computationally efficient and integrates seamlessly with existing NLP pipelines.
- The 0.7 threshold was empirically determined to balance sensitivity and specificity in hallucination detection.

Decision Logic

Process Flow:

- **High Similarity** (≥ 0.7)**:** The response is deemed reliable and returned as the final output to the user.
- **Low Similarity** (< 0.7)**:** The output is flagged for hallucination. A retry loop is initiated, refining the input query or retrieving additional context from ElasticSearch to improve the accuracy of Model 1's output.

Why This Logic?

- **Error Mitigation:** Ensures that hallucinated responses are identified and corrected before being presented to the user.
- **Iterative Refinement:** Provides a systematic way to handle ambiguous or low-confidence queries.
- **End-User Trust:** Increases the reliability of the system, fostering trust in its outputs, particularly in high-stakes medical applications.

This approach balances innovation and reliability by leveraging state-of-the-art LLMs with robust retrieval and validation techniques. It ensures the accuracy and dependability of medical question-answering systems, mitigating hallucination risks effectively.

Large Language Model Integration

In this project, we integrated Amazon Bedrock's Claude-3.5 Sonnet model from Anthropic as the Large Language Model (LLM) for generating accurate, contextual answers to user queries. The LLM is part of the broader Retrieval-Augmented Generation (RAG) pipeline, where retrieved context from relevant documents is combined with generative AI capabilities to provide trustworthy and domain-specific answers.

Architecture and Implementation

Model and Framework

- The Claude-3.5 Sonnet model was accessed using Amazon Bedrock Runtime API.
- Bedrock provides a managed interface to deploy and invoke foundation models without requiring specialized infrastructure, making it scalable and efficient for this project.
- The model version used was `bedrock-2023-05-31`.

Approach

The following steps were implemented to utilize the LLM effectively:

1. **Query Processing:** The user query is taken as input, such as *"What are the genetic changes related to ovarian cancer?"*.
2. **Context Retrieval:** Relevant documents from the MedQUAD dataset are retrieved using Dense Passage Retrieval (DPR) based on semantic similarity. The top-k most relevant documents form the context.
3. **In-Context Learning:**
   - Example question-answer pairs are added as few-shot examples to guide the model in producing accurate answers.
   - The user query and retrieved context are appended as a structured prompt for the LLM.
4. **API Request:** The structured input is sent as a JSON body to the Claude model using the Messages API, adhering to the following parameters:
   - **Input Format:**
     - Few-shot examples (`"role": "user"` and `"role": "assistant"`) are first provided.
     - Context and the main query are then appended.
   - **Key Parameters:**
     - `max_tokens`: Set to 200 to control response length.
     - `anthropic_version`: Defined as `bedrock-2023-05-31`.
5. **Response Parsing:** The model response is returned in a JSON format, from which relevant textual output is extracted.

Key Code Components

- **Context and Few-Shot Example Preparation:** Example Q&A pairs are included in the prompt as guidance for the LLM.

```
medquad_examples = [
    {"question": "What is pneumonia?", "answer": "Pneumonia is an infection that inflames air sacs in one or both lungs."},
    {"question": "How is pneumonia treated?", "answer": "Pneumonia is usually treated with antibiotics and rest."}
]
```

- **Prompt Construction:** The retrieved context and query are added in a conversational structure.

```
messages = example_messages + [
    {"role": "user", "content": f"Based on the following context, if relevant, answer the question accurately. Otherwise, provide a general answer with source of citation:\n\n{context_docs}"},
    {"role": "user", "content": query}
]
```

- **Amazon Bedrock API Invocation:** The Claude-3.5 Sonnet model is invoked via the Bedrock Runtime client using the constructed prompt.

```
response = client.invoke_model(
    modelId="arn:aws:bedrock:us-east-1::foundation-model/anthropic.claude-3-5-sonnet-20240620-v1:0",
    body=body,
    contentType="application/json",
    accept="application/json"
)
```

- **Response Parsing:** The LLM's response is extracted and displayed.

```
content_list = response_json.get('content', [])
assistant_content = ""

for item in content_list:
    if item.get('type') == 'text':
        assistant_content += item.get('text', '') + "\n\n"
```

Why In-Context Learning?

- In-context learning enables the LLM to leverage examples provided dynamically in the prompt, improving its ability to understand the domain-specific task.
- Few-shot examples provide the model with a demonstration of the expected input-output format.
- Adding retrieved context ensures that the LLM generates fact-based responses grounded in domain-specific documents.

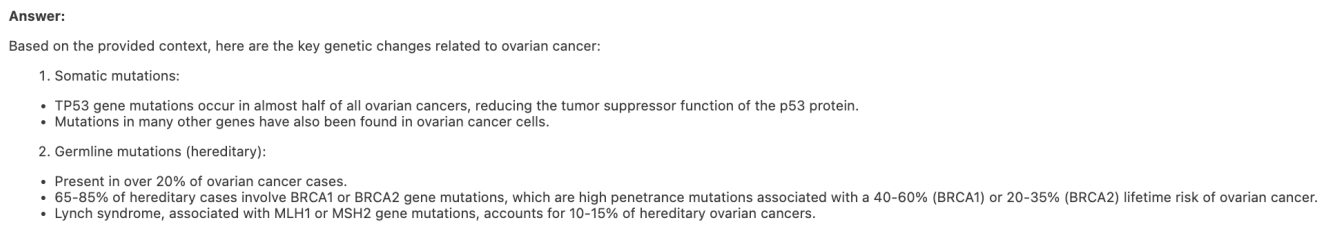
Results and Observations

The results present the responses generated by the Retrieval-Augmented Generation (RAG) pipeline and the integrated Claude-3.5 Sonnet model deployed using Amazon Bedrock Runtime. These responses demonstrate the system's ability to combine retrieved context with a generative model to deliver accurate, context-aware answers.

- Query and Response Example

User Query: *"What are the genetic changes related to ovarian cancer?"*
System Response: The response was generated after retrieving relevant documents using Dense Passage Retrieval (DPR) and passing the contextual data into the Claude-3.5 Sonnet model for answer

synthesis. Below is the structured response provided by the system:



Strengths and Limitations

Strengths

- The model generates clear and concise responses grounded in retrieved context.
- It handles structured queries effectively and provides domain-relevant answers.

Limitations

- For vague or insufficient retrieved context, the model may provide general or incomplete answers.
- Token limits may truncate responses for highly detailed or broad queries.

Conclusion

The integration of Amazon Bedrock's Claude-3.5 Sonnet via in-context learning, combined with a robust retrieval system, demonstrates the effectiveness of Retrieval-Augmented Generation for generating accurate answers. By leveraging few-shot examples and dynamically retrieved context, the system ensures responses are grounded, trustworthy, and domain-relevant.

Hallucination Detection Pipeline

What is Hallucination Detection and Why is it Needed?

Hallucination in Large Language Models (LLMs) occurs when the model generates outputs that are factually incorrect, inconsistent, or not grounded in the provided context. This is a significant challenge, particularly in sensitive domains like healthcare, where even slightly inaccurate information can have significantly harmful consequences. Hallucination detection aims to identify when an LLM generates such unreliable responses by comparing its outputs to a trusted knowledge base or context.

In this project, hallucination detection ensures that the LLM's responses are factually accurate and aligned with verified medical information. This is critical as LLMs like Claude (Anthropic) or GPT-4 are generative models that, without proper grounding, may fabricate plausible-sounding yet incorrect answers. Our solution uses a combination of retrieval-augmented generation (RAG) and cosine similarity-based validation to verify and ensure the accuracy of the outputs.

Data

To provide the LLM with reliable grounding, we created a comprehensive knowledge base by manually compiling information from trusted online medical sources. The primary sources include MedlinePlus, the National Institute of Health (NIH), and Mayo Clinic. These sources were selected for their credibility, breadth of medical topics, and public trust. The collected data covers various medical domains such as oncology, cardiology, and general health sciences. Text from these sources was cleaned and structured into smaller, contextually coherent chunks to facilitate efficient retrieval.

The raw text underwent cleaning to improve readability and alignment with LLM requirements. The key steps to clean the data included:

- Eliminated unnecessary artifacts like page numbers, special characters, excessive whitespace, and HTML remnants.
- Dropped irrelevant sections, such as copyright notices, bibliographic data, and website footers. This was done by setting heuristic rules, such as skipping sections containing keywords like "©" or "References."
- Used LanguageTool Python API to automatically correct minor spelling and grammar issues in the raw text.
- Standardized measurements (e.g., "mg" to "milligrams") and medical terminologies using pre-built synonym dictionaries (e.g., SNOMED CT).

This knowledge base that we have constructed can be used in conjunction with manual validation by Subject Matter Experts (SMEs), which is resource-intensive and not scalable. It serves as the foundation for the retrieval and grounding of LLM responses.

Working of the Hallucination Model

The compiled knowledge base is preprocessed and indexed into Elasticsearch. The preprocessing step involves breaking down the data into manageable, contextually coherent chunks to enable granular retrieval. Each chunk is stored with metadata, such as document identifiers and topic tags, to facilitate targeted querying.

Simultaneously, semantic embeddings of these chunks are generated using a pre-trained model, Sentence Transformers. These embeddings are stored in Elasticsearch using vector fields, enabling similarity-based search. By embedding the data, the system supports queries that extend beyond simple keyword matches, capturing nuanced relationships between the input query and stored data.

When a query is received:

1. **Elasticsearch Keyword Search:** Elasticsearch performs a traditional keyword-based search to retrieve the top-most relevant chunks of text. This serves as a preliminary filter to narrow down the dataset.
2. **Embedding-Based Refinement:** The retrieved chunks are compared with the query using cosine similarity between their embeddings and the query embedding. This ensures that the most semantically relevant chunks are selected, even if the exact phrasing in the query differs from the stored data.
3. **Context Integration for LLM:** The top-ranked chunks are concatenated and passed to the OpenAI GPT-4-32k model as part of the prompt, along with the user's query. The model is instructed to strictly limit its output to the retrieved context, ensuring that responses remain grounded in the indexed knowledge base.

7 Evaluation

To accomplish the task of data evaluation, cosine similarity metrics between the generated response and relevant sections of the document could be incorporated. For instance, a similarity score above a defined threshold (Here we use 0.7) would indicate strong alignment between the output and the source context, while lower scores could flag potential hallucinations. This methodology ensures objective evaluation without manual intervention. The lack of explicit retrieval mechanisms in the current implementation limits the scope of quantitative evaluation metrics. However, manual validation by SMEs or domain experts could be used to ensure factual correctness of responses.

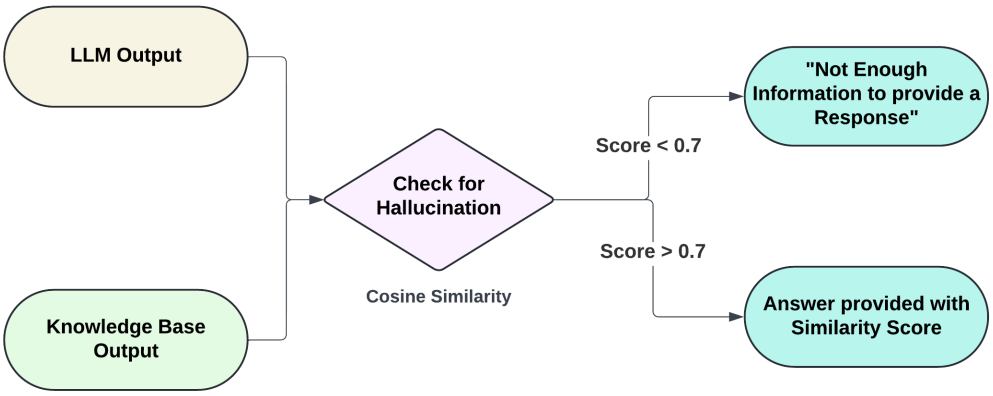


Figure 7: Halluncination Detection Model Workflow

8 Results

High Similarity Example

Question:

What are the risk factors of breast cancer?

LLM Output:

1. *Being female*
2. *Increasing age*
3. *Family history of breast cancer*
4. *Personal history of breast cancer*
5. *Inherited genes (BRCA1 and BRCA2)*
6. *Radiation exposure*
7. *Obesity*
8. *Beginning menstruation at a younger age*
9. *Starting menopause at an older age*
10. *Having first child at an older age*
11. *Postmenopausal hormone therapy*
12. *Drinking alcohol*

Output from Knowledge Base:

1. *Gender: Being female is the most significant risk factor.*
2. *Age: Risk increases with age, particularly after 50.*
3. *Genetic Mutations: Inherited mutations in genes like BRCA1 and BRCA2 elevate risk.*
4. *Family History: Having close relatives with breast cancer increases risk.*
5. *Reproductive History: Early menstruation (before 12) or late menopause (after 55) exposes breasts to hormones longer, raising risk.*
6. *Dense Breast Tissue: Makes detection harder and is linked to higher risk.*
7. *Lifestyle Factors: Obesity, alcohol consumption, and lack of physical activity contribute to increased risk.*

This produced a high similarity score of 0.86, indicating strong alignment between the generated response and the retrieved context.

Analysis:

The high similarity score demonstrates that the LLM not only covered all major points but also structured its response coherently, reflecting the critical information from the knowledge base. Minor differences

in phrasing did not impact the overall alignment. This example confirms the reliability of the model in grounded contexts.

Moderate Similarity Example

Question: Tell me what you know about carcinomas

LLM Output:

Carcinomas are a type of cancer that starts in cells that make up the skin or the tissue lining organs, such as the liver or kidneys.

1. *Adenocarcinoma: Starts in glandular cells that secrete fluids.*
2. *Basal cell carcinoma: Begins in the lower part of the epidermis (outer layer of skin).*
3. *Squamous cell carcinoma: Starts in squamous cells, which are flat cells that line many organs.*
4. *Transitional cell carcinoma: Begins in transitional epithelium, found in the linings of certain organs.*

Output from Knowledge Base:

Carcinomas are a type of cancer that originate in epithelial tissues such as skin, breast, colon, and lungs. They are categorized under solid tumors. There are subtypes of carcinomas including adenocarcinomas, which develop in glandular tissues, and squamous cell carcinomas, which arise from squamous epithelium, affecting areas like the skin, lungs, and esophagus.

Here, the LLM achieved a similarity score of 0.74, reflecting moderate alignment between its generated response and the knowledge base.

Analysis:

The LLM's response was detailed and generally aligned with the knowledge base. However, slight discrepancies in focus (e.g., specific organs emphasized) and additional phrasing differences resulted in a slightly lower similarity score. While the model's response was factually accurate, this score highlights the impact of subtle contextual variations on similarity metrics.

Low Similarity Example: Stroke Symptoms vs. Risk Factors

Question:

What causes an ischemic stroke?

LLM Output:
Smoking and obesity are significant contributors to the risk of ischemic stroke.

Output from Knowledge Base:
Stroke symptoms include sudden weakness, slurred speech, and loss of balance.

The outputs have a very low similarity, with a score of 0.4, demonstrating significant divergence between the LLM output and the context.

Analysis:

These sentences focus on entirely different aspects of strokes—risk factors versus symptoms—resulting in minimal semantic overlap. The low similarity score highlights the importance of grounding the LLM's response in the relevant context, as failure to do so risks hallucination or misalignment with the query.

Insights from Results

The results demonstrate a clear correlation between semantic similarity and response alignment:

- **High Similarity (≥ 0.8):** The LLM closely aligns with the knowledge base, ensuring accurate and reliable answers.
- **Moderate Similarity ($0.7 - 0.8$):** The LLM captures the main idea but introduces minor variations that reduce alignment.
- **Low Similarity (< 0.7):** Misalignment occurs when the context and response diverge, even if both are factually correct within their domains.

The method effectively identifies cases where the model produces accurate, grounded responses and flags potential hallucinations in scenarios with lower alignment. These insights underscore the utility of this approach in ensuring the reliability of LLM outputs in high-stakes domains like healthcare.

9 Discussion

Summary

This project addresses the critical issue of hallucinations in Medical Question Answering (MQA) systems by developing a multi-stage framework that integrates retrieval-augmented generation (RAG), hallucination detection, and answer refinement. By leveraging the MedQuAD dataset, the system retrieves relevant medical knowledge, generates responses using LLMs, and validates outputs against trusted sources using similarity-based metrics.

The findings demonstrate that this approach effectively minimizes hallucination while ensuring that generated responses are accurate and grounded in reliable medical knowledge. Our framework provides a promising solution for improving the trustworthiness of AI-driven healthcare applications.

Conclusion

- Hallucination in MQA systems poses significant risks, but combining retrieval-based grounding with robust validation techniques can substantially mitigate this issue.
- The integration of domain-specific datasets like MedQuAD and similarity-based validation ensures accurate, contextually relevant responses in high-stakes medical applications.
- Reliable AI systems in healthcare require continuous refinement to balance accuracy, efficiency, and adaptability to diverse medical queries.

Future Work and Improvements

If given additional time, we would focus on:

- **Enhancing Training with Domain-Specific Datasets:** Expanding the generation layer by training on more diverse and specialized medical datasets to improve adaptability and precision for handling complex medical queries. Incorporating datasets that address niche medical fields to ensure coverage across a broader range of healthcare topics.
- **Integration of Multiple Similarity Metrics:** Utilizing additional similarity metrics such as Euclidean distance, Jaccard similarity, or BM25 alongside cosine similarity to refine answer validation. Performing external validation against trusted resources like PubMed, UMLS, or WHO guidelines to ensure outputs are consistent with established medical knowledge.
- **Leveraging MCQ-Based Datasets:** Extending our current architecture to train and validate on multiple-choice question (MCQ)-based datasets. Leveraging data analysis already performed on MCQ datasets (documented in our GitHub repository under the Future Work section) to improve model understanding and response generation for structured assessments.

- **Improved Hallucination Detection and Iteration:** Incorporating ensemble approaches by combining outputs from multiple LLMs for better hallucination mitigation. Exploring confidence scoring mechanisms to flag uncertain or low-confidence responses automatically.
- **Human-in-the-Loop Validation:** Introducing a human-in-the-loop component where medical professionals validate flagged responses, enhancing the model's accuracy over time through active learning.
- **Enhanced Multi-Language Support:** Training the system to support multi-language medical queries, making it accessible to non-English-speaking populations while maintaining precision.

Ethical Considerations

While leveraging datasets like MedQuAD, we ensured compliance with data privacy regulations and copyright restrictions by avoiding the use of sensitive or proprietary information. However, if deployed, the potential misuse of hallucinated outputs in critical healthcare decisions remains a concern, emphasizing the need for rigorous oversight and human-in-the-loop validation mechanisms.

10 Acknowledgments

We thank our mentors from Elsevier for their guidance and support throughout this project. We also express our gratitude to the Data Science Institute at Columbia University for providing the resources and academic environment that made this project possible.

References

1. Yunsoo Kim, Jinge Wu, Yusuf Abdulle, Honghan Wu. *MedExQA: Medical Question Answering Benchmark with Multiple Explanations.* Available on arXiv
2. Valentin Liévin, Christoffer Egeberg Hother, Andreas Geert Motzfeldt, Ole Winther. *Can large language models reason about medical questions?* Available on arXiv
3. Mobashir Sadat. *DelucionQA: Detecting Hallucinations in Domain-specific Question Answering.* Available on arXiv.
4. `https://github.com/KhalilMrini/Medical-Question-Answering`
5. `https://github.com/abachaa/MedQuAD`
6. *A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions.* Available on arXiv.
7. `https://github.com/langchain-ai/rag-from-scratch`
8. `https://github.com/facebookresearch/faiss`
9. `https://www.geeksforgeeks.org/cosine-similarity/`
10. `https://www.elastic.co/guide/en/elasticsearch/reference/current/docs-index_.html`
11. `https://aws.amazon.com/blogs/machine-learning/getting-started-with-amazon-bedrock`
12. Vipula Rawte, Swagata Chakraborty, Agnibh Pathak, Anubhav Sarkar. The Troubling Emergence of Hallucination in Large Language Models - An Extensive Definition, Quantification, and Prescriptive Remediations Available on Research Gate.

Appendix

Team Contribution

This project was a collaborative effort where each team member contributed to all aspects of the work. However, specific areas were led by individual team members to leverage their expertise and ensure efficient task division. **Tushar** focused on obtaining the initial data, identifying appropriate datasets, and conducting exploratory data analysis (EDA), along with data cleaning and preprocessing. **Kanisha** took the lead in designing and building the Retrieval-Augmented Generation (RAG) architecture, ensuring seamless integration of retrieval mechanisms with the generative LLM. **Rithika** concentrated on the integration and optimization of the Large Language Model (LLM), working on prompt engineering and model fine-tuning to enhance the generation layer. **Prajna** spearheaded the hallucination detection framework, developing the validation pipeline and implementing cosine similarity metrics to ensure reliable outputs. Our collaboration and coordination helped us collectively achieve the project's objectives.