Knowledge Graph Question Answering: Comparing RAG Approaches

By-
Aayush Singh (G26487644)
Akash Raghavendra (G41868923)
Vardh Jain (G24531843)

Abstract

Retrieval-Augmented Generation (RAG) is a standard technique for grounding Large Language Models (LLMs) in external data, yet conventional implementations often struggle with complex reasoning. This study investigates the hypothesis that integrating Knowledge Graphs into the RAG pipeline ("Graph RAG") yields superior performance compared to standard vector-based "Plain RAG" in the biomedical domain. We conducted a comparative analysis using the PubMedQA dataset to benchmark both architectures. Our evaluation demonstrates that Graph RAG outperforms the Plain RAG baseline, achieving a 12.1% increase in accuracy (from 58% to 65%) and a 52.8% improvement in recall for negative answers. These results confirm that while Plain RAG is susceptible to surface-level keyword matching, Graph RAG significantly improves the model's ability to discern causal negation and handle complex queries.

1. Introduction

1.1 Problem Context

While RAG systems mitigate LLM hallucinations by retrieving relevant context, "Plain RAG" implementations—which rely on semantic similarity search—face inherent limitations. These systems

often fail to capture explicit relationships between entities (e.g., hierarchical medical data) or perform multi-hop reasoning. Consequently, they frequently function as "black boxes," lacking transparency in their retrieval logic.

1.2 Study Objective

The primary objective of this project was to empirically evaluate whether a **Graph RAG** architecture offers a tangible improvement over a **Plain RAG** baseline in a specialized scientific domain. Specifically, this study aimed to determine if the structured relationships within a knowledge graph could resolve the "Yes-Man" bias often observed in Plain RAG, where models hallucinate positive correlations based on keyword overlap. We utilized the PubMedQA dataset to compare both systems on accuracy, recall, and computational efficiency.

2. Methodology

2.1 Dataset

The evaluation utilized the **PubMedQA** dataset, specifically the labeled subset (PQA-L) containing 1,000 Question-Answering pairs derived from biomedical literature.

- **Structure:** Questions are derived from paper titles (e.g., "Is 8 hours of sleep associated with better memory?"), and contexts are drawn from abstracts.
- **Classes:** The dataset requires classification into "yes," "no," or "maybe."
- **Distribution:** The data is imbalanced, with 55.2% "yes," 33.8% "no," and 11.0% "maybe" samples.

2.2 Baseline System: Plain RAG

To establish a performance baseline, we implemented a standard RAG pipeline.

- **Embedding:** Text chunks were vectorized using sentence-transformer
- **Retrieval:** We employed **FAISS** (Facebook AI Similarity Search) to perform Inner Product searches, retrieving the top 3 most similar documents.
- **Generation:** Context was fed into a quantized DeepSeek-R1-Distill-Llama-8B model for response generation.

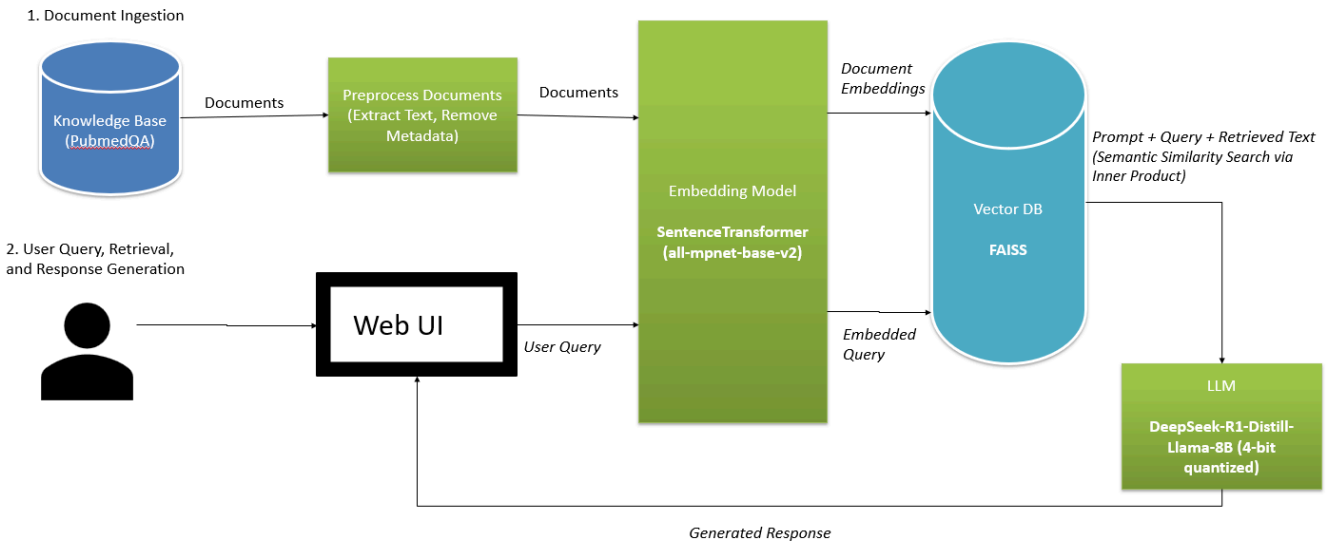


Figure 1: The Baseline Plain RAG Architecture implemented for this study.

2.3 Experimental System: Graph RAG

Our system implements a Graph Retrieval-Augmented Generation (GraphRAG) architecture designed to overcome the limitations of flat vector retrieval in complex biomedical reasoning. The pipeline integrates a structured Knowledge Graph with hybrid retrieval mechanisms to provide the Large Language Model (LLM) with holistic, document-level context rather than fragmented text snippets.

The architecture consists of three primary stages:

2.3.1. Knowledge Graph Construction & Ingestion

The foundation of the system is a graph database built on **ArangoDB**, which replaces the traditional flat vector store used in standard RAG applications. The data source is the **PubMedQA** dataset, specifically the labeled subset of expert-annotated biomedical research questions.

The graph schema is designed to model explicit relationships between data points:

- **Nodes:** The graph consists of *Paper* nodes (containing the full abstract and metadata), *Chunk* nodes (containing segmented text for vector search) and *concept* nodes (representing MeSH terms).
- **Edges:** Relationships are explicitly mapped via edges such as *HAS_CONTEXT*, which links a parent paper to its constituent text chunks, and *MENTIONS*, which links papers to relevant biomedical concepts.

During ingestion, abstracts are preprocessed to remove extraneous metadata and segmented into chunks. These chunks are embedded using the **SentenceTransformer** model and stored within the graph to facilitate semantic search.

2.3.2. Hybrid Retrieval Strategy

The core innovation of the system lies in its multi-stage retrieval process, which combines semantic search with graph traversal to ensure high precision and context completeness.

- **Stage A: Broad Semantic Search (Recall):** When a user submits a query, it is embedded into a vector using the same SentenceTransformer model used during ingestion. The system performs

an approximate nearest neighbor search against the *chunk* collection to retrieve the **Top-75 candidates**. A high recall number is chosen intentionally to cast a wide net and capture relevant information that may be buried in different sections of a paper (e.g., Methods vs. Conclusion).

- **Stage B: Cross-Encoder Re-Ranking (Precision):** To filter noise from the broad vector search, the system employs a **Cross-Encoder**. Unlike the bi-encoder used for the initial search, the Cross-Encoder processes the user query and the retrieved text chunk simultaneously to output a relevance score. This allows the system to re-rank the 75 candidates and select the **Top-3** most relevant chunks with high precision.
- **Stage C: Graph Traversal (Context Reconstruction):** Standard RAG systems often fail because they feed isolated chunks to the LLM, stripping away necessary context (e.g., a chunk might state "results were significant" without mentioning the control group). Our GraphRAG system addresses this by performing a **parent retrieval operation**. Once the Top-3 chunks are identified, the system traverses the *HAS_CONTEXT* edge in the graph to locate the parent *Paper* node. It then retrieves the **full abstract** associated with that paper.

2.3.4. Generative Reasoning

The final stage involves the synthesis of an answer. The system constructs a prompt that combines the user's original question with the fully reconstructed abstracts (Introduction, Methods, Results, and Conclusion) retrieved via the graph.

This comprehensive context is fed into a **DeepSeek-R1-Distill-Llama-8B** (4-bit quantized) Large Language Model. By providing the full study context rather than fragmented sentences, the LLM can perform "skeptical reasoning"—accurately discerning whether a study claims a positive correlation, a negative result, or inconclusive findings. This structure allows the model to generate a definitive "Yes."

"No" or "Maybe" classification with significantly reduced hallucinations compared to the baseline.

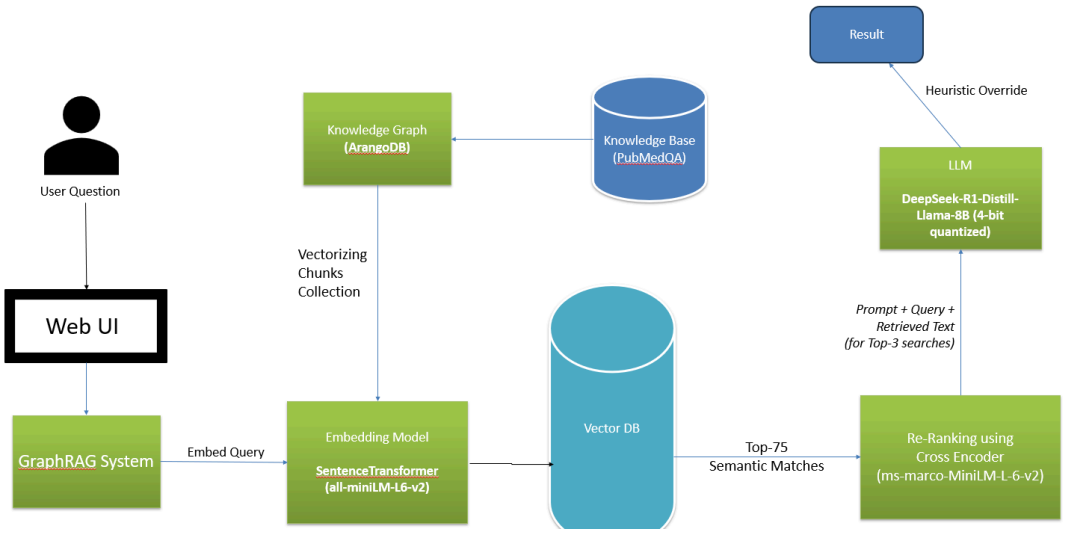


Figure 2: The Experimental Graph RAG Architecture used for comparison.

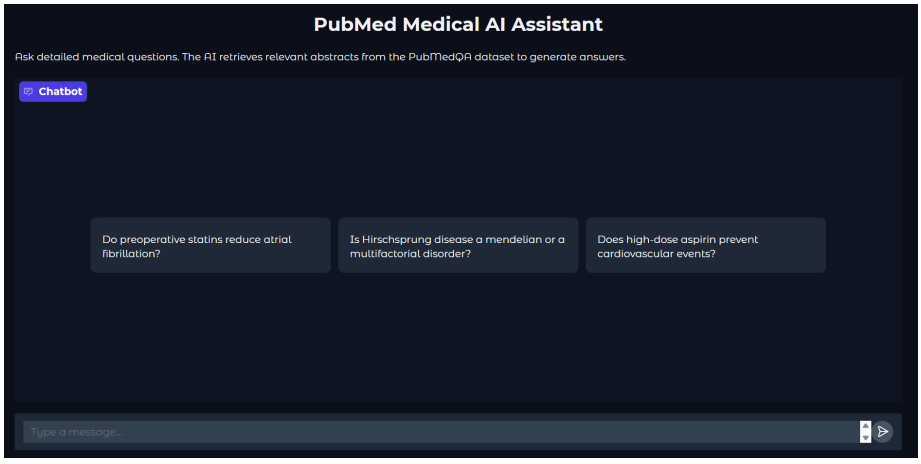


Figure 3: User Interface

3. Experimental Results

3.1 Baseline Performance (Plain RAG)

```
================================================
BENCHMARK RESULTS
================================================
Time taken: 4556.10s
Accuracy: 58.00%

--- Classification Report ---
              precision    recall  f1-score   support

         yes       0.65      0.82      0.73       112
          no       0.56      0.36      0.44        53
       maybe       0.20      0.14      0.17        35

    accuracy                           0.58       200
   macro avg       0.47      0.44      0.44       200
weighted avg       0.55      0.58      0.55       200
```

Figure 4: Evaluation Metrics for Plain RAG

The Plain RAG system achieved an accuracy of **58.00%**. The model exhibited a high false positive rate. Out of 53 actual "no" cases, it incorrectly predicted "yes" 25 times. It struggled significantly with ambiguous queries, misclassifying 24 out of 35 "maybe" cases as "yes." The baseline acted as a "Yes-Man," favoring affirmative answers due to keyword presence.

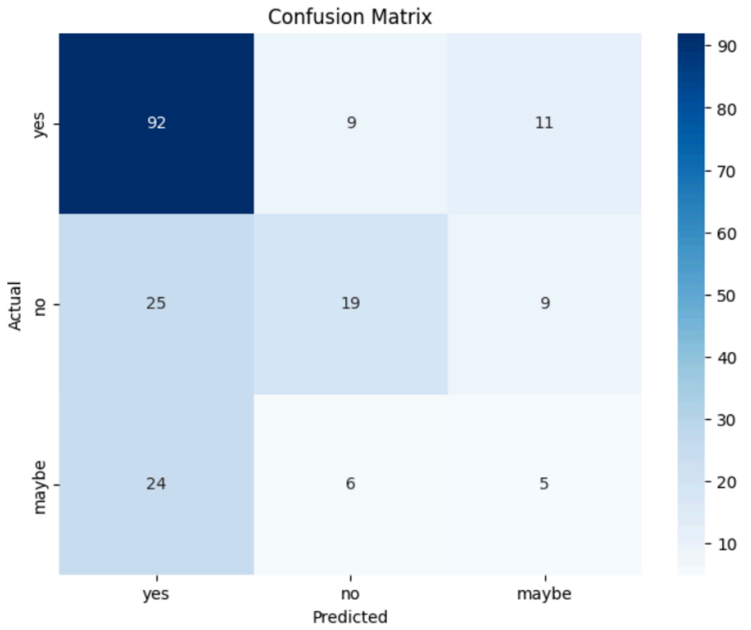
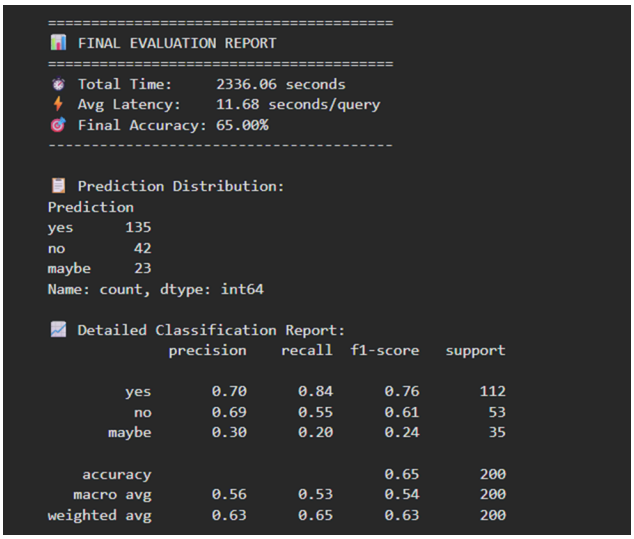


Figure 5: Confusion Matrix for Plain RAG

3.2 Experimental Performance (Graph RAG)



```
======================================
📊 FINAL EVALUATION REPORT
======================================
⏱️ Total Time:     2336.06 seconds
⚡ Avg Latency:    11.68 seconds/query
🎯 Final Accuracy: 65.00%
--------------------------------------

📋 Prediction Distribution:
Prediction
yes      135
no        42
maybe     23
Name: count, dtype: int64

📈 Detailed Classification Report:
              precision    recall  f1-score   support

         yes       0.70      0.84      0.76       112
          no       0.69      0.55      0.61        53
       maybe       0.30      0.20      0.24        35

    accuracy                           0.65       200
   macro avg       0.56      0.53      0.54       200
weighted avg       0.63      0.65      0.63       200
```

Figure 6: Evaluation Metrics for Graph RAG

The Graph RAG system demonstrated a clear performance advantage, achieving an accuracy of **65.00%**. A statistically significant improvement of **+12.1%** over the baseline. The most critical finding was the improvement in the "No" class recall, which rose from 36% in Plain RAG to **55%** in Graph RAG. The Graph RAG pipeline was approximately 50% faster, completing the benchmark in 2,336 seconds compared to 4,556 seconds for the baseline.

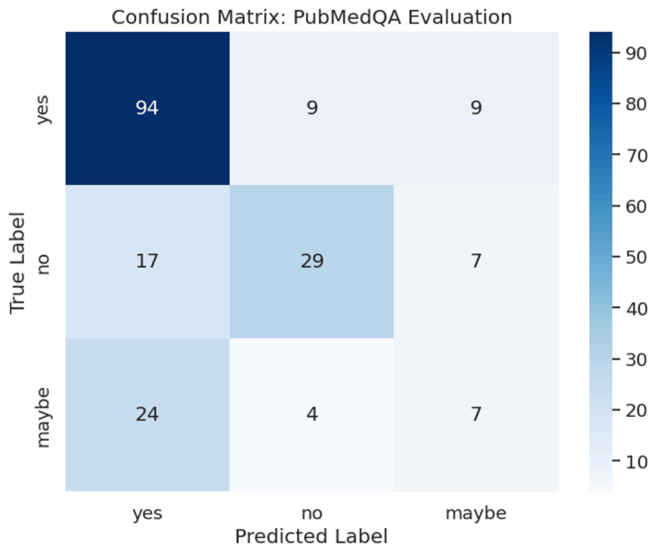


Figure 7: Confusion Matrix for Graph RAG

4. Discussion

4.1 Keyword Matching vs. Reasoning

The comparison highlights a critical flaw in Plain RAG: reliance on surface-level keyword overlap. The baseline model frequently hallucinated correlations because the question and abstract shared medical terminology, leading to a "Yes" prediction even when the abstract concluded no correlation existed. Graph RAG, by leveraging structured data and re-ranking, successfully filtered these false positives, acting as a "skeptical reasoner" rather than a keyword matcher.

4.2 Handling Ambiguity

While Graph RAG outperformed Plain RAG, both systems struggled to accurately classify "Maybe." scenarios (Plain RAG: 0.14 recall; Graph RAG: 0.20 recall). This indicates that while graphs improve the detection of explicit relationships (Yes/No), detecting inconclusive evidence remains a challenge for

both architectures.

Metric	Plain RAG	Graph RAG	Improvement
Accuracy	**58.00%**	**65.00%**	**+12.1%** (Significant)
Total Time	4556.10s	2336.06s	**~50% Faster**
'No' Recall	36%	55%	**+52.8%** (Massive)
'Yes' Recall	82%	84%	+3.7% (Marginal)

Figure 8: Comparative Metrics Table

5. Conclusion

This comparative study confirms that Graph RAG offers a superior alternative to Plain RAG for biomedical question answering. The evaluation showed that Graph RAG not only improved overall accuracy by 12.1% but also effectively mitigated the "Yes-Man" hallucination bias, improving the identification of negative results by 52.8%. These findings suggest that for complex scientific domains, the structured context provided by knowledge graphs is essential for accurate retrieval and reasoning.

GitHub Link: https://github.com/Akash-Raghavendra/Knowledge_Graph_Question_Answering