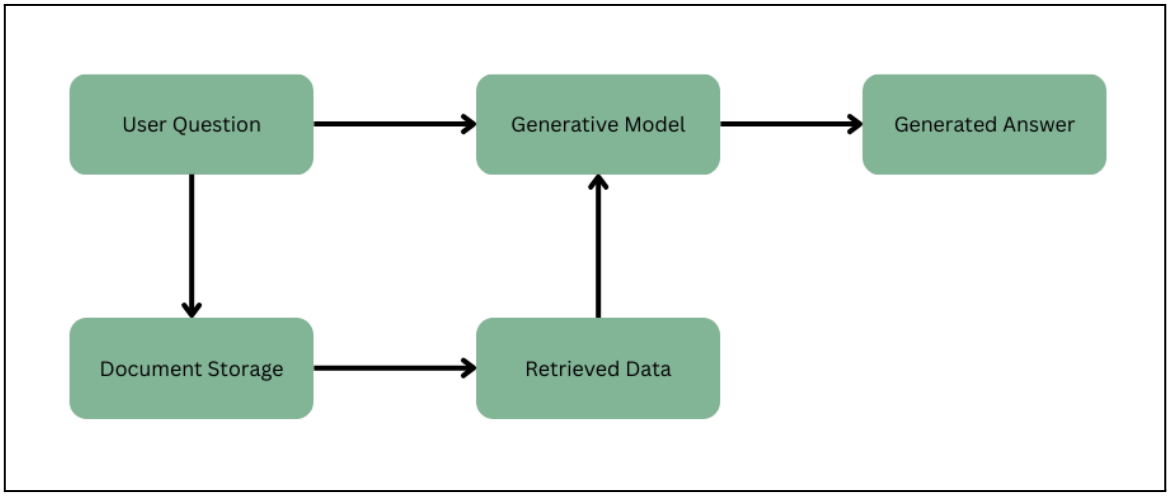
Assumption

By "question answering system," I assume we are considering a RAG-based retrieval system. It contains a similarity search, retrieval component, and answer generation. We should focus on evaluating both the retrieval component and the generation component.

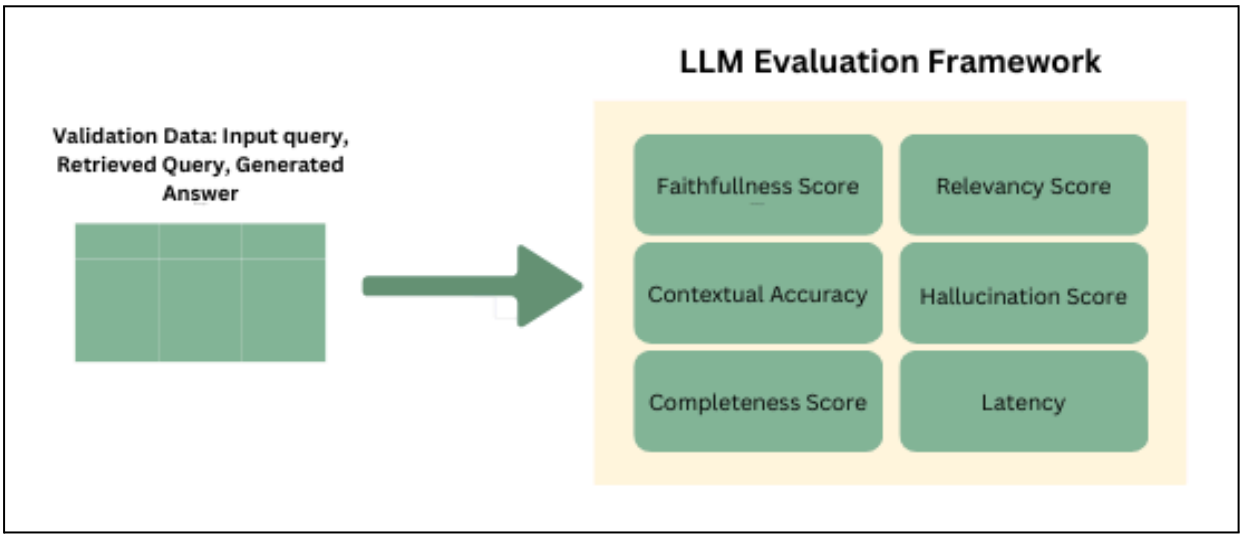
RAG Based Retrieval



Metrics to Consider

As I mentioned above, to trust the output generated, we need to assess how faithful the generated answer is to the document. LLM evaluation metrics, such as answer correctness, semantic similarity, hallucination, coverage, completeness of the answer, and latency, are critical for evaluating an LLM system's output based on the criteria you care about.

The figure shown below provides a high-level abstract structure of an LLM performance evaluation framework. Let's review each metric's details as we move forward in this document.

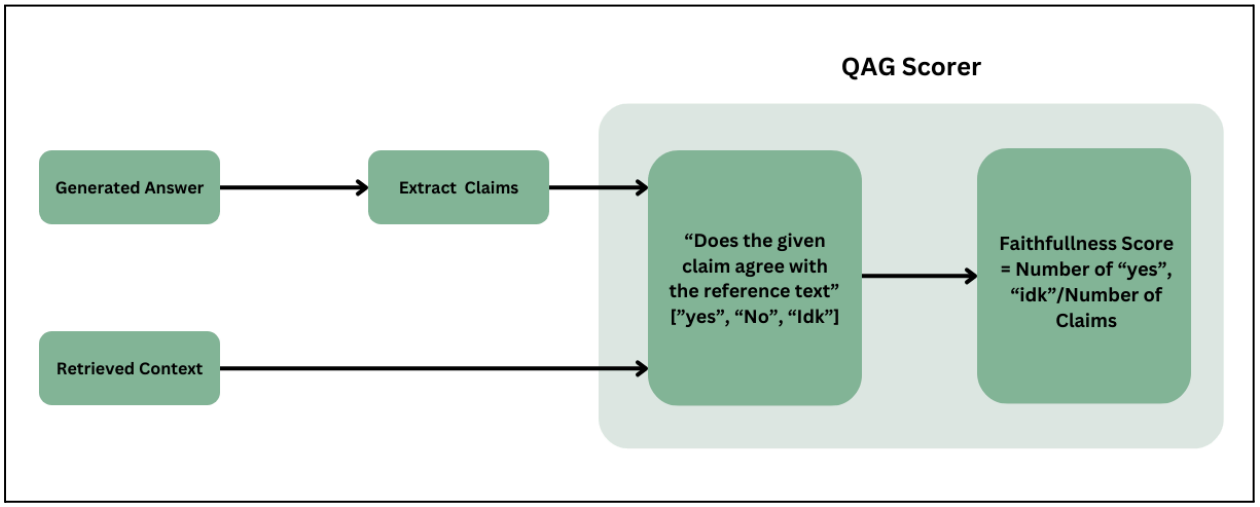


Faithfulness Score *:*

This score evaluates whether the LLM/generator in your RAG pipeline produces outputs that factually align with the information retrieved. Faithfulness, defined as the proportion of truthful claims in an LLM output relative to the retrieval context, can be calculated using the following algorithm:

1. Extract all claims made in the output using LLMs.
2. For each claim, check if it agrees or contradicts each node in the retrieval context.
    - The close-ended QAG question will be: "Does the given claim agree with the reference text?" (Answer: yes, no, or idk).
3. Add up all truthful claims (yes and idk) and divide by the total number of claims made to get the faithfulness score.
4. 1 - faithfulness score to get the hallucination score

The faithfulness score ranges between **0 and 1**, where **1** indicates a perfectly faithful system with complete alignment to the retrieval context, and **0** indicates total hallucination, meaning the output contains no factual basis from the retrieved information.



For info : Hallucination Score:

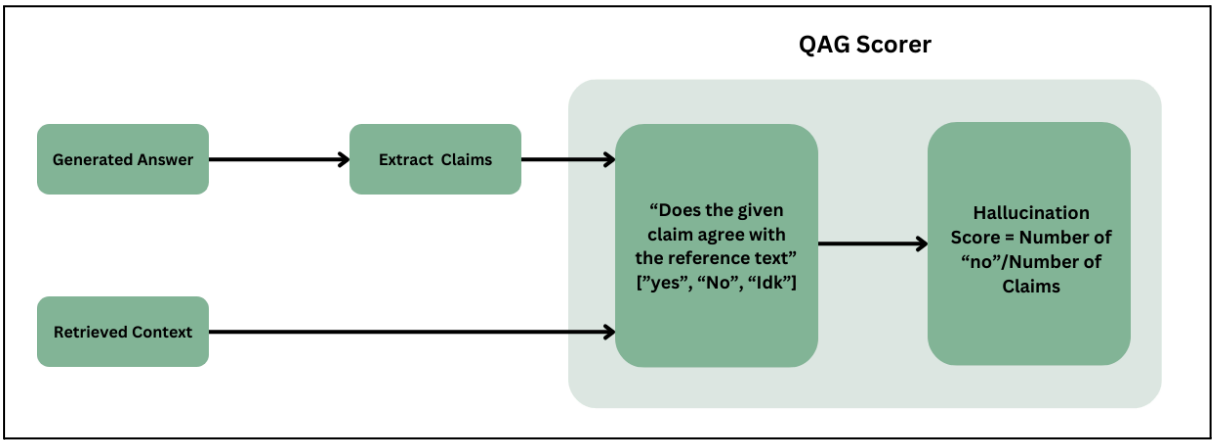
The Hallucination score is designed to quantify the proportion of incorrect or fabricated information generated by a model about the context provided.

Calculation:

1. For each claim generated by the LLM, you would ask the system to evaluate it against the context with a close-ended QAG query like: *"Does the claim '[generated claim]' agree with the context '[retrieved node]'?"*
2. Responses are confined to "yes", "no", or "idk" (where "idk" indicates no relevant context to answer).

3. **Contradicted claims** are those where the system answers "no" for a claim about the relevant context.
4. Or 1 - faithfulness score

A score close to 1 would indicate a high level of hallucination, while a score near 0 would indicate high factual alignment with the context.



***Relevancy Score*:**

This is a RAG metric that assesses the conciseness and relevance of the output. It evaluates how well the generated answers align with the input.

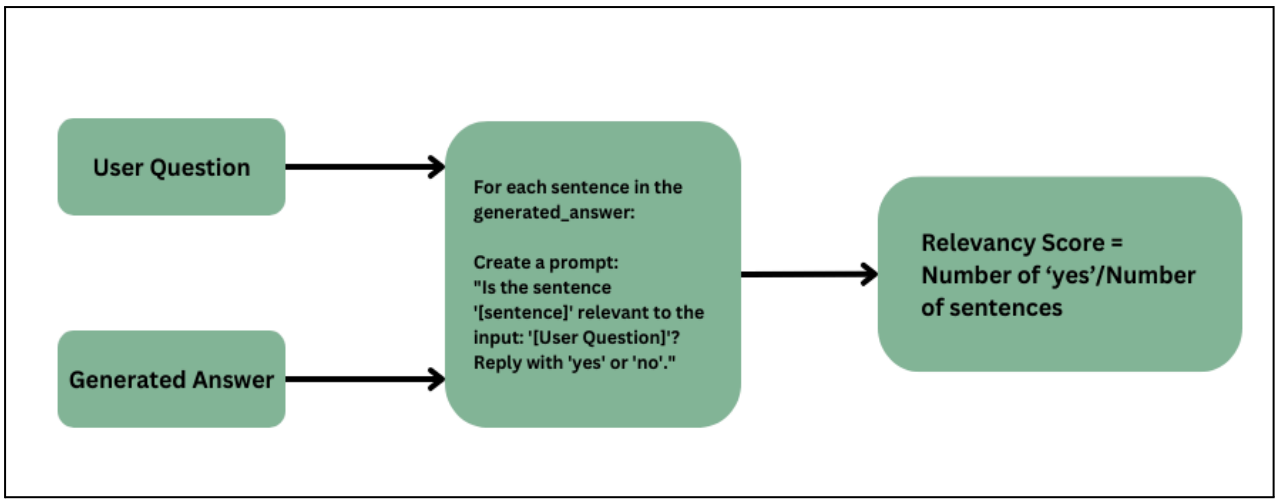
Calculation:

1. Count the number of relevant sentences in the LLM output.
2. Divide that by the total number of sentences.

This score varies between **0 and 1**, where:

- **1** indicates that all sentences in the output are relevant to the input.
- **0** indicates that none of the sentences are relevant.

The higher the score, the more concise and relevant the generated answer is to the given input.



Contextual Accuracy:

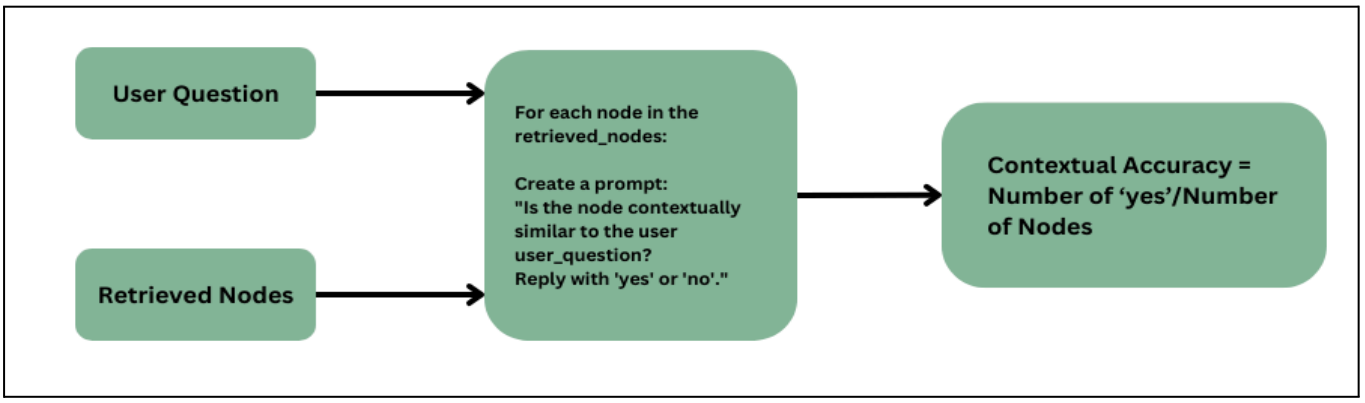
Assesses the quality of the RAG pipeline's retriever by focusing on the relevance of the retrieved context.

Calculation :

1. Obtain a list of nodes from the RAG pipeline based on the user query.
2. **Evaluate Each Node**:
    a. Loop through each retrieved node.
    b. Create a QAG prompt to assess contextual similarity to the user query.
    c. Use the QAG model to determine if each node is contextually similar, incrementing the relevant node counter for each positive response.
3. Divide that by the relevant node count by number of nodes in total.

The score is calculated as the proportion of relevant nodes to the total number of retrieved nodes. The score ranges from **0 to 1**, where:

- **1** indicates all retrieved nodes are contextually similar to the user query.
- **0** indicates none of the retrieved nodes are contextually similar.



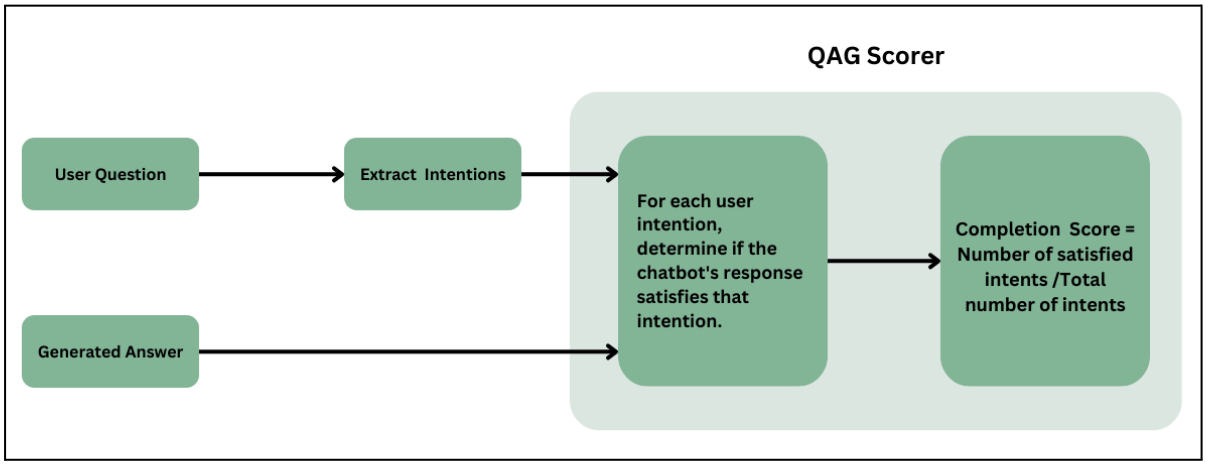
Completeness Score :

It evaluates how well an LLM chatbot satisfies user intentions during a conversation.
It measures the proportion of user intentions the LLM successfully meets throughout the conversation.

Calculation:

1. Use an LLM to extract a list of high-level user intentions from each turn of the conversation.
2. Use the LLM to check whether each intention was met by evaluating the responses provided by the chatbot.
3. If the LLM's response satisfies the user's intention, mark it as "satisfied."
4. Completeness Score = Number of Satisfied User Intentions / Total Number of User Intentions

A high **Completeness score** indicates that most user intentions were successfully fulfilled by the LLM, while a lower score suggests that many of the user's needs were left unmet.

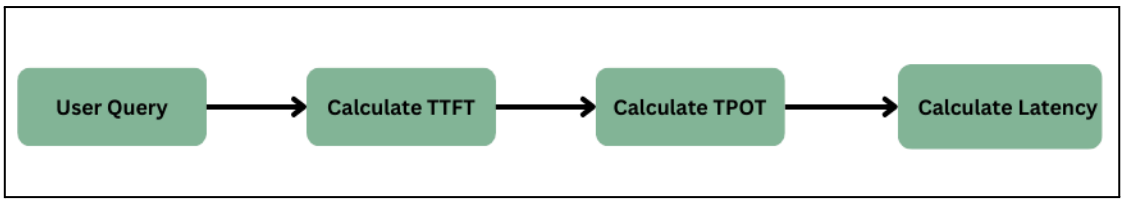


Latency:

Latency represents the total time taken for the model to generate the complete response for a user.

Calculation:

1. **Calculate Time To First Token (TTFT)**:
   - **TTFT** = Time when the first token is generated - Time when the query is submitted.
2. **Calculate Time Per Output Token (TPOT)**:
   - **TPOT** = Total time for full response / Total number of tokens.
3. **Calculate Overall Latency**:
   - **Latency** = TTFT + (TPOT × Number of Tokens).



References :

1. Evaluating Large Language Models: A Comprehensive Survey, Link
2. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena, Link
3. An Empirical Comparison of LM-based Question and Answer Generation Methods, Link
4. LLM Evaluation Metrics: The Ultimate LLM Evaluation Guide, Link