25 TYPES OF RAG-ARCHITECTURES

-By Habib Shaikh

1. Rule-Based RAG

Define:

Uses preDefined rules to retrieve relevant information and generate responses based on these rules.

Benefit:

Ensures structured and predictable outputs.

Best Scenario:

When clear and consistent rules govern the information.

Flow Diagram:

1. User Query

User provides a question.

2. Rule Check

System applies rules to query.

3. Retrieve Relevant Document

Fetches data based on the rule.

4. Generate Response Using Rule

Uses rule-based logic to generate a response.

5. Response Output

Outputs the final answer.

Habib Shaikh
AI Expert

2. Conversational RAG

Define:

Focuses on enabling back-and-forth interactions between users and the system, integrating previous context into the conversation.

Benefit:

Enhances conversational flow and user engagement.

Best Scenario:

Used in chatbots and virtual assistants.

Flow Diagram:

1. User Query

User asks a question.

2. Context Check

Retrieves past conversation context.

3. Retrieve Relevant Data

Fetches related info from the context.

4. Generate Response

Generates a conversation-based answer.

5. Response Output

Delivers response to the user.

Habib Shaikh
AI Expert

3. Iterative RAG

Define:

Continuously refines the response through multiple iterations, improving accuracy with each loop.

Benefit:

Provides increasingly accurate and refined answers over time.

Best Scenario:

Used in tasks requiring high precision over time.

Flow Diagram:

1. User Query

User submits a question.

2. Initial Response

Generates the first answer.

3. Iterate

Refines response by revisiting data or using feedback.

4. Generate Final Response

Delivers improved response.

5. Response Output

Provides refined answer.

Habib Shaikh
AI Expert

4. HybridAI RAG

Define:

Combines generative models and retrieval techniques for better handling of complex queries.

Benefit:

Provides the flexibility of both pre-trained models and dynamic data retrieval.

Best Scenario:

When handling diverse types of user queries that require a balance of generalization and specificity.

Flow Diagram:

1. User Query

User submits a question.

2. Data Retrieval

Fetches relevant data from external sources.

3. Generate Response

Combines retrieved data with generative capabilities.

4. Response Output

Final answer provided to the user.

Habib Shaikh
AI Expert

5. Generative AI RAG

Define:

Uses large generative models (e.g., GPT) to generate responses based on the retrieved data.

Benefit:

High flexibility in creating diverse, context-aware responses.

Best Scenario:

Complex question-answering where creativity and coherence are needed.

Flow Diagram:

1. User Query

User asks a question.

2. Retrieve Data

Retrieves related documents or data.

3. Generate Response

Uses generative model to form a response.

4. Response Output

Delivers the response.

Habib Shaikh
AI Expert

6. Explainable AI (XAI) RAG

Define:

Focuses on making the reasoning behind the generated response transparent and interpretable.

Benefit:

Ensures trust and clarity in AI-generated decisions.

Best Scenario:

When the user needs to understand why a particular answer was given.

Flow Diagram:

1. User Query

User asks a question.

2. Retrieve Data

Relevant documents are fetched.

3. Generate Response

Generates a response with reasoning.

4. Explain Response

Provides an explanation for the generated answer.

5. Response Output

Delivers the answer with explanation.

Habib Shaikh
AI Expert

7. Context Cache LLM RAG

Define:

Uses a cache to store and retrieve contextual information for large language models to generate responses.

Benefit:

Improves efficiency and context awareness.

Best Scenario:

Large-scale applications where the context may change rapidly.

Flow Diagram:

1. User Query

User inputs query.

2. Check Cache

Check for stored context.

3. Retrieve Data

Retrieves data from cache or external sources.

4. Generate Response

Uses context for accurate response.

5. Response Output

Delivers the response.

Habib Shaikh
AI Expert

8. Grokking RAG

Define:

A deeper integration of cognitive models to ‘grok’ or fully understand the user query before generating an answer.

Benefit:

More empathetic and context-aware responses.

Best Scenario:

Complex, nuanced interactions requiring human-like understanding.

Flow Diagram:

1. User Query

User provides input.

2. Contextual Understanding

Deeply interprets user’s question.

3. Retrieve Data

Retrieves relevant data from the knowledge base.

4. Generate Response

Creates a thoughtful, nuanced response.

5. Response Output

Delivers the empathetic answer.

Habib Shaikh
AI Expert

9. Replug Retrieval Feedback

Define:

Uses feedback from retrieved information to improve the relevance of future retrievals.

Benefit:

Provides feedback loops to enhance data relevance.

Best Scenario:

When ongoing refinement of data relevance is required.

Flow Diagram:

1. User Query

User asks a question.

2. Initial Retrieval

Retrieves relevant data.

3. Feedback Loop

User feedback adjusts retrieval.

4. Generate Response

Final response based on improved retrieval.

5. Response Output

Delivers the answer.

Habib Shaikh
AI Expert

10. Attention Unet RAG

Define:

Integrates an attention mechanism to refine the relevance of retrieved data using the Unet architecture.

Benefit:

Improves focus on the most relevant data during generation.

Best Scenario:

Complex data extraction where attention to specific data points is crucial.

Flow Diagram:

1. User Query

User submits query.

2. Attention Mechanism

Focuses on relevant parts of the input data.

3. Retrieve Data

Retrieves focused data points.

4. Generate Response

Uses attention-modulated data to generate response.

5. Response Output

Provides focused output.

Habib Shaikh
AI Expert

11. Corrective RAG

Define:

Corrects or adjusts generated responses based on feedback or additional context.

Benefit:

Ensures accuracy by refining responses through correction mechanisms.

Best Scenario:

Applications that require continuous improvement based on user input or error detection.

Flow Diagram:

1. User Query

User asks a question.

2. Initial Response

Generate an initial response.

3. Error Detection

Check if the response has errors.

4. Apply Correction

Adjust response based on identified issues.

5. Response Output

Deliver the corrected answer.

Habib Shaikh
AI Expert

12. Speculative RAG

Define:

Makes educated guesses about the answer when retrieval data is insufficient.

Benefit:

Can provide answers in cases where full data retrieval is unavailable.

Best Scenario:

Used in situations where answers need to be inferred or speculated.

Flow Diagram:

1. User Query

User asks a question.

2. Retrieve Data

Attempt to retrieve data.

3. Generate Speculative Response

Generate response based on speculation.

4. Provide Output

Output the speculative response.

Habib Shaikh
AI Expert

13. Agenetic RAG

Define:

Uses evolutionary algorithms to iteratively improve the quality of generated responses.

Benefit:

Continuously enhances response quality by simulating natural selection.

Best Scenario:

Long-term improvement and optimization of system responses.

Flow Diagram:

1. User Query

User provides a query.

2. Initial Response

Generate a response.

3. Evolutionary Feedback

Use feedback to evolve response generation.

4. Generate Enhanced Response

Produce an optimized answer.

5. Response Output

Deliver the final output.

Habib Shaikh
AI Expert

14. Self-RAG

Define:

A self-correcting architecture where the model refines its response without external feedback.

Benefit:

Reduces dependency on external corrections, increasing autonomy.

Best Scenario:

Tasks requiring ongoing learning without human intervention.

Flow Diagram:

1. User Query

User submits a question.

2. Initial Response

Generate initial response.

3. Self-Refinement

Refine response by the model's own learning.

4. Generate Final Response

Produce the final, improved answer.

5. Response Output

Output the final answer.

Habib Shaikh
AI Expert

15. Adaptive RAG

Define:

Adapts the retrieval and generation process based on changing contexts or evolving queries.

Benefit:

Provides more personalized and dynamic responses based on context.

Best Scenario:

Used in real-time interactions that require adaptability.

Flow Diagram:

1. User Query

User submits query.

2. Context Assessment

Evaluate the context and adapt approach.

3. Retrieve Data

Gather relevant information based on context.

4. Generate Response

Generate a tailored response.

5. Response Output

Deliver dynamic output.

Habib Shaikh
AI Expert

16. Refeed Retrieval Feedback RAG

Define:

Refines the retrieval process based on user feedback, continuously improving the retrieval mechanism.

Benefit:

Optimizes the search and retrieval process by learning from user interactions.

Best Scenario:

Used in environments where constant feedback improves retrieval accuracy.

Flow Diagram:

1. User Query

User provides a query.

2. Initial Retrieval

Fetches initial data.

3. User Feedback

Collect user feedback on relevance.

4. Refined Retrieval

Adjust retrieval based on feedback.

5. Response Output

Deliver a refined answer.

Habib Shaikh
AI Expert

17. Realm RAG

Define:

Enhances language generation by integrating a powerful retrieval mechanism for more relevant and accurate responses.

Benefit:

Combines the power of large language models with real-time data retrieval.

Best Scenario:

Answering open-domain questions or tasks requiring updated knowledge.

Flow Diagram:

1. User Query

User submits a question.

2. Retrieve Data

Fetch relevant documents or data.

3. Generate Response

Use language model to generate an accurate response.

4. Response Output

Deliver the generated answer.

Habib Shaikh
AI Expert

18. Raptor (Tree-Organized Retrieval) RAG

Define:

Uses a tree-like structure for organizing and retrieving information efficiently, optimizing response generation.

Benefit:

Enhances retrieval efficiency by structuring data hierarchically.

Best Scenario:

Used in situations where data can be logically grouped or organized in trees.

Flow Diagram:

1. User Query

User submits a query.

2. Tree-Based Retrieval

Fetch relevant data using a hierarchical structure.

3. Generate Response

Generate response from the retrieved information.

4. Response Output

Deliver the final output.

Habib Shaikh
AI Expert

19. Memo RAG

Define:

Incorporates memory mechanisms to store relevant past interactions or data for future use.

Benefit:

Improves context retention for future queries, offering more personalized responses.

Best Scenario:

When long-term context is necessary for better interaction history.

Flow Diagram:

1. User Query

User submits a question.

2. Retrieve Contextual Memory

Fetches relevant past interactions.

3. Generate Response

Generate a response based on both current and past data.

4. Update Memory

Store new interaction for future use.

5. Response Output

Deliver the personalized answer

Habib Shaikh
AI Expert

20. Attention-Based RAG

Define:

Uses attention mechanisms to prioritize certain parts of the data or query to improve response generation.

Benefit:

Enhances response relevance by focusing on key data points.

Best Scenario:

When a query contains multiple elements, but certain parts are more important.

Flow Diagram:

1. User Query

User asks a question.

2. Attention Mechanism

Identifies relevant parts of the input.

3. Retrieve Data

Fetches relevant data using attention.

4. Generate Response

Focuses on the important information during generation.

5. Response Output

Deliver focused and relevant output.

Habib Shaikh
AI Expert

21. RETRO RAG

Define:

Enhances the transformer model by integrating retrieval techniques to improve text generation.

Benefit:

Improves the relevance and accuracy of generated text.

Best Scenario:

Used in tasks like open-domain question answering and large-scale text generation.

Flow Diagram:

1. User Query

User asks a question.

2. Retrieve Data

Retrieve relevant documents.

3. Generate Response

Use transformer to generate a response with retrieved data.

4. Response Output

Deliver the final answer.

Habib Shaikh
AI Expert

22. Auto RAG

Define:

Automates the retrieval and response generation process, with minimal human intervention.

Benefit:

Improves efficiency and scalability for automated systems.

Best Scenario:

Used in systems where responses need to be generated automatically at scale.

Flow Diagram:

1. User Query

User submits a query.

2. Automatic Retrieval

System automatically fetches relevant data.

3. Generate Response

Automatically generates a response.

4. Response Output

Provide automatic output.

Habib Shaikh
AI Expert

23. Cost-Constrained RAG

Define:

Focuses on limiting resource usage (e.g., computation, memory) during retrieval and response generation.

Benefit:

Ensures cost efficiency in systems with resource constraints.

Best Scenario:

Environments where computational resources are limited.

Flow Diagram:

1. User Query

User provides a query.

2. Retrieve Data

Perform retrieval within resource limits.

3. Generate Response

Generate response within the cost constraints.

4. Response Output

Deliver the response efficiently.

Habib Shaikh
AI Expert

24. ECO RAG

Define:

Focuses on ecological or environmental efficiency, minimizing energy consumption during retrieval and response generation.

Benefit:

Reduces environmental impact while processing requests.

Best Scenario:

Used in large-scale systems where sustainability is a concern.

Flow Diagram:

1. User Query

User submits a query.

2. Efficient Retrieval

Optimize data retrieval for energy efficiency.

3. Generate Response

Generate a response while minimizing resource usage.

4. Response Output

Deliver a sustainable output.

Habib Shaikh
AI Expert

25. Replug (Retrieval Plugin) RAG

Define:

Allows for easy integration of external retrieval systems via plugins, enhancing the system’s flexibility.

Benefit:

Increases modularity and allows seamless integration with various data sources.

Best Scenario:

Systems that need to integrate with multiple external data sources.

Flow Diagram:

1. User Query

User submits a query.

2. Retrieve Data via Plugin

Use external plugins to fetch relevant data.

3. Generate Response

Generate response based on retrieved information.

4. Response Output

Deliver the final response.

Habib Shaikh
AI Expert