RAG / Local LLM

Jonathan Respeto

Who Am I

Jonathan Respeto -

respondo on discord | @jrespeto on X | jrespeto LinkedIn

What I like to do: Camping, scuba diving, DevOps, Programming in python :)

What do I do:

Security Intelligence Response Team Engineer Senior @ akamai

Volunteer work: HackMiami | Defcon |

What is the talk about

- Exploring RAG (Retrieval-Augmented Generation)
  - How vector-based retrieval enhances Large Language Model responses
- Leveraging Local LLMs
  - Why on-device models matter for privacy, control, and efficiency
- Tool Stack Overview
  - Ollama, OpenWebUI, LangFlow, and QdrantDB—and how they fit together
- Real-World Applications
  - Building custom chatbots, managing knowledge, summarizing documents, and more
- Hands-On Insights (Domes)
  - Best practices for designing, deploying, and scaling RAG workflows in your own projects
- Questions….

https://github.com/jrespeto/RAG_Local-LLM.git

“Retrieval Augmented Generation” RAG at a Glance

RAG: a technique that enhances the capabilities of a large language model (LLM) by allowing it to access and incorporate relevant information from external knowledge bases before generating a response, resulting in more accurate and contextually relevant outputs

Useful: context-aware, accurate, and more efficient responses.

Highlight the value: Minimizes hallucination, and improves performance by leveraging relevant document snippets.

https://aws.amazon.com/what-is/retrieval-augmented-generation/#:~:text=Retrieval%2DAugmented%20Generation%20(RAG),and%20useful%20in%20various%20contexts.

RAG Workflow

A simple schematic showing:

1. User Query → 2. Vector Database (Qdrant) → 3. Relevant Context → 4. LLM Response

Key Points:

Emphasize the “retrieve” step vs. the “generate” step.

Outline the role of embeddings and vector search.

https://github.com/jrespeto/RAG_Local-LLM.git

Embeddings what?

An embedding is a numerical representation of data—often text—that captures its core meaning or context in a vector (a list of numbers). By converting text into vectors, you can compare the similarity or relatedness of different pieces of text mathematically. This is a key step in many AI workflows, including RAG, because it allows you to efficiently find relevant information or content based on semantic meaning rather than just keyword matches.

https://github.com/jrespeto/RAG_Local-LLM.git

Why Local LLMs?

- Data Privacy: Sensitive data never leaves your environment.
- Customization: Tailor models to domain-specific needs.
- Latency & Control: Lower network dependencies; you control hardware and infrastructure.

https://github.com/jrespeto/RAG_Local-LLM.git

Meet Ollama & OpenWebUI

- Ollama:
  - Manages local LLM lifecycle: easy downloading, managing, and running models locally.
  - No API Keys or Tokens :(

- OpenWebUI:
  - Provides a user-friendly interface for working with various LLMs.
  - Facilitates quick prototyping and testing of prompts. (chat)
  - API with API Keys

https://github.com/jrespeto/RAG_Local-LLM.git

Meet LangFlow & QdrantDB

- LangFlow:
    - Use a drag-and-drop interface to design, debug, and visualize LLM pipelines.
    - Allows quick iteration on prompt engineering and workflow design.

- QdrantDB:
    - High-performance vector database for storing and querying embeddings.
    - Scalable, efficient retrieval to power RAG workflows.

https://github.com/jrespeto/RAG_Local-LLM.git

Real-World Use Cases

Custom Chatbots: Domain-specific Q&A with full control over data.

Knowledge Management: Ingest large collections of internal docs; quickly retrieve context.

Summarization: Summarize lengthy reports with context from relevant sections.

Content Creation: Generate targeted content with domain-specific knowledge.

https://github.com/jrespeto/RAG_Local-LLM.git

RAG Workflow in Action

Load Documents into QdrantDB (embeddings, indexing).

User Query hits the LLM.

RAG Pipeline uses Qdrant to retrieve relevant context.

Final Answer is synthesized by the LLM.

https://github.com/jrespeto/RAG_Local-LLM.git

Pros & Potential Pitfalls

- Benefits:
  - Full data control, reduce reliance on external APIs.
  - Lower cost over time for large-scale usage.
- Challenges:
  - Hardware requirements for larger models.
  - Ongoing model updates and maintenance.
  - Prompt engineering complexity.

https://github.com/jrespeto/RAG_Local-LLM.git

Ask Away!

Invite questions and discussion.

GitHub repos: https://github.com/jrespeto/RAG_Local-LLM.git

X: @jrespeto