Rag Application and its use cases

1. What is RAG?
    - Full form: retrieval argument, generation
    - It is the retrieval of relevant data from the source according to user requirement
    - Referred to information outside of its training data source

2. What is similarity search?
    - A type of search which composers the similarity using queries
    - Pardon is a technique to convert data into chunks or break them down into different parts
    - Embedding is a technique in which the texts are converted into vectors
    - Vectors are array of numbers

3. What is vector indexing and TOON?
    - The converted data from normal text to array of numbers
    - TOON: token oriented object notation, this technique is used in AI to reduce the number of tokens used. This is used instead of JSON

4. Vector databases:
    - These are databases which are used to store the vector index that are converted from the basic texts

Basic RAG chatbot using GROQ Api integration

Workflow of the project:

Phase 1:

1. Document indexing: In this step, the user uploads the document through the chatbot, it can be a PDF file.

2. Text chunking: the document is the condo into smaller parts, smaller, the chunk more precise retrieval

3. Embedding generation: convert to vector 384– D, can perform semantic search.

4. Vector storage: once the document is broken into chunks and converted into vectors, the vectors is stored in chroma DB

Phase 2:

1. User question processing: user enters a prompt asking a question or to summarize the file according to the user requirement.

2. Semantic retrieval: the text prompt, which is entered by the user is connected into embedding, and a similarity search is done.

3. Context argumentation: retrieval is converted into context.

4. LLM generation: the context and the question is sent to the groq api, LLM processes and gives us suitable answer

Tech Stacks used in this project

1. Frontend: Steamlit(streamlit>=1.31.0)
    - Role : web application frame work
    - What it does: create user interface
    - Features in the project:
        - Chat interface with message history
        - File upload, widget(PDF/TXT)
        - Sir bar for configuration
        - Real time response streaming
        - Session state management

2. Backend : LangChain(langchain>=0.1.9):
    - Role: RAG pipeline framework
    - What it does: connect all the AI components together
    - Key features used:
        - LCEL
        - Document processing
        - Prompt templates
        - Output parsing

3. AI/LLM layer- Groq API:
    - Role: ultra fast large language, model interface
    - Model used:llama-3.3-70b-versatile

- What it does: generates intelligent answers based on document context

4. Vector database layer: ChromaDB(chromadb>=0.4.22)
    - Role : database for semantic search
    - What it does:
        - Stores document Chung as vector embedding
        - Perform similarity source to find relevant context
        - Persist data to disk

5. Optimization layer: TOON(token oriented object notation -toons>=0.5.0)
    - Role: token optimization for LLM context
    - What it does:
        - Compresses document context before sending to Groq
        - Reduce the token usage, saves memory, and improved speed
        - Can save up to 30 to 50% of tokens