Lecture No. 1

■ Summary

- The video is part of a series focusing on "rag" and will practically create a rag-based system using "lang chain".

- The problem statement is about creating a chat system through which users can discuss various YouTube videos in real-time.

- The system can be created as a Chrome plugin or a Streamlet website, depending on the user's knowledge of HTML, CSS, and JavaScript.

■ Key Terms and Concepts

- rag-based system

- lang chain

- YouTube chat system

- Chrome plugin

- Streamlet website

- HTML

- CSS

- JavaScript

■ Review Questions

1. What was the problem statement discussed in the video?

2. What are the different ways in which the chat system can be designed?

3. What knowledge is needed to create the chat system as a Chrome plugin?

4. What is the alternative approach to creating the chat system if one does not know HTML, CSS, and JavaScript?

5. Why is the chat system considered a useful solution for engaging with YouTube videos?

■ Summary

- Focus on building a RAG system and making it inside a Google Colab notebook.

- Use tools like Streamlet and Langchain for loading the video transcript.

- Steps include loading the transcript, dividing it into chunks, generating embeddings, cr eating a retriever, merging relevant chunks with the query, and sending the prompt to LLM for response generation.

■ Key Terms and Concepts

- Streamlet

- Google Colab notebook

- RAG system (Retrieval-augmented generation)

■ Review Questions

1. What are the key steps in building a RAG system?

2. What tools can be used for loading the transcript of a video?

3. What are the main features of a retriever in the RAG system architecture?

■ Summary

- The tutorial explains a step-by-step process of using an API to load the transcript of a YouTube video.

- To achieve this, it involves obtaining the video ID, retrieving the transcript via the Y ouTube transcript API, and processing the transcript data.

- The time stamp and the duration of text visibility on the screen are shown using Python code.

■ Key Terms and Concepts

- YouTube transcript API

- Video ID

- Python API usage

- Retrieving the transcript

- Text visibility and duration

- Timestamp-based transcript loading

■ Review Questions

1. What are the key steps involved in using the YouTube transcript API to retrieve a video 's transcript?

2. How is the ID of a YouTube video obtained for transcript retrieval?

3. What is the significance of the try-accept block in the Python code?

4. What information is available in the API-converted transcript related to text visibilit y and duration?

5. In what ways did the speaker find this approach to be the best for obtaining video tran scripts?

■ Summary

- Converting a transcript into smaller chunks

- Joining the smaller chunks

- Loading the Hindi transcript for the video

- Using a text splitter to divide the transcript into smaller chunks

- Using a vector store to store the chunks as vectors

■ Key Terms and Concepts

- Transcript

- Text splitter

- Recursive character text splitter

- Chunk size and overlap

- Embedding model

- Open AI embeddings

- Vector store

■ Review Questions

1. How is the transcript converted into smaller chunks?

2. What are the functions used to manipulate the transcript and join the chunks?

3. What kind of embedding model is used to convert the smaller chunks into vectors?

4. What purpose does the vector store serve in this process?

■ Summary

- The video explains the process of using vector embeddings and indexing to create a retriever for document retrieval.

- The retriever embeds a query and searches for the closest matching vectors in the vector store.

- Once the retriever finds the closest vectors, it retrieves the corresponding documents.

■ Key Terms and Concepts

- Vector store

- Embedding and storing chunks

- Creating a retriever

- Retrieval process

- Similarity search

- Prop template

- Retrieved documents

■ Review Questions

1. How does the retriever search for matching vectors in the vector store?

2. What is the purpose of the prop template in the argumentation part?

3. What happens in the retrieval step? How is it associated with the retriever?

4. Explain the process of retrieving documents using the retriever and the query.

■ Summary

- Need to concatenate page content from multiple documents

- Custom code to concatenate page content from each document

- Invoking the custom code with context and final prop

- Generating an answer from the invoked LLM

■ Key Terms and Concepts

- Concatenation of page content

- Custom code

- Invocation of code with context and final prop

- Answer generation from the LLM

- Chaining multiple steps for automation

■ Review Questions

1. What is the purpose of invoking the custom code with context and final prop?

2. How can we automate and streamline the process of calling each step separately?

3. What are the key concepts involved in generating an answer from the invoked LLM?

■ Summary

- The learning journey involves understanding how to create and manage a chain in the invo ke function

- A chain can trigger an entire pipeline automatically, with every step executing and prod ucing an output that serves as input for the next step

- The structure of the chain includes a simple linear flow and two parallel chains

- The construction of a parallel chain involves using Runnable Parallel and defining keys in a dictionary to handle context and question, with the processing being a part of the ch ain

■ Key Terms and Concepts

- Invoke function

- Pipeline automation

- Structure of the chain

- Simple linear flow

- Parallel chains

- Runnable Parallel

- Context and question keys

- Dictionary

- Processing within the chain

■ Review Questions

1. What is the purpose of the invoke function in managing a chain?

2. How is the overall pipeline triggered and executed automatically in the chain structure ?

3. What tools or components are involved in creating a parallel chain?

4. How is the processing of documents integrated into the chain structure?

5. What are the key elements in defining a dictionary within the parallel chain?

■ Summary

- The code implements a chain system to perform indexing, retrieval, and generation.

- A Runnable lambda is used to execute the chain.

- The chain is tested using parallel chain invoke and returns a dictionary with context and question keys.

■ Key Terms and Concepts

- Runnable lambda

- Parallel chain invoke

- Dictionary

- Context and question keys

- Main chain

■ Review Questions

1. What is the purpose of the Runnable lambda in the chain system?

2. Explain the testing process of the chain with parallel chain invoke.

3. What information does the returned dictionary contain?

4. How is the main chain connected to the parallel chain?

■ Summary

- UI-based enhancements for a rack system can be made

- Running a rack system inside a Google Collab notebook, requiring manual user interventio
n

- Final product can be improved to appear as a website or Chrome plugin

- Evaluation of rack systems is critical for industry-grade systems

- Libraries like Rags and Langmith are used for evaluating rack systems

■ Key Terms and Concepts

- UI-based enhancements

- Google Collab notebook

- Chrome plugin

- Evaluation of rack systems

- Rags and Langmith libraries

- Faithfulness, relevance, and context precision in evaluation

- Auto-generated transcripts and their errors

- Document ingestion, document splitting, and vector storage

- Semantic chunker for text splitting

- Vector store libraries like FYERS

- Industry-grade rack systems

■ Review Questions

1. How can a rack system be improved to appear as a website or a Chrome plugin?

2. Why is the evaluation of rack systems important in an industry-grade context?

3. What are some important evaluation metrics for rack systems?

4. How can auto-generated transcript errors be fixed?

5. What are the key stages in the document ingestion process for a rack system?

■ Summary

- Different stages of retrieval in a vector store system

- Tasks before retrieval such as pre-retrieval and multi-query generation

- Domain aware routing for complex rack systems

- Performing reranking in retrieval to improve performance

- Post-retrieval tasks such as contextual compression and answer grounding

- Context window optimization

■ Key Terms and Concepts

- Vector store

- Cloud-based solution

- Pine cone type solution

- Pre-retrieval

- Multi-query generation

- Domain aware routing

- Reranking

- Contextual compression

- Prompt templating

- Answer grounding

- Context window optimization

■ Review Questions

1. What are the different stages of retrieval in a vector store system?

2. How can multi-query generation be beneficial in the retrieval process?

3. What is domain aware routing and when is it used?

4. How does reranking in retrieval improve performance?

5. Explain the concept of answer grounding and its importance in retrieval.

■ Summary

- LLMs process a certain number of tokens in the input

- Context window optimization involves trimming the context to ensure it does not cross the window limitation

- LLM generates the answer, also allowing for citations and guard railing

- Multimodal rack system processes text, images, and videos

- Agentic rack system operates as an AI agent, not just a chatbot

- Memory-based rack system can be personalized based on past interactions

■ Key Terms and Concepts

- LLMs

- Context window optimization

- Multimodal rack system

- Agentic rack system

- Memory-based rack system

- Guard railing

- Advanced Rag

■ Review Questions

1. What is the purpose of context window optimization in LLMs?

2. How does the multimodal rack system differ from the agentic rack system?

3. What is the concept of guard railing in relation to LLM output?

4. How does the memory-based rack system personalize interactions based on past conversati ons?

5. What is the significance of Advanced Rag in the industry?

■ Summary

- Advanced Rag systems will be covered in a separate playlist called Advanced Rag after th e Lang Chain playlist is completed.

- The goal of the video was to explain how to create a simple functional rack system.

■ Key Terms and Concepts

- Lang Chain playlist

- Advanced Rag

- Functional rack system

■ Review Questions

1. When will the Advanced Rag systems be covered?

2. What was the goal of the video?

3. What will be covered in the separate playlist called Advanced Rag?