**Sahil Wani**

SUID: 466848553

Jan 30th report

**Introduction**

The goal of this update was to improve our existing course retrieval system by integrating it with a chatbot powered by Cohere AI. Previously, we had an efficient search mechanism using Pinecone and sentence-transformers, but users could only retrieve course names. The chatbot lacked the ability to provide detailed, natural language responses about courses. To address this, I implemented the following enhancements:

1. Ensuring Pinecone contains up-to-date course data.
2. Using Cohere AI to process user queries more effectively.
3. Enhancing the chatbot to provide a Q&A experience for Syracuse University graduate courses.

**Key Updates**

Retrieving Full Course Information

* Previously, the chatbot only retrieved course names from Pinecone.
* Now, it fetches both course names and descriptions, allowing for more detailed responses.

Improving Cohere’s Response Generation

* Before, Cohere AI generated responses based solely on user input.
* Now, the chatbot first retrieves relevant course data from Pinecone and includes it in the Cohere prompt. This makes responses more informative and context aware.

Enhancing the Chatbot’s Q&A Capabilities

* The chatbot can now answer user queries using actual course descriptions rather than generic LLM knowledge.

**Step-by-Step Guide to Creating the Vector Database**

Step 1: Data Collection

* Course information was sourced from Syracuse University’s iSchool course catalog: <https://courses.syracuse.edu/content.php?catoid=39&navoid=4900&p1170=1#ent_courses1170>
* Each course entry includes:
  + course\_name: The name of the course.
  + description: A detailed summary including credits and key topics.

Step 2: Selecting Pinecone as the Vector Database

* Pinecone was chosen for its ability to store and quickly retrieve high-dimensional embeddings.
* A Pinecone account was created, and API credentials were generated.

Step 3: Setting Up the Environment

* Required dependencies were installed using:
* pip install sentence-transformers pinecone-client numpy cohere streamlit
* Libraries Used:
  + sentence-transformers: Generates embeddings from course descriptions.
  + pinecone-client: Interacts with the vector database.
  + numpy: Handles numerical data.
  + cohere: Generates AI-powered responses.
  + streamlit: Provides the chatbot UI.

Step 4: Generating and Storing Embeddings

1. Initializing Pinecone

A screen shot of a computer

Description automatically generated

1. Embedding Course Descriptions
   * Using sentence-transformers to convert course descriptions into 384-dimensional embeddings:
2. from sentence\_transformers import SentenceTransformer

A black background with yellow and orange letters

Description automatically generated

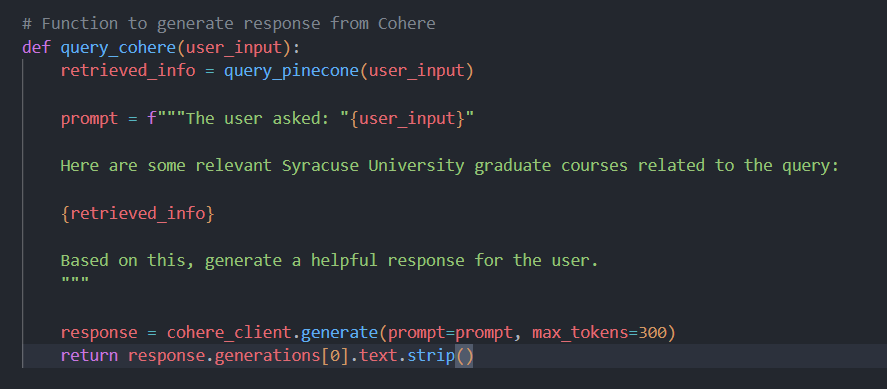
Step 5: Querying the Vector Database

* The chatbot retrieves the most relevant courses based on user queries:

A screen shot of a computer code

Description automatically generatedStep 6: Integrating the Vector Database with Cohere AI

* The chatbot first retrieves relevant course information and includes it in Cohere’s prompt before generating a response:



Is the Application Leveraging the Vector Database or LLM Directly?

* Hybrid Approach: The chatbot first queries Pinecone for relevant course data and then passes that information to Cohere.
* This allows the chatbot to:
  1. Retrieve accurate course information instead of relying solely on the LLM’s pre-trained data.
  2. Provide context-aware responses tailored to the user’s question.
* Vector Database Usage: Pinecone stores embeddings and enables semantic search.
* LLM Usage: Cohere generates natural language responses based on retrieved course data.

Output:

A screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a chat

Description automatically generated