Grand Theft Auto Multimodal RAG application

Problem Statement

Develop a **Grand Theft Auto Multimodal Retrieval-Augmented Generation (RAG) Application** that leverages LanceDB as a vector database, OpenAI's "ViT-L/14" for multimodal embedding, and the GTA-Image-Captioning-Dataset provided. The application should be capable of processing and understanding complex queries related to the Grand Theft Auto universe, providing accurate and contextually relevant responses that combine textual and visual information.

Step-by-Step Guide

Step 1: Understanding the Technologies

- **LanceDB**: Familiarize yourself with LanceDB's capabilities as a vector database to handle efficient storage and retrieval of high-dimensional data.
- **OpenAI's "ViT-L/14"**: Study the "ViT-L/14" model to understand how it can be used to create embeddings for images that capture both visual and textual information.
- **Hugging Face Dataset**: Explore the "vipulmaheshwari/GTA-Image-Captioning-Dataset" to understand the structure and content of the dataset.

Step 2: Setting Up the Environment

- Install and configure LanceDB.
- Set up OpenAI's "ViT-L/14" model for use in your application.
- Download and prepare the GTA-Image-Captioning-Dataset for training and testing.

Step 3: Integrating the Components

- Develop a system to ingest data from the GTA-Image-Captioning-Dataset and create multimodal embeddings using "ViT-L/14".
- Implement a retrieval system using LanceDB to store and query the embeddings effectively.

Step 4: Building the RAG Application

- Design the RAG framework to utilize the multimodal embeddings and LanceDB's retrieval capabilities.

- Ensure the application can handle queries and return responses that integrate both text and image data.

Step 5: Testing and Refinement

- Conduct thorough testing to ensure the application's accuracy and performance.
- Refine the model and retrieval system based on test results.

Step 6: Documentation and Deployment

- Create comprehensive documentation outlining the application's functionality and usage.
- Prepare the application for deployment in a suitable environment, preferably host it using Gradio in Huggingface Spaces.

Resources and Technologies

- **Vector Database**: LanceDB for storing and querying vector embeddings.
- **Multimodal Embedding Model**: OpenAI's "ViT-L/14" for generating embeddings from images.
- **Dataset**: Hugging Face's GTA-Image-Captioning-Dataset for training and testing the application.
  Link: vipulmaheshwari/GTA-Image-Captioning-Dataset · Datasets at Hugging Face

Reference resources:

1. Multi-Vector Retriever for RAG on tables, text, and images (langchain.dev)
2. Multimodal RAG applications - Blixxi Labs (vipul-maheshwari.github.io)