Project Overview

This project demonstrates how to use LangChain to create a question-and-answer (Q&A) agent based on a large language model (LLM) and retrieval augmented generation (RAG) technology.
The project leverages the IBM Watsonx Granite LLM and LangChain to set up and configure a Retrieval Augmented Generation (RAG) pipeline.

Introduction

In the era of information overload, finding accurate and relevant answers to specific questions can be challenging. Traditional search engines often return an overwhelming amount of information, much of which may not be directly relevant to the user's query. This is where Question-and-Answer (Q&A) agents come into play, providing users with precise answers to their questions by leveraging advanced natural language processing (NLP) and machine learning technologies.

The rapid advancements in large language models (LLMs) have significantly improved the capabilities of Q&A systems. However, these models sometimes struggle with providing grounded and contextually accurate answers, especially when dealing with complex or niche queries. This is where Retrieval Augmented Generation (RAG) technology becomes essential. RAG enhances the performance of Q&A agents by retrieving relevant information from external knowledge bases and integrating it into the generation process.

This project demonstrates how to build a sophisticated Q&A agent using LangChain, IBM Watsonx Granite, and RAG technology.

(LangChain provides a robust framework for creating language model-based applications, while Granite offers powerful API capabilities for enhanced retrieval and processing. By combining these tools with RAG, we can create a Q&A agent that not only understands user queries but also delivers precise and contextually enriched answers.)

Background

Large Language Models (LLMs)

Large Language Models (LLMs) are advanced artificial intelligence systems trained on vast amounts of text data. These models, like GPT-4, are capable of understanding, generating, and manipulating human language with high accuracy. LLMs are often based on the Transformer architecture.

Introduced in the famous “Attention is All You Need” paper by Google researchers in 2017, the Transformer architecture is a neural network design that excels in processing sequences of data, such as text.

LLMs are pre trained on a vast array of language tasks—like translation, summarization, and text completion—using extensive datasets. This pretraining helps the models learn the intricacies of language, enabling them to understand and generate human-like text with high accuracy.This makes them able to perform a wide range of language-based tasks. They have transformed various industries by enabling applications like chatbots, virtual assistants, and content creation tools.

LangChain

LangChain is a framework that helps you build applications using large language models (LLMs). It provides tools to easily connect different parts of an AI application, like retrieving information and generating text. With LangChain, you can create complex workflows for natural language processing (NLP) tasks without needing to write a lot of code from scratch. This makes it easier for developers to build sophisticated AI applications quickly and efficiently.

IBM Watsonx Granite

IBM Granite is a family of artificial intelligence (AI) models purpose-built for business, engineered from scratch to help ensure trust and scalability in AI-driven applications. (Source)

These IBM models — built on a decoder-only architecture — aim to help businesses scale AI. For instance, businesses can use them to apply retrieval augmented generation for searching enterprise knowledge bases to generate tailored responses to customer inquiries; use summarization to condense long-form content — like contracts or call transcripts — into short descriptions; and deploy insight extraction and classification to determine factors like customer sentiment. (Source)

Granite offers robust natural language understanding and generation capabilities. Granite is designed to handle a wide range of language tasks, providing powerful APIs for developers to build and integrate AI solutions into their applications. By leveraging Watsonx Granite, developers can enhance their applications with advanced language processing capabilities, ensuring high accuracy and performance.

Retrieval Augmented Generation (RAG)

RAG is an AI framework for retrieving facts from an external knowledge base to ground large language models (LLMs) on the most accurate, up-to-date information and to give users insight into LLMs' generative process.

LLMs, despite being trained on vast datasets, have a fixed knowledge base up until their training cut-off. RAG allows these models to dynamically access external databases or documents, effectively extending their knowledge beyond the training data. This is crucial for providing up-to-date and comprehensive information.

In addition, LLMs sometimes generate plausible but incorrect information, known as hallucinations. RAG mitigates this by providing concrete references from retrieved documents, ensuring the generated responses are based on verified information.

Question-and-Answer (Q&A) Agent

A Q&A agent is an AI system designed to answer user queries by understanding and processing natural language inputs. These agents utilize LLMs and other NLP techniques to interpret questions and generate precise answers. Q&A agents are used in various applications, including customer support, virtual assistants, and educational tools, providing users with quick and accurate information based on their queries.

In reference to a documentation on Hugging Face :

“Question Answering models can retrieve the answer to a question from a given text, which is useful for searching for an answer in a document. Some question answering models can generate answers without context!

Question Answering (QA) models can be used to automate the response to frequently asked questions by using a knowledge base (documents) as context. In other words the answers are derived from the context provided using metrics like exact-match and f1-score.

Read more on Hugging Face”

Libraries Used in This Lab

langchain

LangChain is used for integrating language models and retrieval models. It provides tools and abstractions to connect various components of an AI application, enabling seamless workflows for natural language processing (NLP) tasks. With LangChain, developers can easily combine different models and systems to build sophisticated language-based applications.

ibm-watsonx-ai

This library is used to access the Watsonx Granite language model. It provides APIs to interact with IBM's advanced LLM, which offers robust natural language understanding and generation capabilities. By using ibm-watsonx-ai, developers can leverage Granite's powerful features to enhance their AI applications with high accuracy and performance.

wget

The wget library is used for downloading files from the internet. It allows for easy retrieval of data, scripts, or other resources needed for the lab. This utility is essential for fetching external files and datasets required for processing and analysis.

sentence-transformers

Sentence-Transformers is used for computing dense vector representations (embeddings) for sentences, paragraphs, and images. These embeddings are essential for various NLP tasks, such as semantic search, clustering, and retrieval, as they capture the contextual meaning of the text.

chromadb

ChromaDB is an open-source embedding database used to store and manage embeddings generated by models like Sentence-Transformers. It allows efficient retrieval of similar embeddings, which is crucial for tasks like document search and retrieval augmented generation (RAG).

pydantic

Pydantic is used for data validation and settings management. It ensures that data structures are correct and consistent, reducing the risk of errors. Pydantic is particularly useful for validating input data and configurations in your AI application.

sqlalchemy

SQLAlchemy is a SQL toolkit and Object-Relational Mapping (ORM) library. It provides tools for database management, allowing you to interact with SQL databases using Python objects. SQLAlchemy simplifies tasks such as querying, updating, and managing database records, making it easier to handle persistent data in your application.

This project is below by IBMSkillsNetwork with some modifications : [Build a grounded Q/A Agent with LangChain, Granite and RAG (cognitiveclass.ai)]