 # Language Model-Based Question Answering with LLM

Introduction

This document outlines a Python script that utilizes a Language Model-Based Question Answering system with LLM (Large Language Models). The script employs transformers, language embeddings, and custom vector stores to answer questions based on a given context. The context is derived from PDF documents, and the question answering model is powered by a quantized LLM.

Objective

The primary objective of this code is to answer questions provided by users based on a context derived from PDF documents. It achieves this through the following steps:

1. **Installation**: Necessary Python libraries and packages are installed using `pip`. These include transformers, langchain, sentence_transformers, PyPDF, and other dependencies.

2. **PDF Text Extraction**: The code reads the text content of a PDF file (in this case, "Sheraton.pdf") and extracts it for further processing.

3. **Vector Store Creation**: The extracted text content is divided into chunks, and a vector store is created using Hugging Face embeddings. This vector store will be used for retrieving relevant documents based on user questions.

4. **Loading LLM Model**: An LLM model, specifically "vilsonrodrigues/falcon-7b-instruct-sharded," is loaded with quantization configuration. This model is responsible for generating answers based on provided questions.

5. **Question Answering Pipeline**: A pipeline is set up for text generation using the loaded LLM model. It takes a question, context, and generates answers based on the provided inputs.

6. **User Query Answering**: In the main function, a user question is defined ("How can I access the internet in my room?"). The code retrieves relevant documents from the vector store based on this question and context. It then constructs a prompt for the LLM model and generates an answer.

Libraries Used

The code relies on various Python libraries and frameworks:

- **transformers**: The Hugging Face Transformers library is used for loading and utilizing the LLM model.
- **langchain**: Langchain is used for document loading, text splitting, and vector store creation.
- **sentence_transformers**: This library is used for sentence embeddings.
- **PyPDF**: PyPDF2 is used for extracting text from PDF documents.
- **torch**: PyTorch is used for machine learning operations.

Steps Overview

Let's delve into each step in more detail:

Installation

- Required Python libraries and packages are installed using `pip`.

PDF Text Extraction

- The code reads the text content of a specified PDF file ("Sheraton.pdf") using PyPDF2.

Vector Store Creation

- The extracted PDF text is split into chunks and processed using Hugging Face embeddings. A vector store is created to store these chunks for retrieval.

Loading LLM Model

- An LLM model ("vilsonrodrigues/falcon-7b-instruct-sharded") is loaded with quantization configuration.

Question Answering Pipeline

- A pipeline is set up for text generation using the loaded LLM model. It takes a question and context and generates answers.

User Query Answering

- In the main function, a user question ("How can I access the internet in my room?") is provided. Relevant documents are retrieved from the vector store based on this question and context. A prompt is constructed for the LLM model, and an answer is generated.