INTEL UNNATI INDUSTRIAL TRAINING

FINAL REPORT

TOPIC - Running GenAI on Intel AI Laptops and Simple LLM Inference on CPU and fine-tuning of LLM Models using Intel® OpenVINO™

SUBMITTED BY - Ashutosh Jha
REGISTRATION NUMBER - 210907370
COLLEGE - MANIPAL INSTITUTE OF TECHNOLOGY (MAHE)

Email : ashutoshjha2015nov@gmail.com

PROBLEM STATEMENT

Running GenAI on Intel AI Laptops and Simple LLM Inference on CPU and fine-tuning of LLM Models using Intel® OpenVINO™ :

In this problem statement we must perform simple LLM inference on a CPU and understand the process of fine-tuning LLMs for custom applications. We also must become familiar with the basics of Generative AI and its applications. In the end we must produce a Custom Chatbot with Fine-tuned Pre-trained Large Language Model (LLM) using Intel AI Tools.

UNIQUE SOLUTION APPROACH

This project showcases an innovative approach to enhancing the capabilities of Large Language Models (LLMs) in answering questions related to a given Pdf document.

The project is divided into two main components: a preparation program and a question-and-answer (Q&A) program utilizing the LLM.

The preparation program is designed to process a PDF file, extracting its contents and creating a structured database known as a vector store. This vector store serves as a repository of information that the LLM can efficiently search through when answering questions.

The Q&A program leverages this vector store to provide precise and contextually relevant answers. When a user submits a query, the LLM model retrieves a pertinent subset of the document from the vector store. It then uses this subset to generate a well-informed and accurate response to the user's question.

This process is known as Retrieval Augmented Generation (RAG), which combines the strengths of retrieval-based and generative methods to enhance the accuracy and relevance of the answers provided by the LLM.

FEATURES OFFERED

1. **Document Preparation and Vector Store Creation** : The preparation program reads the entire content of a PDF document. It processes this content to create a vector store, a specialized database that allows for efficient similarity searching. Each fragment of the document is encoded into a high-dimensional vector, capturing the semantic meaning of the text.

2. **Efficient Retrieval Mechanism** : When a user poses a question, the LLM doesn't blindly generate an answer from scratch. Instead, it first uses the vector store to retrieve the most relevant fragments of the document. This retrieval process is powered by sophisticated algorithms that ensure the selected fragments are highly relevant to the query.

3. **Augmented Generation** : The retrieved document fragments are then used as a context for the LLM to generate an answer. This approach ensures that the generated answer is grounded in the actual content of the document, leading to higher accuracy and reliability. The LLM combines the information from the retrieved fragments with its own generative capabilities to produce a coherent and contextually appropriate response.

FEATURES OFFERED

4. **Enhanced Accuracy and Relevance** : The RAG technique bridges the gap between retrieval-based methods (which are precise but limited to existing content) and generative models (which can produce new content but might lack accuracy). By augmenting generation with retrieval, the system benefits from the best of both worlds, resulting in answers that are both relevant and informative.

5. **Scalability and Flexibility** : The vector store can handle large volumes of data, making this approach scalable to extensive documents or multiple documents. The system is flexible and can be adapted to various types of documents and queries, providing a robust solution for a wide range of Q&A applications.

PROCESS FLOW

This detailed process flow illustrates how each component of the project works together to provide a functional PDF-based Q&A chatbot using Retrieval-Augmented Generation (RAG) and optimized inference with OpenVINO.

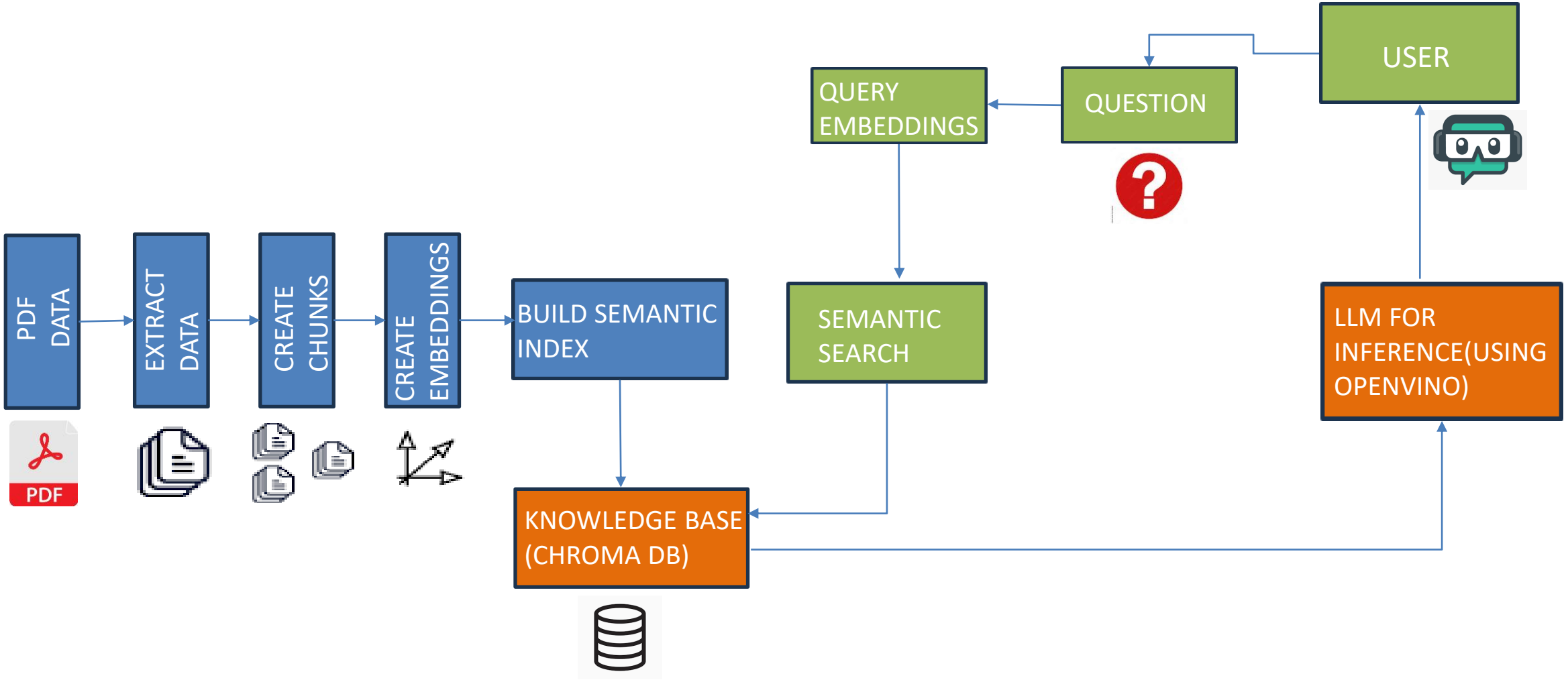
1. PDF Data Extraction and Processing:

Step 1: PDF files are read, and text is extracted page by page.
Step 2: The extracted text is split into smaller chunks for efficient processing.
Step 3: Embeddings for the text chunks are generated using the specified embeddings model.
Step 4: The generated embeddings are stored in a vector database (Chroma DB) to create a semantic index.

2. Query Handling and Response Generation:

Step 1: User interacts with the chatbot client (Streamlit app) and inputs a query.
Step 2: The query is sent to the FastAPI server.
Step 3: The server processes the query by generating embeddings for it.
Step 4: A semantic search is performed in the vector store to retrieve relevant text chunks.
Step 5: The language model generates a response based on the retrieved text chunks.
Step 6: The response is sent back to the chatbot client and displayed to the user.

ARCHITECTURE



TECHNOLOGIES USED

- **Python:** Main programming language.
- **FastAPI:** API server framework.
- **Streamlit:** Frontend framework.
- **Transformers (HuggingFace)**: A library providing general-purpose architectures for Natural Language Understanding (NLU) and Natural Language Generation (NLG).
- **Optimum.intel.openvino**: Model optimization for Intel hardware.
- **LangChain**: For managing language model chains and retrieval-augmented generation.
- **Chroma**: Vector database for storing and querying embeddings.
- **dotenv**: For environment variable management.
- **tqdm**: For progress bars.
- **Logging**: For monitoring and debugging.
- **CharacterTextSplitter**: From LangChain, to split large texts into smaller chunks for efficient processing and embedding generation.

CODE EXPLANATION

1. Environment Configuration (Environment File)

Role: Sets up the configuration parameters for the project, including logging level, model specifications, cache directories, server details, and model precision.

2. LLM Downloader Code

Role : Prepares and optimizes the language model by downloading it and converting it into various formats (FP16, INT8, INT4) suitable for inference using OpenVINO.

We are using TinyLlama model.(TinyLlama/TinyLlama-1.1B-Chat-v0.6)

Functions : prepare_model: Downloads and converts the model into specified formats.

main: Initiates the model preparation process.

Output : Optimized language model files in different precisions stored in directories.

CODE EXPLANATION

3. PDF Processing and Vector Store Generation Code

Role : Extracts text from PDF files, splits the text into manageable chunks, generates embeddings for the text chunks, and stores them in a vector database (Chroma DB).

Functions : read_pdf: Reads and extracts text from each page of the PDF.

split_text: Splits the extracted text into smaller chunks.

generate_vectorstore_from_documents: Generates embeddings for the text chunks and stores them in Chroma DB.

generate_vectorstore_from_pdf: Orchestrates the reading, splitting, and embedding generation process.

parse_arguments: Parses command-line arguments for the script.

main: Initiates the vector store generation process.

Output : A vector store containing embeddings of the text chunks from the PDF.

CODE EXPLANATION

4. Chatbot Server code

Role : Sets up a FastAPI server that handles user queries, retrieves relevant information from the vector store, and generates responses using the language model.

Functions : run_generation: Handles the query-answering process by utilizing the vector store and language model.

root: Endpoint for handling user queries and returning responses.

Components : Initializes embeddings and vector store then sets up the language model pipeline for inference then creates a RetrievalQA chain for querying the vector store.

Output : A FastAPI server running on specified URL and port, capable of handling user queries.

5. Chatbot Client Code

Role : Provides a Streamlit-based frontend interface for users to interact with the chatbot. It sends user queries to the FastAPI server and displays the responses.

Components : Custom CSS for styling. It also does session state management for storing chat history. It also performs User input handling and display of messages and Communication with the FastAPI server to get responses.

Output : A user-friendly web interface for interacting with the chatbot.

OUTPUT

This is the chatbot that I have made, on asking the question, firstly , it gives us the CONTEXT from the pdf and then answers the question.

I glanced over his shoulder, and it was true. Three tables away sat an intrepid woman in an old-fashioned felt hat and a purple scarf, eating without haste and staring at him. I recognised her right away. She had grown old and fat, but it was Frau Frieda, with the snake ring on her index finger.

This is the snippet of the pdf where we can find the answer

OUTPUT

how did he spend the morning ?

I AM USING THE FOLLOWING CONTEXT FROM THE PDF

away, and woke ten minutes later, as children do, when we least expected it. He appeared in the living room refreshed, and with t

stepped on Spanish soil for the first time since the Civil War, on a stopover during a long sea voyage to Valparaiso. He spent a morning

ruda to his sacred siesta, which he took in our house after solemn

r during a long sea voyage to Valparaiso. He spent a morning with us hunting big game in the second-hand bookstores, and at Porte

QUESTION: how did he spend the morning ? **ANSWER:** He spent the morning with us hunting big game in the second-hand bookstores.

Word count: 118, **Processing Time:** 7.5 sec, 1.6e+01 words/sec

Your input here.

Before the disaster in Havana, I had seen Frau Frieda in Barcelona in so unexpected and fortuitous a way that it seemed a mystery to me. It happened on the day Pablo Neruda stepped on Spanish soil for the first time since the Civil War, on a stopover during a long sea voyage to Valparaíso. He spent a morning with us hunting big game in the second-hand bookstores, and at Porter he bought an old, dried-out volume with a torn binding for which he paid what would have been his salary for two months at the consulate in Rangoon. He moved through the crowd like an invalid elephant, with a child's curiosity in the inner workings of each thing he saw, for the world appeared to him as an immense wind-up toy with which life invented itself.

This is the snippet of the pdf where we can find the answer

VIDEO(RUNNING THE CHATBOT)

LINK :

https://drive.google.com/file/d/15GW5Ufbc2SNGUnf90CT-Tq6Ss-vt_3Zm/view?usp=sharing

This link has a video of me running the chatbot from the scratch , I have also written the steps on GitHub for running the chatbot , but in case of any issue , please refer this video.

TEAM MEMBERS AND THEIR CONTRIBUTION

This project has been done INDIVIDUALLY by me.

NAME : ASHUTOSH JHA
REGISTRATION NUMBER : 210907370
COLLEGE : MANIPAL INSTITUTE OF TECHNOLOGY
EMAIL : ashutoshjha2015nov@gmail.com

Conclusion

The PDF-based Q&A Chatbot project successfully integrates advanced machine learning and natural language processing technologies to create an efficient system for retrieving and processing information from PDF documents. By utilizing text extraction, HuggingFace transformers for generating embeddings, and Chroma for vector storage, the project ensures high-quality semantic search capabilities. The deployment of Intel's OpenVINO toolkit for optimized model inference further enhances performance, making the system both fast and resource-efficient. With a user-friendly interface built using FastAPI and Streamlit, the chatbot provides real-time, accurate responses to user queries, demonstrating the project's potential in intelligent document processing and offering a robust foundation for future developments in the field.

References

- YouTube.
- OpenVINO Documentation

THANK YOU !