Qazi Saim

AI Engineer

Kazisahim121@gmail.com LinkedIn GitHub Contact

Professional Summary

AI Engineer Trainee passionate about building intelligence system, skilled in deep learning, NLP, and LLM application, with hands-on experience end to end AI solution using Tensorflow, PyTorch, LangChain.

Projects

Twitter Sentiment Classification

Tech Stack: TensorFlow, Keras, NLTK, Python, Streamlit

- Built an end-to-end sentiment classifier to predict comment sentiment.
- Pre-processed text data and converted it into embeddings.
- Designed a neural network using LSTM and GRU achieving 88% accuracy.
- Saved the trained model in .keras format.
- Developed a Streamlit app for real-time sentiment prediction.

SMART-MCQ GENERATOR

Tech Stack: AWS Bedrock, LangChain, Streamlit, PyPDF2, Pinecone

- Built a smart MCQ generator from PDFs.
- Extracted PDF text using PyPDF2 and stored it in Pinecone vector DB.
- Implemented similarity search to fetch relevant content.
- Used LangChain and LLAMA-3 for AI-generated MCQs.
- Created a Streamlit web app for user input and MCQ generation.

Health Care Bot (Fine-Tuning)

Tech Stack: DeepSeek-1.5b, WanDB, LangChain, unsloth

- Fine-tuned **DeepSeek-R1-Distill-Llama-8B** on a medical chain-of-thought dataset using **Unsloth & LoRA (PEFT)**.
- Designed **structured prompts** with <think> reasoning steps to improve explainability in clinical decision-making.
- Pre-processed and trained on **FreedomIntelligence medical reasoning dataset** with Hugging Face TRL's **SFTTrainer**.
- Tracked experiments and metrics with **Weights & Biases (wandb)** for reproducibility.

Document Summarizer (RAG)

Tech Stack: Gemma-2b-it, PyPDF2, pandas, PyTorch, flash-attention

- Built a **Retrieval-Augmented Generation (RAG) pipeline** combining document embeddings with a local LLM for contextual Q&A.
- Processed a **nutrition textbook PDF** using PyMuPDF, spaCy, and custom chunking for sentence-level granularity.
- Generated **dense vector embeddings** with **SentenceTransformers (all-mpnet-base-v2)** and performed **semantic similarity search** using PyTorch and cosine/dot-product scoring.
- Integrated **Gemma LLM (2B/7B)** with quantization for efficient local inference, enabling context-aware answer generation.
- Designed reusable functions (retrieve_relevant_resources, ask) for scalable **semantic search + LLM generation**, forming a complete end-to-end RAG application.

Technologies

Programming Language & Frameworks

Python, Streamlit

Machine Learning

Linear Regression, Logistic Regression, Scikit-learn

Deep Learning

Neural Network, ANN, RNN, LSTM, GRU, Transformers, PyTorch, Tensorflow, Keras

Generative AI

LangChain, Pydantic-AI, Ollama, RAG, CrewAI, LangGraph, wandb, pinecone, Aws, Aws Bedrock.

Education

Bachelors in Information Technology
Years - 2023

Certification

Full Stack Generative AI Engineer (PwSkills)