**Eng-Tel-Translator Project Documentation**

**Date:** July 24, 2025

**1. Eng-Tel (English to Telugu Neural Machine Translator)**

**Overview**

Implements a Neural Machine Translation (NMT) system using Facebook’s mBART-50 model to translate English text into Telugu. Includes model fine-tuning, evaluation, and a Streamlit UI for real-time translation.

**Requirements**

* Python packages: transformers, datasets, sentencepiece, torch, accelerate, streamlit
* Install via:

text

pip install -r requirements.txt

or manually as above.

**Process**

* **Dataset Preparation:**
  + Parallel English-Telugu sentences in eng-tel.txt and eng-tel.jsonl.
* **Model Fine-Tuning:**
  + Fine-tune facebook/mbart-large-50-many-to-many-mmt on the dataset.
  + Training code in eng-tel.ipynb.
  + Checkpoints and tokenizer saved in mbart-en-te-checkpoints/ and mbart-en-te-model/.
* **Streamlit UI:**
  + app.py provides a web interface for translation.
  + Run with: streamlit run app.py
  + Enter English text, get Telugu translation.

**2. Fine Tuning**

**Overview**

Provides a notebook for estimating GPU memory usage for different fine-tuning strategies (Full, LoRA, QLoRA) and demonstrates LoRA on a Hugging Face transformer model using the peft library.

**Requirements**

* Python packages: transformers, peft, bitsandbytes, accelerate
* Install via:

text

pip install transformers peft bitsandbytes accelerate

**Process**

* **VRAM Estimation:**
  + Notebook (fine-tuning.ipynb) contains logic to estimate memory usage for various strategies.
* **LoRA Setup Example:**
  + Demonstrates LoRA on a tiny GPT-2 model.
* **Parameter Statistics:**
  + Compares base vs. trainable layers.

**3. Ollama/rag\_mistral\_app (RAG Chatbot with Mistral 7B)**

**Overview**

A Retrieval-Augmented Generation (RAG) chatbot using LangChain, Mistral 7B (via Ollama), and Streamlit. Answers questions from a custom book dataset (yoursdocs.txt) using semantic search.

**Requirements**

* Python packages: streamlit, langchain, faiss-cpu, sentence-transformers, ollama
* Install via:

text

pip install -r requirements.txt

**Process**

* **Document Ingestion:**
  + Load and chunk documents from data/yoursdocs.txt.
  + Embeddings generated using SentenceTransformer.
  + Stored in FAISS vector database.
* **Chatbot UI:**
  + app.py (Streamlit) provides chat interface.
  + User asks questions; relevant document chunks retrieved and passed to Mistral 7B for answers.
  + Run with: streamlit run app.py

**4. RAG/RAG-Chatbot & Pavitr-10/RAG-Chatbot (RAG QnA & Summarization Chatbot)**

**Overview**

Streamlit-based chatbot for QnA and summarization using RAG architecture. Uses ChromaDB for document storage and retrieval, and OpenAI/Gemini API for LLM responses.

**Requirements**

* Python packages: streamlit, openai, tiktoken, chromadb, markitdown[all]
* Install via:

text

pip install -r requirements.txt

**Process**

* **Document Ingestion:**
  + Upload documents via UI (pages/ingest\_page.py).
  + Convert to text, summarize, chunk, and ingest into ChromaDB.
* **QnA Chatbot:**
  + Ask questions about ingested documents (pages/chatbot\_page.py).
  + Retrieves relevant context from ChromaDB and generates answers using LLM.
* **App Navigation:**
  + Main entry (main.py) provides navigation between ingestion and chatbot pages.
  + Run with: streamlit run main.py