🧠 OSOS AI Technical Test: Dr. X RAG System Report

🧑‍💻 Author: Hamood AL-Shabanoti

📌 Overview

This project delivers a **fully local NLP pipeline** to process and analyze a mysterious corpus of research left behind by Dr. X. The core of the system is a **Retrieval-Augmented Generation (RAG) Q&A framework**, augmented with:

- File parsing across multiple formats
- Token-aware sentence chunking
- Vector database storage with `ChromaDB`
- Semantic search using `nomic` embeddings
- Local LLaMA model for generation, translation, and summarization
- ROUGE-based summarization evaluation
- Translation into English or Arabic
- Token-per-second performance benchmarking

All models and databases run **entirely offline**, aligned with the challenge constraints. CUDA 11.8 was used for GPU acceleration.

🔧 Pipeline Architecture

```
        ┌──────────────┐
        |  MR.X Data   |
        |   Folder     |
        └──────────────┘
```

```
             |
             ▼
     📄 Document Extractor
             |
             ▼
     🔪 Sentence Chunker
             |
             ▼
   🧠 Nomic Embeddings
             |
             ▼
     👂 Vector DB (ChromaDB)
             |
   ┌──────────┬──────────┐
   ▼                  ▼
🔍 RAG Q&A        📄 Summarization
   |                  |
   ▼                  ▼
🧠 LLaMA 3 (8B)    📊 ROUGE Scores
             |
             ▼
        🌍 Translation
```

📁 Document Processing

- **Supported Formats**: `.pdf`, `.docx`, `.csv`, `.xls`, `.xlsx`, `.xlsm`
- **Library Used**: `PyMuPDF`, `docx2txt`, `pandas`
- **Table Handling**: Flattened into readable plain-text; layout not reconstructed
- **Metadata Recorded**:
  - `source` (filename)
  - `page` number
  - `chunk_number`

    - `type` (`text` or `table`)

🧩 Chunking Strategy

- **Tokenizer**: `cl100k_base` (via `tiktoken`)
- **Method**: Sentence-aware chunking (`nltk`) with token limit and overlap
- **Chunk Size**: Max 500 tokens, 50-token overlap
- **Bonus**: Supports alternative token-only chunking if needed

---

🧠 Embedding & Vector DB

- **Model**: `nomic-embed-text-v1.5` via `sentence-transformers`
- **DB**: `ChromaDB` (persistent, local)
- **Performance Logging**: Logs tokens/sec for each chunk embedded
- **UUIDs**: Each chunk stored with a unique ID for traceability

---

💬 RAG Q&A System

- **LLM**: `llama3-8B.gguf` using `llama.cpp`
- **Retrieval**: Top-K semantic search using query embedding
- **Generation**: LLaMA generates context-aware answers
- **Multi-Turn Memory**: Maintains Q/A history for follow-up support
- **Fallback Handling**: Gracefully responds to irrelevant or empty queries

---

🌍 Translation System

- **Auto Language Detection**: via `langdetect`
- **Target Languages**: English and Arabic
- **LLM-Powered Translation**: Primary via LLaMA prompt
- **Fallback**: HuggingFace transformers pipeline (offline)
- **Extra**: Optional grammar refinement using LLaMA

📄 Summarization + ROUGE Evaluation

- **Strategy**: Prompt-based summary generation (overview or insight)
- **LLM**: LLaMA used for generation
- **Evaluation**: ROUGE-1, ROUGE-2, ROUGE-L computed using `rouge_score`
- **Use Case**: Works on single chunks, full documents, or corpus-wide summaries

---

📊 Performance Metrics

Task	Tokens/sec (Avg)	Notes
Embedding	~7512–27876 tokens/sec	SentenceTransformer on GPU
LLaMA Generation	~250-1096 tokens/sec	8B model with GPU acceleration
Translation	~84-956 tokens/sec	Prompt + fallback supported
Summarization	~45–521 tokens/sec	

✨ Creative Features

- 🔍 **Table-aware Chunking**: Chunks flagged as `table` or `text` using heuristics
- 🧠 **Manual Q&A Scoring Tool**: For human evaluation of LLM responses
- 🧪 **Data Query Detection**: Tags questions referencing datasets/tables
- 🔁 **Model Reloading Tool**: Reloads LLaMA in-memory without kernel reset
- ✅ **Memory Reset Utility**: Clears conversational history for clean runs
- ⚛️ **Post-Translation Grammar Refiner**: Boosts fluency of translated text

---

🧪 Example Q&A

Q: What was Dr. X researching?
A:

> Dr. X explored interdisciplinary topics involving ancient knowledge (e.g., Giza pyramids), alchemy, and cognitive-behavioral science. His latest documents propose connections between symbolic psychology and advanced technological frameworks.

Translated (Arabic):

> في موضوعات متعددة التخصصات تشمل المعرفة القديمة (مثل أهرامات الجيزة)، والخيمياء، وعلوم X بحث الدكتور
> السلوك المعرفي.

🦾 Requirements

All dependencies are listed in `requirements.txt`. Key packages:

```
llama-cpp-python
sentence-transformers
```

```
chromadb
langdetect
torch
transformers
tiktoken
rouge-score
nltk
fitz (PyMuPDF)
docx2txt
pandas
```

🔐 Constraints & Compliance

- ✅ Fully offline and local models
- ✅ No computer vision or OCR used
- ✅ GPU-accelerated (CUDA 11.8)

📌 Conclusion

This project successfully builds a local, scalable, and intelligent NLP system capable of analyzing complex research archives with high precision. From semantic search and language generation to summarization and translation, the system delivers robust, multi-functional NLP tools while honoring offline-first constraints.

This RAG system not only supports investigation into Dr. X's work but also showcases the potential of low-latency, locally-deployed AI pipelines for enterprise document intelligence.
