Dostoor LLM Assistant

**Group Members:**

Tony Begemy

Ahmed Galal

Ali Mohamed Ali

Hala Sedki

**Use Case:**

As an Egyptian citizen, you need to know your dostoor to understand your laws and rights. Reading through a large PDF full of text could be fun for some people, but for most this is tedious. The objective is to optimize the GPT model to make a specialized LLM specifically in the dostoor so you can ask it any question you need. This model aims to assist lawyers, students, legal professionals, and the general public in understanding their dostoor.

**Data Collection:**

* Data was collected the official presidency government website: <https://www.presidency.eg/ar/%D9%85%D8%B5%D8%B1/%D8%A7%D9%84%D8%AF%D8%B3%D8%AA%D9%88%D8%B1/>
* Was collected by a mix of scrapping and manual collecting.
* The data is split into 7 parts “الباب الأول \ الباب الثاني \ etc.”
* الباب الثاني and الباب الخامس are split into further parts “الفصل”:
  + الباب الثاني is split into 3 “الفصل”
  + الباب الخامس is split into 11 “الفصل”
* Was collected by a mix of scrapping and manual collecting.
* Inside each باب follows a specific topic, with multiple “مادة” discussing different topics about the same general topic.

**Data Preprocessing:**

* In the collected data, some of the numbers were in English while some where in Arabic, we unified them to all be in Arabic
* As mentioned, لباب الثاني and الباب الخامس had multiple الفصل in them while the rest didn’t, so the column ‘fasl\_name’ had to be filled with ‘’ instead of NaNs

**RAG, Embedding, and Vector Database**

* Used Microsoft’s e5 embedding as it was researched in Dr. Ammar Mohamed’s, Dr. Hazem Abdelazim’s, and Dr. Mohamed Tharwat’s Research paper “Semantic Embeddings for Arabic Retrieval Augmented Generation (ARAG)” that it was the best embedding model for Arabic.
* Used Chroma Vector Database with the previously mentioned e5 embedding model
* The RAG chunks were separated for each مادة as a chunk

**Libraries Used:**

* Pandas, Numpy, Langchain, Chainlit, chromadb, sentence\_transformer

**How to use**:

Simply run the file in the terminal using chainlit run Chainlit\_app.py

This will open the chainlit UI where you can type the prompt you want.