QuantumLEX: AI Legal Engine Insights: - By ANKIT SHAW (ankit25@uw.edu)

Abstract

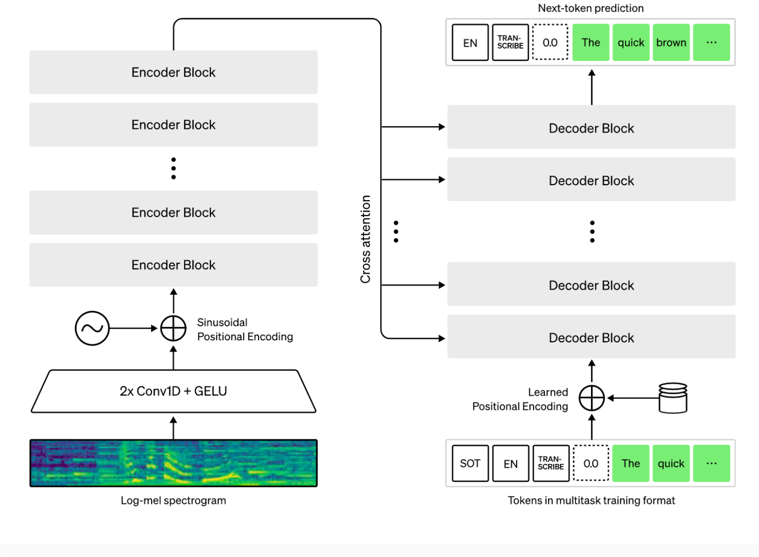
QuantumLEX is an innovative AI-powered Legal Engine designed to assist attorneys in analyzing cases through offline methods. Leveraging cutting-edge technologies such as Whisper for audio processing, Llava15 for image-related responses, and Mistral for text prompts, QuantumLEX offers a comprehensive solution for legal professionals. This technical report provides an overview of the technology stack, results achieved, and outlines potential avenues for future advancements.

Keywords

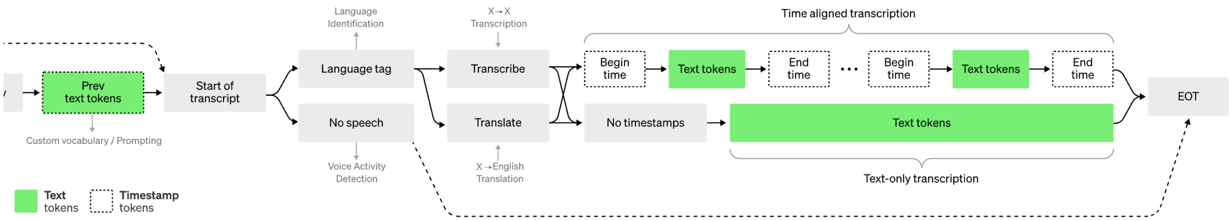
AI, Legal Engine, Whisper, Llava15, Mistral, Chatbot, Natural Language Processing, Image Processing, Legal Analysis

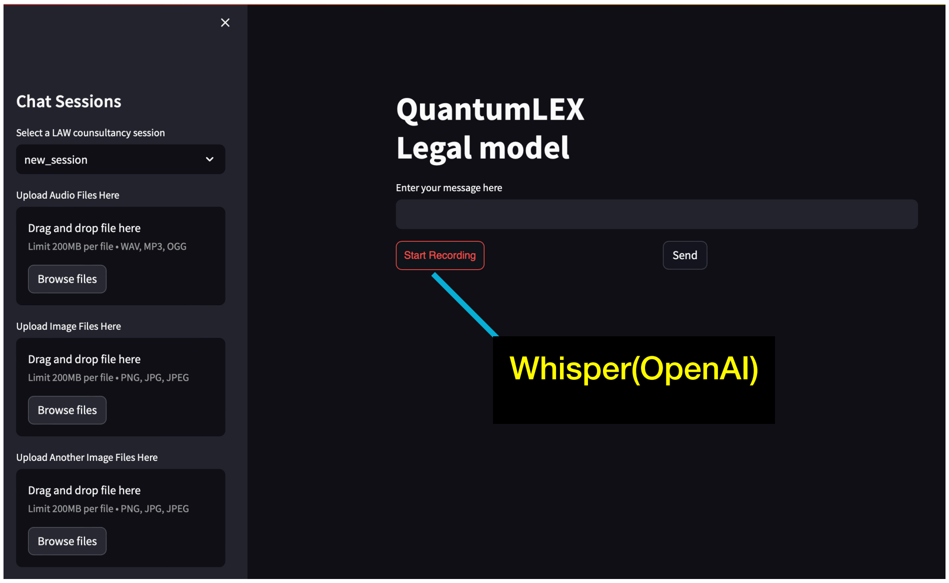
Technology Used

1. **Whisper-Automatic Speech Recognition (ASR) (https://github.com/openai/openai-whisper)**



Whisper, an automatic speech recognition system, is employed for processing audio instructions. Utilizing the open-source model "openai/whisper-small.en", QuantumLEX transcribes spoken words into text for further analysis.It is developed by OpenAI, is an ASR system designed for transcribing spoken language into written text. It plays a crucial role in QuantumLEX, enabling the system to understand and process audio instructions provided by legal professionals. The specific model used is "openai/whisper-small.en," which is optimized for English speech recognition.




×

Chat Sessions

Select a LAW counsultancy session

new_session

Upload Audio Files Here

Drag and drop file here

Limit 200MB per file • WAV, MP3, OGG

Browse files

Upload Image Files Here

Drag and drop file here

Limit 200MB per file • PNG, JPG, JPEG

Browse files

Upload Another Image Files Here

Drag and drop file here

Limit 200MB per file • PNG, JPG, JPEG

Browse files

QuantumLEX
Legal model

Enter your message here

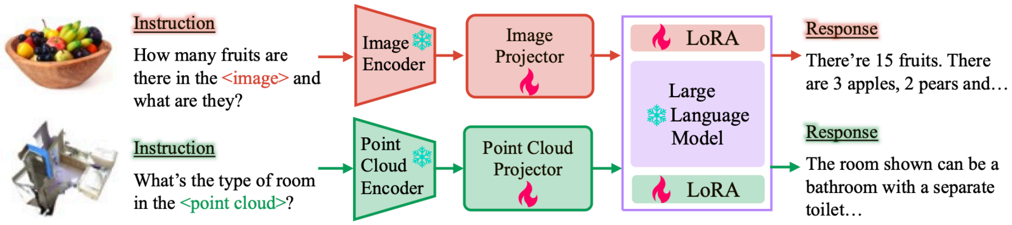
Start Recording

Send

Whisper(OpenAI)

The "Start Recording" Tab in the software is abled by the Whisper model by OpenAI, it transcribes the sentences of attorney into text, so that software can take verbal input as well, this is very useful for text segmentation of attorney during consultation session with client. AI will listen the conversation and interpret the legal sections and other case relevant data by conversation.

2. **Llava15-Image Description Generation (https://github.com/rob-romijnders/llava)**


Instruction
How many fruits are there in the <image> and what are they?
Instruction
What's the type of room in the <point cloud>?
Image Encoder
Point Cloud Encoder
Image Projector
Point Cloud Projector
LoRA
Large Language Model
LoRA
Response
There're 15 fruits. There are 3 apples, 2 pears and…
Response
The room shown can be a bathroom with a separate toilet…

Llava15, integrated into QuantumLEX as the image processing module, leverages the llava model for handling image-related responses. It works by interpreting user queries and generating detailed descriptions for uploaded images. Llava15 is a part of the Llava project, providing a framework for multimodal AI interactions. QuantumLEX utilizes Llava15 for generating detailed descriptions of uploaded images. The Llava project combines language models with computer vision to create comprehensive responses. The specific model paths used are "models\llava\mmproj-model-f16.gguf" and "models\llava\ggml-model-q5_k.gguf."

3. **Mistral**

Mistral serves as the text prompt engine, providing intelligent responses to legal queries. Its ability to comprehend complex language constructs enhances the natural interaction between the AI system and legal professionals.The Mistral model used here is instantiated using the [Transformers library](https://huggingface.co/transformers/) from Hugging Face. This library acts as a unified interface for various pre-trained language models, enabling easy integration and experimentation.

4. **LangChain - Language Processing Framework (https://github.com/M-Image-Analysis/langchain)**

LangChain, mentioned in the context of "QuantumLEX: An AI Legal Engine - Technical Report," is utilized as a language processing framework. It facilitates the handling of text-based interactions, memory management, and chaining different language models together. The exact functionalities and features provided by LangChain are not explicitly detailed in the provided snippets, but it plays a role in orchestrating the interaction between different components of the system.

5. **Chromadb - Vector Store (https://github.com/rob-romijnders/chromadb)**

Chromadb, a persistent client for [Chroma](https://github.com/rob-romijnders/chroma), is employed in QuantumLEX for storing vector embeddings associated with documents. It allows for efficient retrieval and comparison of document embeddings, supporting the vector search functionalities needed for legal case analysis.

6. **PyPDFium2 - PDF Text Extraction (https://github.com/zaabitt/pypdfium2)**

PyPDFium2 is used for extracting text from PDF documents. It enables QuantumLEX to process legal documents in PDF format, extracting relevant text for further analysis.

7. **Streamlit (https://github.com/streamlit/streamlit**)

Streamlit is employed as the user interface framework for QuantumLEX. It simplifies the creation of interactive and visually appealing web applications. QuantumLEX leverages Streamlit to provide a user-friendly interface for attorneys and legal professionals.

8. **PyTorch and Transformers (https://pytorch.org and https://huggingface.com/transformers/)**

PyTorch is a deep learning library that forms the foundation for various components in QuantumLEX. Transformers, a library built on top of PyTorch, provides pre-trained models and tools for working with transformer-based models, including language models used in Mistral.

Certainly! Below is a simplified workflow for the QuantumLEX: An AI Legal Engine based on the provided code snippets. Please note that this is a high-level representation, and the actual workflow may have more intricate details based on the internal workings of the libraries and models used.

QuantumLEX Workflow

1. **User Interaction**

   - Input:

     - User provides input through text messages, voice recordings, or image uploads.
     - Audio inputs are transcribed using the Whisper ASR system.

2. **Chat History and Session Management**

   - Functionality:

     - The system manages chat history, allowing for multi-turn conversations.
     - Different chat sessions can be selected and tracked.
     - Session state is maintained for proper context handling.

3. **Language Model Chain (LLM Chain)**

   - Components:

     - The `llm_chains.py` module orchestrates the LLM (Language Model) chain.
     - LLMs (Transformers models) process user inputs and generate responses.
     - LangChain is employed for memory management and interaction chaining.

4. **Audio Input Processing**

   - Components:

     - Audio inputs are transcribed using the Whisper ASR model in the `audio_handler.py` module.
     - Transcribed text is fed into the LLM chain for language understanding and response generation.

5. **Image Input Processing**

- Components:

  - Images are processed using the Llava15 image description generation model in the `image_handler.py` module.
  - Image content is embedded into the chat history for context.
  - Llava15 generates detailed image descriptions for user-uploaded images.

6. **PDF Text Extraction**

- Components:

  - PDF documents are processed using the PyPDFium2 library in the `pdf_handler.py` module.
  - Text content is extracted from PDFs and embedded into the chat history.
  - Extracted text is used for further analysis and responses.

7. **Conversation Memory and Prompts**

- Components:

  - The `memory_prompt_template` is employed for prompt creation and conversation memory.
  - Conversational context is stored using the LangChain framework for more coherent responses.

8. **Vector Database (ChromaDB) Integration**

- Components:

  - Vector embeddings associated with documents are stored and retrieved using ChromaDB.
  - ChromaDB integration in `llm_chains.py` enables efficient document comparison and retrieval.

9. **Streamlit UI**

- Functionality:

  - The Streamlit framework is used to create an interactive and user-friendly interface.
  - Users can input text, upload audio, images, and PDFs through the Streamlit UI.
  - Chat history and responses are displayed in the UI.

10. **User Interaction Loop**

- Flow:
  - User interactions trigger the appropriate processing modules (audio, image, PDF).
  - The LLM chain responds based on the combination of input modalities.
  - Responses are displayed in the Streamlit UI.
  - The conversation history is updated.

11. **Data Storage and Logging**

- Components:

  - Chat history and session data are stored in JSON format.
  - Logs of user interactions, responses, and processed data are maintained for analysis and future reference.

This workflow provides an overview of the various components and interactions within QuantumLEX, demonstrating how it processes user inputs, maintains conversation context, and generates relevant responses across different input modalities.

Future Advancements

The QuantumLEX project opens the door to several future advancements and improvements:

1. Enhanced Whisper Models: Continuously updating and fine-tuning Whisper models to improve transcription accuracy and support a wider range of accents.

2. Extended Llava15 Functionality: Expanding Llava15 capabilities to handle more diverse legal scenarios and refining image description generation for increased accuracy.

3. Advanced Mistral Models: Integrating more advanced Mistral models to enhance natural language understanding and provide more context-aware responses.

4. Interactive Legal Consultation: Implementing features for interactive legal consultation, allowing users to engage in a dynamic conversation with QuantumLEX, making the analysis process more user-friendly.

5. Security and Privacy Measures: Strengthening security measures to ensure confidential legal information remains protected, adhering to legal and ethical standards.

6. Continuous Training and Improvement Regularly updating the AI models with the latest legal precedents and case law to ensure QuantumLEX stays current and relevant.Apart from these using advanced GPU Hardware for fast processing

7. Fine tuning: Fine tuning and improving the models by using Lora and other such tool, apart from this creating an advanced dataset for analysing the case.

Problem Faced/Challenges

1) Hardware Limitations: Running the large commercial model on a local machine without GPU proved to be infeasible. This posed a significant barrier as deploying the project on cloud platforms would incur high costs due to the large model size. Additionally, processing inputs took an extended amount of time, sometimes up to 25-30 minutes to generate a response for a single line prompt.

2) Integration Complexity: Integrating multiple open-source AI models presented challenges, as it required careful consideration to avoid potential bugs and ensure smooth functioning. The complexity arose from the need to sequence the models correctly and handle potential conflicts between different components.

3) Testing : Testing the project was a time-consuming task due to limited hardware resources. The lengthy processing time, averaging 20-25 minutes per input, made it challenging to iterate quickly and identify errors. Even simple code errors required waiting 10-15 minutes for the output, significantly slowing down the debugging process.

4) Debugging Complexity: Debugging was challenging due to the project's architecture, where different components were divided across multiple files. This division, while intended to improve organisation, made it difficult to trace errors to their source. Even with the assistance of tools like ChatGPT, pinpointing errors often required focused human intervention and consumed a considerable amount of time.

5) Framework Compatibility: Some frameworks, such as LLM Studio, were not compatible with the local machine for testing of AI models, necessitating the use of alternative frameworks for testing. Switching between frameworks added another layer of complexity and contributed to the overall time-consuming nature of the project.

Conclusion

QuantumLEX, with its multi-modal AI approach, stands at the forefront of AI-powered legal analysis. The integration of Whisper, Llava15, and Mistral creates a powerful tool for legal professionals seeking efficient and insightful case analysis. As the legal tech landscape evolves, QuantumLEX remains poised for further advancements to meet the ever-growing demands of the legal industry.

Overall (Wrap up in one line)

QuantumLEX: An advanced legal insights engine leveraging state-of-the-art AI models for comprehensive case analysis and client interactions, revolutionising the legal industry with cutting-edge technology.

References:

https://arxiv.org/pdf/2306.06687.pdf

https://github.com/deep-diver/LLM-As-Chatbot

https://github.com/aniketmaurya/llm-inference

https://github.com/Abonia1/Context-Based-LLMChatbot

https://openai.com/research/whisper

https://github.com/openai/whisper