**🧾 Legal Document Summarizer Using Open-Source LLMs**

**📘 Executive Summary**

The Legal Document Summarizer is a Gradio-based web application that leverages cutting-edge open-source language models to automate the analysis of legal documents. The system supports PDF, DOCX, and TXT formats and extracts structured insights such as document summaries, legal glossaries, and AI-inferred verdicts. Additionally, it enables users to pose custom questions to the AI regarding the uploaded legal text.

**🎯 Objectives**

Automate legal document comprehension using LLMs

Simplify complex legal language for broader accessibility

Provide decision support through AI-predicted verdicts

Offer a lightweight, open-source, deployable solution for legal tech applications

**🛠️ System Architecture**

**📂 Components**

|  |  |
| --- | --- |
| **Module** | **Functionality** |
| Frontend (Gradio) | User interface for file upload, display, and interaction |
| Text Extraction | Uses pdfplumber and python-docx for parsing |
| Summarization | Powered by google/pegasus-xsum |
| Text Generation | Powered by MBZUAI/LaMini-T5-738M |
| Glossary Formatter | Converts raw glossary into HTML |
| Report Generator | Saves output to .txt for download |

**🔄 Workflow**

Document Upload

Text Extraction (via pdfplumber or python-docx)

**AI Analysis:**

Summary

Glossary (legal terms and definitions)

Verdict inference

Interactive Q&A

Export Report

**🧠 Models Used**

|  |  |
| --- | --- |
| **Task** | **Model** |
| Summarization | google/pegasus-xsum |
| Text Generation (Glossary, Verdict, Q&A) | MBZUAI/LaMini-T5-738M |

These models were selected for their strong performance in abstractive summarization and legal-style generation, while remaining lightweight enough for Hugging Face Spaces deployment.

**💻 Deployment Environment**

Platform: Hugging Face Spaces

Interface: Gradio 4.14

**Python Dependencies:**

transformers==4.40.1

torch

gradio==4.14.0

pdfplumber==0.10.3

python-docx==1.1.0

**📄 Features Overview**

|  |  |
| --- | --- |
| **Feature** | **Description** |
| ✅ Multi-format Upload | Accepts .pdf, .docx, .txt |
| ✅ Extracted Text Viewer | Displays full extracted content |
| ✅ Summary Generation | Key facts, issues, arguments, and outcomes |
| ✅ Glossary of Legal Terms | Definitions of all legal terms in document |
| ✅ Verdict Prediction | AI-generated possible verdict |
| ✅ Document-based Q&A | Answer custom questions from the user |
| ✅ Downloadable Report | Saves .txt summary with all results |

**🧪 Sample Use Case**

Document: Civil lawsuit

Input: PDF contract dispute filing

Output:

Summary: Key issues and case trajectory

Glossary: Definitions of legal jargon (e.g., "force majeure", "fiduciary duty")

Verdict: “The court is likely to rule in favor of the plaintiff due to breach of agreement…”

**📁 Project Structure**

├── app.py # Main Gradio app

├── requirements.txt # Python dependencies

├── legal\_doc\_llms\_demo.mp4 # [Optional] Demo video

└── README.md # Documentation

**🔒 Ethical & Legal Considerations**

No data storage: all processing is local to the session

Models may hallucinate; not intended for official legal advice

Designed for research, educational, and prototyping use

**📌 Future Improvements**

Add citation or source tracing for summaries

Use fine-tuned legal-domain LLMs (e.g., CaseLaw-BERT)

Extend document length capability with chunked analysis

Multilingual support for legal docs in non-English jurisdictions