# AI LEGAL ADIVSOR

### A MINI PROJECT REPORT

### 18CSC305J - ARTIFICIAL INTELLIGENCE

***Submitted by***

**RUCHI SHAH [RA2111026010387]**

**AKHILA S KUMAR [RA2111026010397]**

*Under the guidance of*

## DR. KARPAGAM M

Assistant Professor, Department of Computer Science and Engineering

***in partial fulfillment for the award of the degree of***

### BACHELOR OF TECHNOLOGY

in

## COMPUTER SCIENCE & ENGINEERING

of

### FACULTY OF ENGINEERING AND TECHNOLOGY



S.R.M. Nagar, Kattankulathur, Chengalpattu District

### MAY 2024

**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

(Under Section 3 of UGC Act, 1956)

## BONAFIDE CERTIFICATE

Certified that Mini project report titled **“AI LEGAL ADVISOR”** is the bona fide work of **RUCHI SHAH (RA2111026010387) & AKHILA S KUMAR (RA2111026010397)** who carried out the minor project under my supervision. Certified further, that to the best of my knowledge, the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

#### SIGNATURE

DR. KARPAGAM M

Assisstant Professor

Computational Intelligence

# ABSTRACT

This paper introduces "LLAMA," an AI Legal Advisor Chatbot empowered by an advanced Legal Language Model (LLM). The integration of artificial intelligence (AI) in the legal domain has seen significant advancements, but the challenge persists in creating tools that effectively assist legal professionals and laypersons in navigating complex legal landscapes. LLAMA aims to bridge this gap by utilizing state-of-the-art natural language processing (NLP) techniques, combined with legal expertise encoded in its LLM architecture.

LLAMA is designed to provide tailored legal advice, research assistance, and document analysis, catering to diverse legal needs across various jurisdictions and domains. Its LLM backbone enables it to comprehend legal texts, case law, statutes, and regulatory frameworks, allowing users to interact with it conversationally in natural language. LLAMA's capabilities extend to contract review, legal document drafting, legal research, and even predicting potential legal outcomes based on case specifics.

The development of LLAMA involves training on vast corpora of legal documents, precedents, and case law, fine-tuning its language understanding and legal reasoning abilities. Moreover, LLAMA integrates ethical considerations, ensuring compliance with legal standards, confidentiality, and impartiality in its recommendations and interactions.

The implementation of LLAMA represents a significant step towards democratizing access to legal expertise, particularly for individuals and small businesses with limited resources. By leveraging AI technologies, LLAMA has the potential to enhance the efficiency and accuracy of legal processes, alleviate the burden on legal professionals, and empower users to make more informed decisions in legal matters. However, challenges such as ensuring the reliability of AI-generated advice, addressing biases in training data, and maintaining user trust remain critical areas for ongoing research and development in the field of AI-powered legal assistance.

**TABLE OF CONTENTS**

[ABSTRACT 3](#_TOC_250002)

TABLE OF CONTENTS 4

[ABBREVIATIONS 5](#_TOC_250001)

1. INTRODUCTION 6
2. LITERATURE SURVEY 7
3. SYSTEM ARCHITECTURE AND DESIGN 8
4. METHODOLOGY 10
5. CODING AND TESTING 12
6. SREENSHOTS AND RESULTS 17
7. CONCLUSION AND FUTURE ENHANCEMENT 19

[REFERENCES 2](#_TOC_250000)0

**ABBREVIATIONS**

**LLM** Large Language Model

**NLP** Natural Language Processing

**MLP** Multi-Layer Perceptron

### INTRODUCTION

In an era characterized by rapid technological advancements, the legal profession stands on the brink of transformation through the integration of artificial intelligence (AI). With the exponential growth of legal information and the complexities inherent in navigating legal landscapes, there arises a pressing need for innovative solutions that can augment the capabilities of legal professionals and empower individuals with access to legal expertise.

This project report introduces "LLAMA," an AI Legal Advisor Chatbot equipped with a sophisticated Legal Language Model (LLM). LLAMA represents a pioneering effort in the realm of AI-powered legal assistance, aiming to revolutionize how legal information is accessed, interpreted, and applied. By harnessing the capabilities of natural language processing (NLP) and machine learning, LLAMA endeavors to provide tailored legal guidance, research support, and document analysis in a conversational manner.

The development of LLAMA stems from the recognition of the challenges faced by both legal practitioners and laypersons in grappling with the intricacies of the law. Traditional methods of legal research and consultation often prove time-consuming, costly, and inaccessible to many. LLAMA seeks to address these challenges by offering a user-friendly interface through which individuals can interact with a virtual legal advisor, receiving prompt and reliable assistance across various legal domains and jurisdictions.

At the heart of LLAMA lies its LLM architecture, meticulously trained on vast datasets comprising legal texts, case law, statutes, and regulatory frameworks. This foundation enables LLAMA to comprehend and analyze complex legal documents, extract relevant insights, and provide informed recommendations. Whether it's reviewing contracts, drafting legal documents, conducting legal research, or predicting potential outcomes in legal proceedings, LLAMA aims to streamline processes and enhance decision-making in legal matters.

Furthermore, the development of LLAMA incorporates ethical considerations to ensure adherence to legal standards, safeguard user privacy, and maintain impartiality in its interactions. While LLAMA holds immense potential in democratizing access to legal expertise and improving the efficiency of legal processes, it also raises important questions regarding the reliability of AI-generated advice, the mitigation of biases, and the preservation of user trust.

This project report delves into the design, development, and evaluation of LLAMA, shedding light on its capabilities, limitations, and future implications for the legal profession and society at large. Through a comprehensive analysis, we aim to contribute to the ongoing discourse surrounding the intersection of AI and law, paving the way for innovative solutions that bridge the gap between legal theory and practice

**LITERATURE SURVEY**

The landscape of AI-powered legal assistance has witnessed notable advancements in recent years, with various platforms striving to address the growing demand for efficient legal solutions. In this literature survey, we examine existing solutions such as ROSS Intelligence and LawGeex, highlighting their strengths, limitations, and the identified gaps that pave the way for the development of our project, LLAMA.

ROSS Intelligence represents one of the pioneering efforts in leveraging natural language processing (NLP) for legal applications. By parsing and understanding legal documents, ROSS aims to assist legal professionals in their research endeavors. However, while ROSS demonstrates proficiency in basic NLP tasks, it often falls short in achieving deeper contextual understanding, hindering its ability to provide nuanced legal advice across diverse scenarios.

Similarly, LawGeex offers automated contract review capabilities, relying on predefined rules to analyze and assess contracts. While this approach streamlines contract review processes to a certain extent, LawGeex's reliance on static rule sets limits its adaptability to new or unforeseen legal scenarios. Without the ability to learn and evolve in real-time, LawGeex struggles to keep pace with dynamic legal environments characterized by evolving regulations and case law.

The identified gaps in existing legal bots underscore the need for more advanced solutions capable of addressing complex legal reasoning and multi-jurisdictional laws. Most notably, existing platforms are constrained by the scope of their training data, often lacking the breadth and depth necessary to comprehensively understand and navigate diverse legal landscapes. Moreover, the absence of real-time learning capabilities impedes their ability to adapt to changing legal contexts and provide accurate, up-to-date advice.

Against this backdrop, our project, LLAMA, seeks to bridge these gaps by leveraging a sophisticated Legal Language Model (LLM) architecture. By harnessing state-of-the-art techniques in natural language processing and machine learning, LLAMA aims to offer a versatile and adaptive legal advisor capable of comprehensively understanding legal texts, conducting nuanced legal analysis, and providing tailored recommendations in real-time.

Through a comparative analysis of existing solutions and the identified gaps therein, this literature survey lays the groundwork for the development and evaluation of LLAMA. By addressing the shortcomings of current approaches and leveraging advances in AI technology, LLAMA aims to redefine the landscape of AI-powered legal assistance, empowering legal professionals and individuals alike with access to reliable, efficient, and contextually aware legal guidance

### SYSTEM ARCHITECTURE AND DESIGN

### System Architecture:

### Input: User queries in natural language.

### Processing: Tokenization and encoding of text to transform natural language into a format suitable for neural network processing.

### Normalization Layer:

### RMSNorm: Applies root mean square layer normalization to stabilize the variances of input features before they are processed by attention mechanisms. This helps in normalizing user inputs, especially in handling diverse legal terminology and phrasing.

### Attention Mechanism:

### Transformer-based Attention:

### Purpose: To focus on relevant parts of the input legal query to derive context and importance, enabling the model to better understand the specific legal nuances and user intent.

### Output: Contextualized representations of the user input that capture the key legal aspects and relationships in the text.

### Further Normalization:

### RMSNorm: Another layer of normalization post-attention to ensure the outputs are normalized before being passed to higher processing layers. This is crucial for maintaining numerical stability throughout the network.

### Multi-Layer Perceptron (MLP):

### Function: To interpret and process the attention-focused, normalized input into specific responses or legal advice.

### Layers: Consists of multiple dense layers with nonlinear activations to construct the final output from the processed inputs.

### Output: Generates responses that provide legal advice, information, or answers to the user's queries.

### Output Layer:

### Response Generation: Converts the MLP’s output into coherent, natural language responses.

### Function: Tailors the complex legal advice into user-friendly language, ensuring clarity and accessibility.

### Feedback Loop:

### User Feedback Collection: Allows users to provide feedback on the advice received.

### Adaptation: Uses feedback to adapt and refine response mechanisms and legal advice quality.

### Design:

### 

### METHODOLOGY

1. **Data Ingestion (ingest.py)**

* Purpose: Handles the initial loading, processing, and preparation of legal datasets.
* Process: This script is likely responsible for parsing raw legal documents, extracting relevant information, and possibly pre-processing text for the AI model. This could include cleaning the data, tokenization, and structuring the data in a way that is suitable for training or inference.

1. **Utility Functions (utils.py)**

* Purpose: Provides supporting functions that are used across different parts of the project.
* Details: This file might contain functions for various tasks like text normalization, error handling, and helper functions for data manipulation. These utilities ensure that repetitive tasks are modular and reusable throughout the project.

1. **Application Logic (app.py)**

* Purpose: Serves as the main entry point for the legal advisor application.
* Features:
  + User Interface: Manages interactions with users where they can input their legal queries.
  + Model Interaction: Handles the communication between the user inputs and the LLaMA model, ensuring that queries are processed and responses are generated accurately.
  + Response Handling: Formats the output from the LLaMA model into understandable legal advice and presents it back to the user.

1. **Dependencies (requirements.txt)**

* Purpose: Lists all Python libraries and their versions required to run the project.
* Key Components:
  + AI and ML Libraries: Includes libraries like torch, transformers, and sentence-transformers which are essential for running the LLaMA model.
  + Web Frameworks: Tools like streamlit or flask could be used for creating a web-based interface for the application.
  + Data Handling: Libraries such as pandas and numpy for data manipulation, and SQLAlchemy for database interactions.

1. **Integration and Testing**

* Integration: All components are integrated to ensure seamless data flow from user input through the model to the final advice presentation.
* Testing: Rigorous testing to ensure the system handles legal queries accurately and efficiently, including unit tests for individual components and integration tests for the whole system.

**Diagram:**

**CODING AND TESTING**

**Modules:**

1. Ingest.py

# Importing Dependencies

from langchain.text\_splitter import RecursiveCharacterTextSplitter

from langchain.document\_loaders import PyPDFLoader, DirectoryLoader

from langchain.embeddings import HuggingFaceEmbeddings

from langchain.vectorstores import FAISS

# Dataset Directory Path

DATASET = "dataset/"

# Faiss Index Path

FAISS\_INDEX = "vectorstore/"

# Create Vector Store and Index

def embed\_all():

"""

Embed all files in the dataset directory

"""

# Create the document loader

loader = DirectoryLoader(DATASET, glob="\*.pdf", loader\_cls=PyPDFLoader)

# Load the documents

documents = loader.load()

# Create the splitter

splitter = RecursiveCharacterTextSplitter(chunk\_size=800, chunk\_overlap=200)

# Split the documents into chunks

chunks = splitter.split\_documents(documents)

# Load the embeddings

embeddings = HuggingFaceEmbeddings()

# Create the vector store

vector\_store = FAISS.from\_documents(chunks, embeddings)

# Save the vector store

vector\_store.save\_local(FAISS\_INDEX)

if \_\_name\_\_ == "\_\_main\_\_":

embed\_all()

1. App.py

import streamlit as st

from utils import qa\_pipeline

chain = qa\_pipeline()

def main():

global chain

# Set the title of the web application

st.title('Indian Law Q&A Bot')

# Initialize the session state if it doesn't exist

if 'chat\_log' not in st.session\_state:

st.session\_state.chat\_log = []

# Get the user's question

user\_input = st.text\_input("You:")

# On user input, generate response and add to the chat log

if user\_input:

# Generate the answer

bot\_output = chain(user\_input)

bot\_output = bot\_output['result']

# Add the user input and bot output to the chat log

st.session\_state.chat\_log.append({"User": user\_input, "Bot": bot\_output})

# Clear the input box

st.text\_input("You:", value="", key="unique")

# Display the chat log

for exchange in st.session\_state.chat\_log:

st.markdown(f'\*\*You:\*\* {exchange["User"]}')

st.markdown(f'\*\*Bot:\*\* {exchange["Bot"]}')

if \_\_name\_\_ == "\_\_main\_\_":

main()

1. Utils.py

# Importing Dependencies

from transformers import AutoModelForCausalLM, AutoTokenizer, pipeline

from langchain import PromptTemplate, HuggingFacePipeline

from langchain.embeddings import HuggingFaceEmbeddings

from langchain.vectorstores import FAISS

from langchain.chains import RetrievalQA

# Faiss Index Path

FAISS\_INDEX = "vectorstore/"

# Custom prompt template

custom\_prompt\_template = """[INST] <<SYS>>

You are a trained bot to guide people about Indian Law. You will answer user's query with your knowledge and the context provided.

If a question does not make any sense, or is not factually coherent, explain why instead of answering something not correct. If you don't know the answer to a question, please don't share false information.

Do not say thank you and tell you are an AI Assistant and be open about everything.

<</SYS>>

Use the following pieces of context to answer the users question.

Context : {context}

Question : {question}

Answer : [/INST]

"""

# Return the custom prompt template

def set\_custom\_prompt\_template():

"""

Set the custom prompt template for the LLMChain

"""

prompt = PromptTemplate(template=custom\_prompt\_template, input\_variables=["context", "question"])

return prompt

# Return the LLM

def load\_llm():

"""

Load the LLM

"""

# Model ID

repo\_id = 'meta-llama/Llama-2-7b-chat-hf'

# Load the model

model = AutoModelForCausalLM.from\_pretrained(

repo\_id,

device\_map='auto',

load\_in\_4bit=True

)

# Load the tokenizer

tokenizer = AutoTokenizer.from\_pretrained(

repo\_id,

use\_fast=True

)

# Create pipeline

pipe = pipeline(

'text-generation',

model=model,

tokenizer=tokenizer,

max\_length=512

)

# Load the LLM

llm = HuggingFacePipeline(pipeline=pipe)

return llm

# Return the chain

def retrieval\_qa\_chain(llm, prompt, db):

"""

Create the Retrieval QA chain

"""

# Create the chain

qa\_chain = RetrievalQA.from\_chain\_type(

llm=llm,

chain\_type='stuff',

retriever=db.as\_retriever(search\_kwargs={'k': 2}),

return\_source\_documents=True,

chain\_type\_kwargs={'prompt': prompt}

)

return qa\_chain

# Return the chain

def qa\_pipeline():

"""

Create the QA pipeline

"""

# Load the HuggingFace embeddings

embeddings = HuggingFaceEmbeddings()

# Load the index

db = FAISS.load\_local("vectorstore/", embeddings)

# Load the LLM

llm = load\_llm()

# Set the custom prompt template

qa\_prompt = set\_custom\_prompt\_template()

# Create the retrieval QA chain

chain = retrieval\_qa\_chain(llm, qa\_prompt, db)

return chain

**TESTING:**

### SCREENSHOTS AND RESULTS

**CONCLUSION AND FUTURE ENHANCEMENTS**

In conclusion, the development of LLAMA represents a significant milestone in the evolution of AI-powered legal assistance. By leveraging advanced natural language processing techniques and a sophisticated Legal Language Model (LLM), LLAMA offers users a versatile and intelligent legal advisor capable of comprehensively understanding, analyzing, and providing guidance on a wide range of legal matters.

Throughout this project, we have demonstrated the feasibility and efficacy of LLAMA in addressing the shortcomings of existing legal assistance solutions. By bridging the gap between traditional legal research methods and modern AI technologies, LLAMA has the potential to revolutionize how legal expertise is accessed and utilized by legal professionals, businesses, and individuals alike.

Moving forward, LLAMA's development roadmap includes several key enhancements aimed at further enhancing its capabilities and effectiveness in providing legal assistance. One area of focus involves enhancing LLAMA's contextual understanding of legal language and nuances, enabling it to offer more accurate and relevant recommendations tailored to specific legal scenarios. Additionally, plans are underway to expand LLAMA's coverage to encompass a broader range of jurisdictions and legal systems, facilitating its adoption by users with diverse legal needs across different regions. Real-time learning mechanisms will be implemented to enable LLAMA to adapt and improve continuously based on user feedback and evolving legal landscapes. Integration with external legal databases and repositories will enrich LLAMA's knowledge base, providing users with access to a wider array of legal resources and information. Strengthening LLAMA's security measures and privacy controls remains a priority to ensure the protection of user data and compliance with legal and regulatory requirements. Advanced decision support capabilities, such as predictive analytics and risk assessment models, are also on the horizon to empower LLAMA in assisting users in making informed legal decisions. Continuous refinement of LLAMA's user interface will further enhance usability, accessibility, and overall user experience, contributing to its widespread adoption and acceptance as a leading AI-powered legal assistance platform. Through these future enhancements, LLAMA endeavors to solidify its position as a valuable tool for augmenting legal expertise and empowering users with reliable, efficient, and contextually aware legal guidance.

**REFERENCES**

1. https://blog.rossintelligence.com/
2. https://huggingface.co/meta-llama/Llama-2-7b-chat
3. https://www.lawgeex.com/
4. https://www.artificiallawyer.com/