# 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

The legal industry faces significant challenges in managing voluminous documentation and delivering timely, accurate legal advice. To address these challenges, we developed an AI-powered legal advisor chatbot, leveraging the LLaMA-7B model integrated with Langchain technology. This project incorporates advanced natural language processing (NLP) capabilities to interpret complex legal language and generate precise legal responses. Our system utilizes a combination of the Huggingface Transformers library for model management and Streamlit for creating an interactive user interface, which enhances user experience by providing a straightforward and responsive platform for legal inquiries.

The chatbot operates on a robust architecture where legal documents are vectorized and stored efficiently, allowing the model to retrieve relevant information rapidly. The system is structured in two distinct setups: a Normal Structure that processes inputs through attention mechanisms followed by normalization and multi-layer perceptrons (MLPs), and a Pre-normalization Structure that adjusts the order of operations to optimize training and performance.

Our methodology includes meticulous data ingestion processes to maintain data integrity and utility functions that ensure seamless data preprocessing and system operations. The outcome is a dynamic, reliable tool that not only elevates the standard of legal advice but also adapts and learns from user interactions to continually refine its advice quality.

This AI legal advisor represents a significant advancement in legal technology, promising to revolutionize how legal professionals manage their workload and interact with clients, thereby contributing to more informed and efficient legal decision-making processes.

**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 the rapidly evolving legal landscape, professionals continually seek innovative solutions to enhance efficiency and accuracy in legal service delivery. Recognizing the need for advanced technological assistance in the legal domain, our team embarked on an ambitious project to develop an AI-powered legal advisor chatbot. This cutting-edge system utilizes Meta's LLaMA-7B model, integrated with Langchain technology, to provide sophisticated natural language processing capabilities tailored to the legal industry.

The genesis of this project was motivated by the pressing challenges faced by legal professionals today, including the overwhelming burden of document management and the demand for quick, precise legal consultation. Traditional methods, which often rely on manual labor and rudimentary software tools, are increasingly inadequate in an era where speed and accuracy are paramount. Our solution leverages the transformative power of AI to not only streamline complex document analysis but also to interpret and respond to legal queries with unprecedented precision.

This report outlines the development process of the AI legal advisor chatbot, from conceptualization to implementation, including the detailed architecture and methodologies employed. It discusses the integration of state-of-the-art AI technologies—specifically the LLaMA-7B model and Langchain framework—highlighting how these tools enhance the chatbot’s capabilities. We will also delve into the system’s operational mechanics, featuring the innovative use of Transformer model structures such as Normal Structure and Pre-normalization to optimize performance and training efficacy.

Furthermore, this report evaluates the chatbot’s performance, offering insights into its ability to transform legal practice by augmenting decision-making processes and reducing operational burdens. By detailing the project's scope, challenges, and breakthroughs, this introduction sets the stage for a comprehensive exploration of a pioneering tool poised to redefine the legal advisory landscape.

**LITERATURE SURVEY**

The development of our AI-powered legal advisor chatbot is grounded in a comprehensive understanding of the interplay between artificial intelligence and legal technology. This literature survey encompasses a review of seminal works and recent advancements in both fields, shedding light on the evolving landscape of AI applications in legal contexts, as well as identifying technological gaps our project aims to bridge.

Historically, AI in legal tech has transitioned from basic document search tools to sophisticated systems capable of parsing and interpreting complex legal documents. Research papers and case studies, such as those examining ROSS Intelligence and LawGeex, highlight these tools' reliance on natural language processing (NLP) techniques. However, despite significant advancements, many current applications still struggle with deep contextual understanding and lack the ability to learn dynamically from real-time interactions.

In the domain of NLP, there has been a notable shift towards employing Transformer models, which are renowned for their ability to handle long-range dependencies within text, thanks to their attention mechanisms. Scholarly articles and technical reports from leading institutions like Google and OpenAI have documented how these models surpass traditional approaches in various NLP tasks. Additionally, innovations such as pre-normalization in Transformer architectures are being explored for their potential to enhance training stability and model performance, as noted in research disseminated by Meta AI on their LLaMA models.

The application of AI in creating legal advisor systems presents unique challenges, including the integration of extensive legal knowledge bases and ensuring the accuracy and trustworthiness of the advice provided. Industry analyses and expert commentaries have pointed out that while AI can significantly streamline processes like contract analysis and litigation prediction, there remains a critical need for systems that can adapt and respond to new legal scenarios without direct human intervention.

Furthermore, real-world case studies of AI deployment in legal settings suggest that while these technologies can drastically reduce the time required for tasks such as legal document analysis, they also raise issues concerning the balance between automation and human oversight. Comparative studies have begun to assess the reliability of AI systems against traditional legal services, indicating a gradual increase in the acceptance of AI tools for routine tasks, though complex decision-making still largely relies on experienced legal professionals.

This literature survey forms the bedrock of our project, providing key insights into the capabilities and limitations of existing AI systems in legal technology. By addressing these challenges, our AI-powered legal advisor chatbot aims to deliver a robust, intelligent, and user-friendly tool that enhances the efficiency and effectiveness of legal professionals.

### 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:

### figure 3

### 

### 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**

The development of our AI-powered legal advisor chatbot marks a significant step forward in the integration of advanced artificial intelligence within the legal industry. Leveraging the capabilities of Meta's LLaMA-7B model and the adaptability of Langchain technology, this project successfully demonstrates how sophisticated natural language processing can enhance legal services. The system effectively addresses key industry challenges, including the need for rapid, accurate legal advice and efficient document management. Our architecture, utilizing both Normal and Pre-normalization structures, ensures that the chatbot not only provides high-quality responses but also maintains a scalable and flexible operation suited to various legal contexts.

Throughout the project, the chatbot has shown potential in reducing the workload of legal professionals by automating routine tasks and providing preliminary legal consultations. The feedback from initial deployments has been overwhelmingly positive, highlighting the chatbot’s ability to deliver precise and contextually relevant legal advice. This success underscores the transformative impact of AI in law, promising to enhance decision-making processes and improve client interactions.

The AI-powered legal advisor chatbot is set for several strategic enhancements to increase its global accessibility and effectiveness. Key upgrades include multilingual support to serve diverse linguistic regions and the integration of a broader range of legal databases, which will enable the chatbot to address a wider array of legal questions with enhanced accuracy. Additionally, customization options will be developed to meet the specific needs of various law practices, such as commercial and environmental law, enhancing its utility in specialized legal areas.

Significant improvements are also planned for the chatbot’s learning algorithms, enabling it to learn dynamically from user interactions and continuously refine its responses to legal queries. Moreover, strengthening the chatbot’s security measures will be a priority, focusing on advanced encryption and robust data protection protocols to safeguard sensitive legal information and build user trust.

These future developments will position the chatbot as a cutting-edge tool in legal technology, offering more intelligent, rapid, and dependable legal advice while evolving to meet user needs more effectively.

**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/