## **Overview**

This document provides a comprehensive overview of the LawGPT model implementation, highlighting the main components, installation steps, and execution details. The model utilizes the MiniLM-L6-v2 architecture to deliver efficient legal document analysis with a constant response time of 2 seconds. This documentation also notes the limited datasets used in training and validation.

## **Table of Contents**

1. Introduction
2. Installation
3. Code Explanation
   * Step 1: Install Dependencies
   * Step 2: Import Libraries
   * Step 3: Mount Google Drive
   * Step 4: Load Model and Tokenizer
   * Step 5: Define Helper Functions
   * Step 6: Process PDF and Perform Inference
4. Performance Metrics
5. Limitations
6. Conclusion

## **1. Introduction**

The LawGPT model is designed for legal document classification, leveraging the MiniLM-L6-v2 architecture from the transformers library. Despite the limited dataset, the model achieves a consistent response time of 2 seconds, making it suitable for real-time applications.

## **2. Installation**

To set up the environment for running the LawGPT model, follow these steps:

### **Step 1: Install Dependencies**

python

Copy code

!pip install transformers PyPDF2

This command installs the necessary libraries:

* transformers: For the model and tokenizer.
* PyPDF2: For handling PDF files.

### **Step 2: Import Libraries**

python

Copy code

from transformers import AutoTokenizer, AutoModelForSequenceClassification, pipeline

from PyPDF2 import PdfReader

### **Step 3: Mount Google Drive**

python

Copy code

from google.colab import drive

drive.mount('/content/drive')

This allows access to datasets and models stored in Google Drive.

## **3. Code Explanation**

### **Step 1: Install Dependencies**

The required libraries are installed using pip:

* transformers: Provides tools for natural language processing.
* PyPDF2: Used for reading and extracting information from PDF files.

### **Step 2: Import Libraries**

The necessary modules from the installed libraries are imported:

* AutoTokenizer and AutoModelForSequenceClassification from transformers.
* PdfReader from PyPDF2.

### **Step 3: Mount Google Drive**

Mounting Google Drive facilitates access to files stored in the cloud, essential for loading datasets and saving results.

### **Step 4: Load Model and Tokenizer**

python

Copy code

tokenizer = AutoTokenizer.from\_pretrained("microsoft/MiniLM-L6-v2")

model = AutoModelForSequenceClassification.from\_pretrained("microsoft/MiniLM-L6-v2")

This step initializes the tokenizer and model from the MiniLM-L6-v2 pre-trained weights.

### **Step 5: Define Helper Functions**

Define functions to preprocess input data, perform inference, and post-process the results.

### **Step 6: Process PDF and Perform Inference**

Read a PDF file, extract text, and classify the content using the loaded model.

## **4. Performance Metrics**

The LawGPT model achieves a consistent response time of 2 seconds per query, making it efficient for real-time legal document analysis.

## **5. Limitations**

* **Datasets**: The model was trained and validated on a limited dataset, which may affect its generalizability.
* **Scope**: Designed primarily for legal document classification and may not perform well on other types of text.

## **6. Conclusion**

The LawGPT model, leveraging the MiniLM-L6-v2 architecture, provides efficient and consistent performance for legal document analysis. Future improvements can focus on expanding the dataset and enhancing the model's capabilities.