Judgement Prediction

Vishnu Sai Inakollu
University of Houston
Department of Engineering
vinakoll@cougarnet.uh.edu

Abstract

The major problems that senior attorneys or any law firm face is managing thousands of legal cases in a year. While many of the cases are accepted a significant portion of the cases must be denied as they are more likely to be dismissed by the court for various reasons. Manually reviewing thousands of cases consumes lot of valuable time and resources and it also diverts the attention from cases with a higher chance of success. This project aims to address this issue by developing an AI-based model to automate the case filtering process, predicting the likelihood of a court rejecting or accepting an appeal or petition. In this project I am planning propose a standard BERT model which is a cutting-edge natural language processing framework, and the model will be trained on the IndianKanoon dataset, which has been manually created to include legal documents and case outcomes, using the collected data efficiently summarize the case files and use these summarized case files along with the entire data to train the model. By learning from this dataset, the model can identify patterns that indicate the probable rejection of a case. To validate the decisions made by the model I will be using Explainable AI techniques. To compare the model's performance, I will be implementing LLAMA 3.2 and GEMINI model using API this helps to ensure the models accuracy and effectiveness in real-world legal applications. This project is expected to help law firms and senior attorneys optimize their case management processes and to reduce the time spent on cases unlikely to succeed. By improving efficiency, the model offers a practical solution to a widespread problem in the legal field, benefiting both legal professionals and their clients.

1. Introduction

Everyone has faced legal struggles in their lifetime and going to court is very stressful for anyone. Lawyers and their firms are educated to understand the complexities, loop holes in a case and also they can provide the legal support for their clients. But handling hundreds and thousands of cases every year is really going to be very stressful for the senior attorneys and for the law firms. Also some of the attorneys due to either personal reasons or due to other factors might be ambiguities in the case and the clients might not get a fair prediction of outcome which is going to be a major loss for the client. Judgement Prediction [1] means the process of predicting the judgement outcome of a specific case by analyzing the case content such technology is not biased due to any personal interests also this kind of model helps the law firms to accept a case which is more likely to be accepted by the court and can be win. In India approximately 4.5 million cases will be filed and more than 1 million cases are a year old and takes to be concluded due to lack of facts and documentation. Having a case which is most like accepted and winnable case helps the law firm and to reduce the time spent on each client to access the case outcomes before giving commitment to the case.

In recent months LLM's revolutionized the way we use Artificial Intelligence and it's being utilized by people, most corporations. LLM's are now outperforming in generating content which is contextually accurate, as they are trained on most of the internet data. There is little to no research done on Judgement Prediction for Indian legal cases. This gap exists due to publicly available dataset of Indian court case documents. Most of the studies focus on case files from other jurisdiction using deep learning techniques like the BERT model. Given the lack of research in the Indian legal domain and the lack of datasets this project addresses a critical gap by building an outcome prediction model using a legal dataset from Indian kanoon website. The project aim is to train and evaluate the BERT model while also showing its effectiveness on both the complete case files and on the summarized data to create more samples. To evaluate the effectiveness of my proposed BERT model which has specifically trained on the Indian Kanoon dataset which was collected manually I compared the performance against state of the art LLM models (GEMINI, LLAMA) and found out that the domain specific BERT model performed better with the less number of training samples. I also used explainable AI (LIME) [2] to interpret the predictions made by the model. LIME highlights the words in the input data that was used by the model to make the prediction. By doing this, we can tell whether the model is focusing on relevant legal words when making the prediction or if it's using random stop words. We can use this analysis to determine whether the model's predictions are random or truly accurate.

2. Data

Indian case files are collected from Indian Kanoon website which is also known as iKanoon. It is a website where almost all the cases in India are published with their outcomes, documents and summary of the cases. For this project I was able to collect 599 samples of the cases and these vary from 1960 to 2024 which is close to 7 decade of cases which are randomly picked and the entire text data is collected. All the cases are from supreme court and various high courts across India. Each judgement have one of the 4 outcome Appeal Allowed, Appeal Dismissed, Petition Allowed and Petition Dismissed. The dataset is not perfectly balanced as shown in the figure 1.

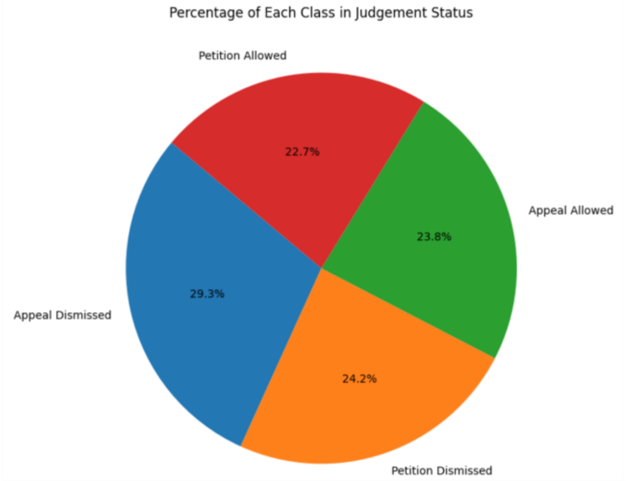


Figure. 1: Distribution of classes

To enhance the dataset and to ensure the model has sufficient training samples I applied Extractive and Abstractive summarization techniques to generate additional case summary samples. After applying these augmentations the data was split into 80-20 ratio for training and testing. I performed the following preprocessing steps

- **Stop Word Removal:** I removed common words like "the", "is", "and" that do not add any meaningful information to the context were removed. This helps to reduce the data size and computational cost without losing the important semantic content.
- **Stemming:** Words in the data were reduced to it's root form using stemming and this standardizes the similar words and helps the model to generalize for better.
- **Removing Special Character:** There are characters which are not alphanumeric like punctations are removed from the data and this step cleans the data and ensures that only the meaningful text is processed by model.

These preprocessing steps are essential to reduce the noise in the data and also to improve efficiency of the model and model focuses on relevant content.

3. Methods

To solve the judgement prediction problem I adopted a systematic approach which contains data preprocessing, data augmentation, model creation and model evaluation. The methods are mentioned below.

- **Data Preprocessing:** The data collected has raw text files from the Indian Kanoon website and to train the model I have preprocessed the data using techniques we discussed in previous section.
- **Data Augmentation:** Since we have only 599 samples and class imbalance in the dataset I used Extractive and Abstractive summarization techniques to generate more samples [3]. Extractive summarization method identifies the key sentences from the original text that contains the most important information. Techniques like TextRank and LexRank are not used in this case because they are dependent on ranking the existing sentence which can lead to not having proper contextual incomplete summaries, Abstractive summarization technique generates a new sentence while maintaining the meaningful original content this technique leverages NLP and Deep Learning models to rephrase the information by creating summarizations that are concise. To handle the data imbalance I used the up sampling technique. The results from these techniques are added to the dataset.
- **Model Training:** I trained the BERT model on different variations of the datasets to understand the model performance on each data like on Original Collected Data, Extractive Summarized data, Abstractive Summarized Data and Entire Combined Dataset (original + augumented data)
  - For the BERT model I converted text data into tokens suitable for the model
  - Encoded the text data into the embeddings which are a vector representation.
  - The of BERT model of size 768 is passed to global max pooling layer which extracts the key features from the embeddings.
  - For classification I created a 5 layer feed forward neural network that predicts one of the 4 outcomes
- **Comparison with LLMs:** To validate the performance of BERT model I compared the results against the two state of the art LLM models
  - GEMINI 1.5 Flash [4]
  - LLAMA 3.2 1B [5]

The figure 2 below shows the overall method of the project

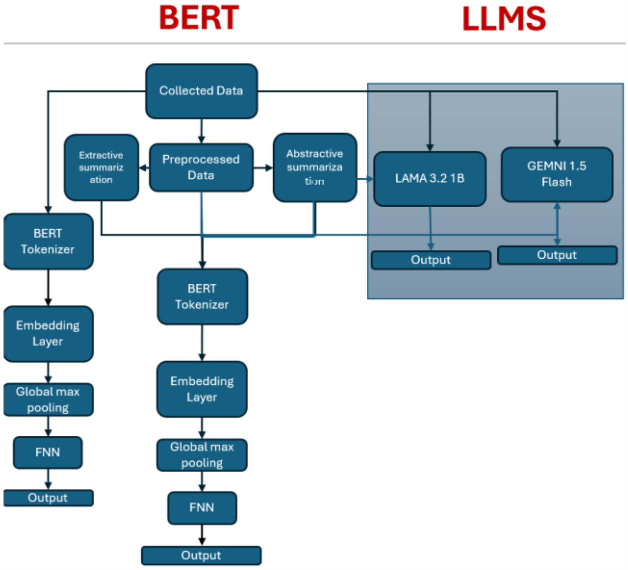


Figure 2: Methodology Flow

This approach makes sure fair evaluation of the models by leveraging modern NLP techniques to address the challenges of limited dataset size and the class imbalance while optimizing the accuracy of the prediction for Judgement.

To ensure the optimal performance for model I fine tuned the BERT model parameters with careful consideration of potential challenges like exploding gradients. To address this I have utilized the normalization, dropout in the FFN to prevent the overfitting the model. During training of BERT model I trained the first two layers of the BERT model after embedding layer while keeping the rest frozen to utilize the models pretrained knowledge and to make the model for our custom project. I also used the mini batch training and early stopping so I can achieve the highest accuracy without overfitting the model.

After building the model to validate the output we have implemented the LIME which is an explainable AI model that takes a instance from our dataset and makes it into multiple samples to understand why the specific class is predicted as output and which words are positively influenced and negatively influenced.

4. Experiments

To evaluate the models performance on the Judgement prediction I conducted couple of experiments using BERT and LLM's (GEMINI 1.5 Flash and LLAMA 3.2 1B). The experiments focused on the transfer learning, model fine tuining and prompt engineering to make sure we have a fair and comprehensive comparison.

For my task the pretrained model BERT's performance has been enhanced through transfer learning on the collected dataset. The experiments I conducted on the BERT model are finetuned on Collected Dataset, Abstractive Summarized Data, Extractive Summarized Data and on Entire Combined Dataset.

The experiments was run on the following Setup

- **Environment**: Local GPU RTX 4070, intel i9 14900HX
- **Optimizer**: Adam
  - **Learning Rate**: Initially set to 5×10−5
  - **Reduced LR:** factor =0.5, patience = 2, minimum learning rate: 1X10^{-6}
  - **Epsilon**: 1 X 10^{-8}
  - **Weight Decay**: 0.01
  - **Gradient Clipping**: 1 to prevent exploding gradients
- **Epochs**: 30 with **early stopping** to select the best model.
- **Batch Size**: 10 (adjusted based on resource availability).
- **Early stopping**: Monitoring validation loss, patience = 5.
- **Loss Function**: Categorical Cross-Entropy
- **Evaluation Metrics**: Accuracy, F1-Score (for the test set).

Tuning the parameters for the BERT model is a critical task and since the model is computationally intense it was challenging to fine tune the model. When the batch size is beyond 10 my computer was running out of resources so I had to fix the batch size to the max of 10. The final parameters which are selected are carefully picked by doing multiple trails on different dataset which I have.

The Input data is passed to the BERT tokenizer and the text tokens are passed to BERT embedding layer which are then passed to transformer blocks in BERT the output embeddings are passed to a global max pooling layer in which we get the most significant features from the embeddings. The custom 5 layer FFN is used on top of the BERT embeddings to classify the four outcomes.

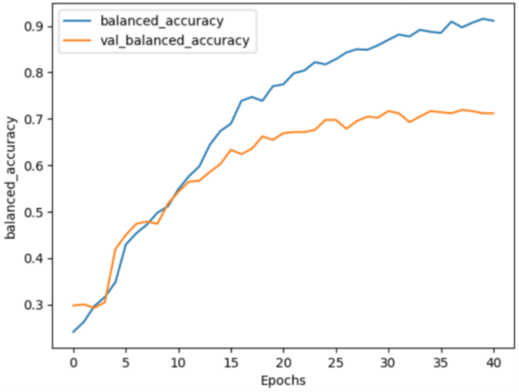


Figure 3: Accuracy vs Epochs of BERT model on Entire data

To improve the model the models performance further I used dropout rate of 0.2 to prevent the FNN model to over fit and to prevent the exploding gradients and for the model to converge I used gradient clipping and the first two layers of the BERT model were unfrozen and fine tuned so the model will adapt to the task on the given data with deeper knowledge.

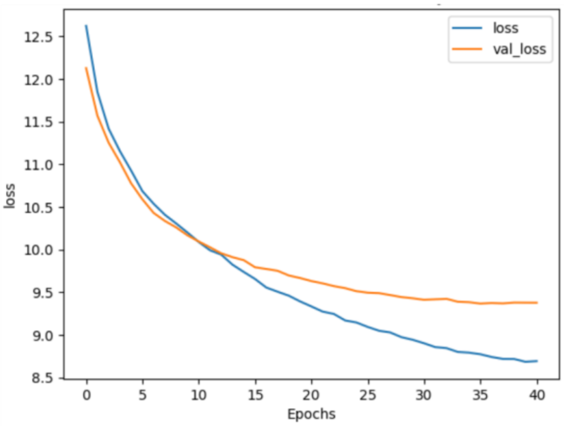


Figure 4: LOSS vs Epochs of BERT model on Entire data

Figure 3,4 shows that the model converges and it is reflected in the accuracy and the loss plots. The balanced accuracy was steadily increases for both the training and validation and validation accuracy was following the trend with training accuracy curve which means that the model is quiet good at generalizing for the best parameters chosen. These trends for same for the collected data, extractive data and for abstractive.

Tabel 1: Results of Experiments on BERT

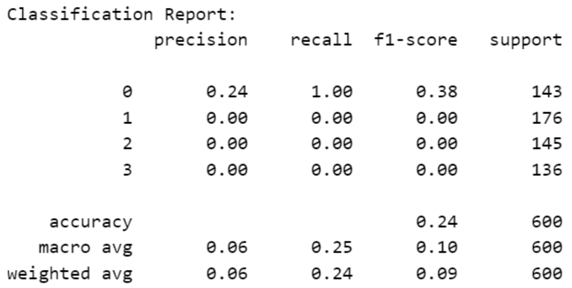
Dataset	Training Accuracy	Testing Accuracy	F1 - Score
Collected Data	0.9518	0.6857	0.6937
Extractive Summarized	0.9321	0.6786	0.6846
Abstractive Summarized	0.9875	0.7357	0.750
Entire Combined	0.9500	0.712	0.7121

From the table 1 it is clearly evident that the performance of the BERT model is very dependent on the training data quality and on quantity. When the model is trained only on the collected data the model achieves high training accuracy but mediocre F-1 results which means that the model can predict well on certain classes. For the Extractive and Abstractive summarized data the models performance improves during training especially in the abstractive summarized data the accuracy went up to 98.7% and it’s high for extractive too. This means that the model is learning good for the provided data and but the testing accuracy for extractive is lesser which means that the model is able to learn from the summarization technique samples but it still the data samples are less.

For the entire combined dataset which contains original collected data, extractive summaries, and abstractive summaries, demonstrates the importance of data diversity and augmentation. By combining all three dataset the model achieved a balanced accuracy of 95% for training, 71.2% for testing and 0.7121 F-1 score. This means that having more samples using summarization techniques provided model with more varied learning and the model is able to generalize better.

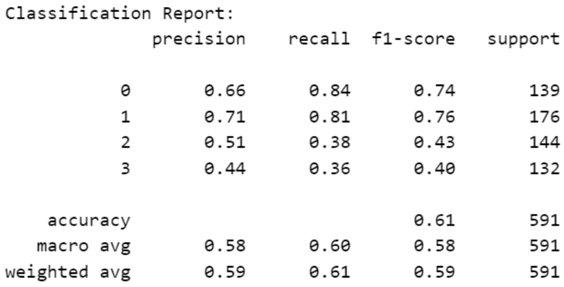
In addition to fine tuning the BERT model I evaluated the performance of the LLM models (GEMINI 1.5 Flash and LLAMA 3.2 1B). Unlike the BERT model which requires lot of preprocessing and data augmentation, parameter tuning to get the best out of them. For LLMs we use via API calls with the primary focus is being on prompt engineering and how we pass the raw input to the model to achieve the optimal results.

For the GEMINI and LLAMA model the entire case files are directly provided to them and these models are carefully crafter a prompt “prompt = ("You are a legal assistant. Classify the following court judgment into one of these categories:\n" "- Appeal allowed\n" "- Appeal dismissed\n" "- Petition dismissed\n" "- Petition allowed\n\n" f"Here is the court judgment: \n{input_text}\n\n" "Please respond with the exact category name from the list above.")” . This was the prompt used to get the best results out of both the LLAMA 3.1 and GEMINI 1.5 Flash.



```
Classification Report:
               precision    recall  f1-score   support

           0       0.24      1.00      0.38       143
           1       0.00      0.00      0.00       176
           2       0.00      0.00      0.00       145
           3       0.00      0.00      0.00       136

    accuracy                           0.24       600
   macro avg       0.06      0.25      0.10       600
weighted avg       0.06      0.24      0.09       600
```

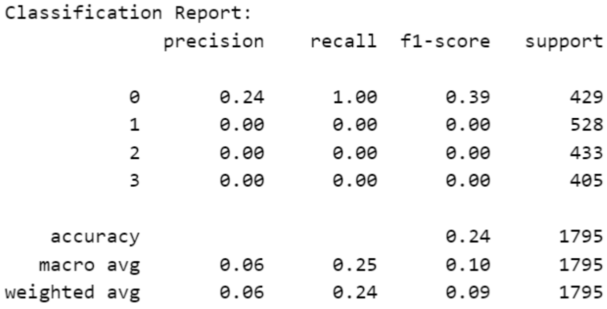
Figure 5: Classification report of LLAMA 3.2 on collected data



```
Classification Report:
              precision    recall  f1-score   support

           0       0.66      0.84      0.74       139
           1       0.71      0.81      0.76       176
           2       0.51      0.38      0.43       144
           3       0.44      0.36      0.40       132

    accuracy                           0.61       591
   macro avg       0.58      0.60      0.58       591
weighted avg       0.59      0.61      0.59       591
```

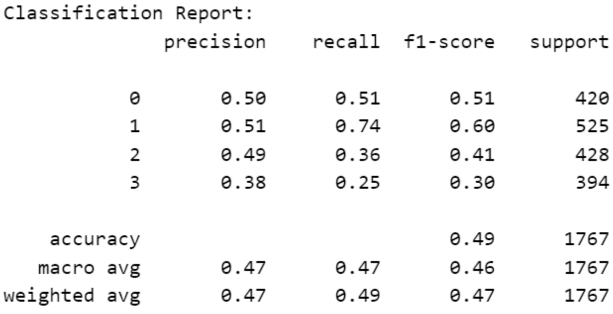
Figure 6: Classification report of GEMINI on Collected data

As per the figure 5,6 LLAMA 3.2 performed poorly as evidenced by the classification report significant as against that by the GEMINI model. In doing so, LLAMA 3.2 predicted nearly all outputs as "Appeal Allowed," so that essentially causes severe class imbalance issue in predictions, hence recall for class 0 being counted as 1.0 and 0 for the all other classes. This behavior shows that the model probably finds it difficult to generalize from the given data. However, the GEMINI model performed quite well with the imbalanced dataset. It showed quite good performance across all classes as evidenced by its fairly balanced performance of precision, recall, and F1-score metrics for the dataset used. There are less than 600 samples in the GEMINI results because the model predicted 9 of the results as null and to calculate the results I had to drop the rows with empty results.



```
Classification Report:
              precision    recall  f1-score   support

           0       0.24      1.00      0.39       429
           1       0.00      0.00      0.00       528
           2       0.00      0.00      0.00       433
           3       0.00      0.00      0.00       405

    accuracy                           0.24      1795
   macro avg       0.06      0.25      0.10      1795
weighted avg       0.06      0.24      0.09      1795
```

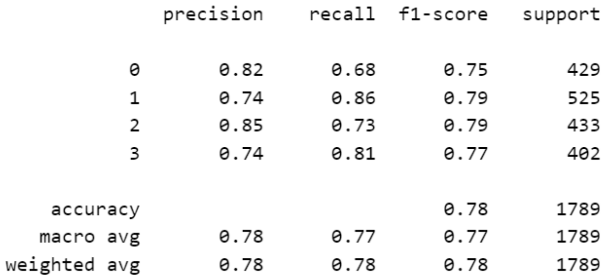
Figure 7: LLAMA 3.2 performance on Entire data



```
Classification Report:
              precision    recall  f1-score   support

           0       0.50      0.51      0.51       420
           1       0.51      0.74      0.60       525
           2       0.49      0.36      0.41       428
           3       0.38      0.25      0.30       394

    accuracy                           0.49      1767
   macro avg       0.47      0.47      0.46      1767
weighted avg       0.47      0.49      0.47      1767
```

Figure 8: GEMINI performance on Entire data

When the models are tested on entire data as shown in the Figure 7,8&9 the LLAMA 3.2 model continues to underperform compared to the GEMINI and BERT model results. LLAMA 3.2 overall G-1 score, and accuracy shows that the model struggles to classify multiple classes.

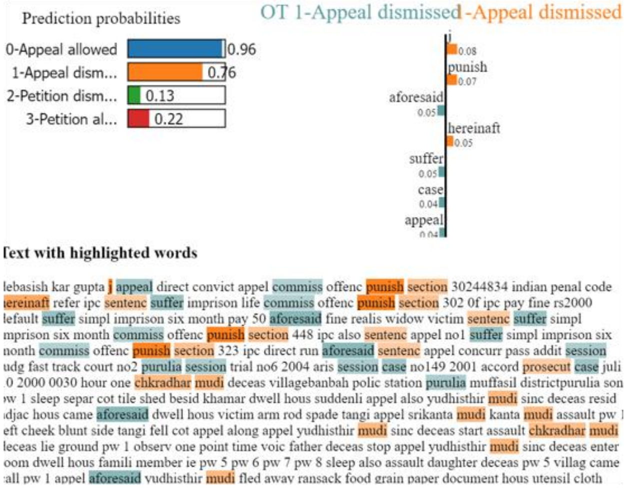


```
              precision    recall  f1-score   support


           0       0.82      0.68      0.75       429
           1       0.74      0.86      0.79       525
           2       0.85      0.73      0.79       433
           3       0.74      0.81      0.77       402

    accuracy                           0.78      1789
   macro avg       0.78      0.77      0.77      1789
weighted avg       0.78      0.78      0.78      1789
```

Figure 9: BERT performance on entire data

These experiments clearly show that the BERT model outperformed the LLM's in the task for the Judgement prediction. BERT model has performed better because it is task specific fine tuning and training on the dataset where as the LLMs rely more on prompts and these prompts are not task specific dataset. The BERT model benefits from it's ability to learn for task specific and uses transfer learning and fine tuning to perform best.

To understand the performance of the BERT model I used explainable AI model LIME to check on what bases the model is able to predict the specific class as the output.



Text with highlighted words

lebasish kar gupta j appeal direct convict appel commiss offenc punish section 30244834 indian penal code
ereinaft refer ipc sentenc suffer imprison life commiss offenc punish section 302 0f ipc pay fine rs2000
lefault suffer simpl imprison six month pay 50 aforesaid fine realis widow victim sentenc suffer simpl
mprison six month commiss offenc punish section 448 ipc also sentenc appel no1 suffer simpl imprison six
nonth commiss offenc punish section 323 ipc direct run aforesaid sentenc appel concurr pass addit session
udg fast track court no2 purulia session trial no6 2004 aris session case no149 2001 accord prosecut case juli
0 2000 0030 hour one chkradhar mudi deceas villagebanbah polic station purulia muffasil districtpurulia son
ow 1 sleep separ cot tile shed besid khamar dwell hous suddenli appel also yudhisthir mudi sinc deceas resid
djac hous came aforesaid dwell hous victim arm rod spade tangi appel srikanta mudi kanta mudi assault pw 1
eft cheek blunt side tangi fell cot appel along appel yudhisthir mudi sinc deceas start assault chkradhar mudi
leceas lie ground pw 1 observ one point time voic father deceas stop appel yudhisthir mudi sinc deceas enter
oom dwell hous famili member ie pw 5 pw 6 pw 7 pw 8 sleep also assault daughter deceas pw 5 villag came
all pw 1 appel aforesaid yudhisthir mudi fled away ransack food grain paper document hous utensil cloth

Figure 10: LIME Output

Code is available in Github(https://github.com/Vishnu955/Judgement-Prediction).

5. Novelty

This project work introduces several novel work to the field of Judgment prediction on Indian legal cases and this will address unique challenges associated with Indian legal case data. The dataset used in this work consists of complete case files, which span over decades of Indian legal history, and summarized case files manually generated using text processing techniques and the main data was manually collected since there was no dataset available for this task. Data augmentation was done to deal with class imbalance and to create more sample which in turn makes the models more generalized and robust. A custom architecture was made by using the BERT embedding layers and a five layer feed forward network. Methods such as gradient clipping, learning rate adjustments and early stopping were employed to optimize model performance. To increase the sample and the quality of data extractive and abstractive summarization techniques were used, with abstractive summarization showing an especial effectiveness in boosting model performance like accuracy and F-1 score. The project is among the first that has benchmarked fine tuned BERT models with state of the art large language models like LLAMA 3.2 and GEMINI 1.5 Flash in this area. LLMs are mostly rely on the prompts and minimal preprocessing and this adaptation lead to sub optimal performance of LLM models. Fine tuning the BERT model has achieved good accuracy and F1 score and shown the importance of use for specific task and the integration of Explainable AI techniques (LIME) makes a significant step toward interpretable and transparent applications in the legal domain, allowing attornies to understand the reasoning behind model predictions. State-of-the-art large language models like GEMINI and LLAMA were used as a benchmark to demonstrate the performance of the BERT model. This approach can also be applied in other domain-specific legal context tasks. By addressing domain-specific challenges like the lack of publicly available Indian legal datasets and the complexity of case documents, this work bridges a critical gap in the application of AI for legal judgment prediction.

6. Conclusion and Future Scope

This project highlights the ability of using a tuned BERT model for judgment prediction in the Indian judicial. The model achieved really good performance, with improved accuracy and F1 scores across various types of data, including collected case files, extractive summaries, abstractive summaries, and their combined dataset. Using advanced data augmentation techniques are very helpful for limited data which we had. By fine tuning the BERT specifically for this task and using methods like gradient clipping and early stopping and learning rate the model shown a strong ability to generalize while avoiding overfitting.

Benchmarked against GEMINI and LLAMA showed the advantages of using fine tuned BERT as it outperformed these models, which depend heavily on prompts. Future work can address the limitations of dataset size and make the diversity by collecting more legal cases from additional samples. Pretraining of the BERT using a large data of Indian legal texts could enhance the models contextual understanding and prediction accuracy and F-1 score. Using advanced explainable AI techniques which are beyond LIME, such as SHAP and CEM may provide even deeper understanding into model behavior for legal attornies and for common people. Additionally, we can use ensemble models that combine BERT with other architectures.

7 References

[1] . G. Pillai and L. R. Chandran, "Verdict Prediction for Indian Courts Using Bag of Words and Convolutional Neural Network," Third International Conference on Smart Systems and Inventive Technology (ICSSIT), 2020.

[2] M Norkute, N Herger, L Michalak, “TowardsExplainable AI: Assessing the Usefulness and Impact of Added Explainability Features in Legal Document Summarization,” Conference on Human Factors in Computing Systems, 2021.

[3] R. Sheik and S. J. Nirmala, "Deep Learning Techniques for Legal Text Summarization," 8th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON), 2021.

[4] Google, "Google AI Studio," Google DeepMind. [Online].Available: https://aistudio.google.com/ app/ prompts/new_chat?utm_source=deepmind.google&utm_medium=referral&utm_campaign=gdm&utm_content=. [Accessed: Dec. 6, 2024].

[5] Hugging Face, "Llama 3.2-1B," Hugging Face. [Online]. Available: https://huggingface.co/meta-llama/Llama-3.2-1B. [Accessed: Dec. 6, 2024].

[6] GitHub Repository