LegalEval: Court Judgement Prediction

Biplab Roy
22CS60R01

Sagnik Basu
22CS60R26

Sombit Bose
22CS60R31

Tamal Kanti Baksi
22CS60R60

Rahul Arvind Mool
22CS60R72

Abstract

Court Judgement prediction problem is a binary classification problem, given a legal judgement document, predicting the outcome of the case petition as accepted or rejected. We have aimed at designing an automated Natural Language Processing (NLP) system that could assist a judge in predicting the outcome of a court case.

We have trained different variations of baseline models like Bidirectional Encoder Representations from Transformer (BERT) (Devlin et al., 2018) and Robustly Optimized BERT (RoBERTa) (Liu et al., 2019) on a legal text dataset and reported the accuracy of the models using Selective Text Analysis, Segmented Text Analysis and a Sentence-level Text Analysis approach.

***Keywords*—** Court Judgement Prediction, Natural Language Processing, BERT, RoBERTa, Selective Text Analysis, Segmented Text Analysis, Sentence-level Text Analysis

1 Introduction

India is a highly populated country and suffers from the problem of high case pendency in various levels of judiciary. As on December 31, 2022, the total pending cases in district and subordinate courts was pegged at over 4.32 crore and over 69,000 cases are pending in the Supreme Court, while there is a backlog of more than 59 lakh cases in 25 high courts.[1] The total pending cases come to over 4.92 crore.

In these circumstances, AI-based legal applications can play a vital role in alleviating the stress on the judicial system by providing assistance of automated judgements and other legal services efficiently. Basic AI-blocks such as a Court Judgement predictor may prove to be immensely helpful in developing the above-mentioned applications for expediting the judicial process.

Apart from a court judgement predictor, other AI-based legal applications that can be developed include document review and analysis, contract review and analysis, legal research and analytics, and case management systems. These applications can help lawyers and judges with time-consuming tasks such as document review and research, allowing them to focus on more complex legal work.

The rest of the report is organised as follows. Section 2 discusses the dataset that we have used for this task. Section 3 lists all the different methods that we have applied for the judgement prediction task. Section 4 discusses about the results that we have got from our experiments and Section 5 concludes the report.

2 Dataset

The Indian Legal Documents Corpus (ILDC) is a publicly available dataset of legal documents from India. ILDC is a large corpus of 35k Indian Supreme Court cases annotated with original court decisions. It was created to facilitate research on natural language processing (NLP) and machine learning (ML) applications in the legal domain.

We have used ILDC$_{single}$ dataset for Court Judgement Prediction task. This dataset includes 7593 documents among which 5082 are for training, 994 are for validation and 1517 are for testing.

2.1 Dataset Description

The CSV file of ILDC$_{single}$ dataset, which has four columns ['text', 'label', 'split', 'name'].

- "text" contains the preprocessed data.
- "label" contains either '0' or '1'.
    - '0' represents all petitions have been rejected.

[1] Data collected from National Judicial Data Grid - https://main.sci.gov.in/

  - ‘1’ represents all petitions have been accepted.

- “split” maintains that the file belongs to either train set, validation set or test set.

- “name” shows the name of file.

3 Methodology

3.1 Data Preprocessing

The training data contains 3147 instances labeled as 0 and 1935 instances labeled as 1. This indicates an imbalance in the dataset, where the number of instances labeled as ‘0’ is significantly higher than the number of instances labeled as ‘1’.

To counter this imbalance during training, the model weights are adjusted such that the weight for instances labeled as ‘0’ is 1, and the weight for instances labeled as ‘1’ is 2. This means that the loss function during training will give twice as much importance to instances labeled as ‘1’ compared to instances labeled as ‘0’.

3.2 Selective Text Analysis Approach

In addition, to keep the input size manageable, only the first 500 tokens of each document are taken as input to the model, and the rest are discarded. This is done because the average token size of the documents is 3884, which is relatively large and can cause computational issues during training. By limiting the input to the first 500 tokens, the model can process the data efficiently and without memory issues.

For the given task we have used 4 pre-trained base models for training, validation, and testing. These models are as follows:

1. RoBERTa-base (Liu et al., 2019): A pre-trained model based on the Transformer architecture, which has been trained on a large corpus of text data.

2. BERT-base-uncased (Devlin et al., 2018): A pre-trained model based on the Transformer architecture, which has been trained on a large corpus of text data, and uses an uncased version of the input text.

3. legalBERT-base-uncased (Chalkidis et al., 2020): A pre-trained model based on the BERT architecture, which has been specifically trained on legal text data.

4. legal-RoBERTa-base (Geng et al., 2021): A pre-trained model based on the RoBERTa architecture, which has been specifically trained on legal text data.

3.3 Segmented Text Analysis Approach

A Segmented Text Analysis is a sliding window-based approach to capture the entire document as the model input. This approach involves dividing each document into several windows, each containing 512 tokens. These windows will slide by a stride of 64, which means that each subsequent window will overlap with the previous one by $512 - 64 = 448$ tokens.

The output of the model will be generated for each window, and these outputs will be averaged out to predict the final judgement. This approach has the advantage of capturing information from the entire document, rather than just the first 500 tokens as in the previous scenario. By sliding the windows with a stride of 64, the model can capture overlapping information from different parts of the document.

The average of the output from all the windows is taken to predict the judgement because it ensures that the predictions are not biased towards any particular window. By averaging the predictions from different windows, the model can capture a more comprehensive representation of the document and make more accurate predictions.

3.4 Sentence Level Text Analysis Approach

We have also experimented with a sentence-level approach using Hierarchical BERT Model (HBM) (Lu et al., 2021). HBM learns sentence-level features of the text and works well in scenarios with limited labelled data. Usually HBM gives high performance in long document classification tasks with only 50 to 200 labelled instances.

We have trained 3 Hierarchical BERT models on the dataset. One model is based on BERT-base-uncased, the other one is RoBERTa-base and the third one is DistilBERT. We added 2-layer Transformer encoder stacks on top of each pre-trained models. For each model these encoder stacks have the same configuration as that of BERT, RoBERTa and DistilBERT respectively. For all Hierarchical BERT models we have chosen 32 segments with max segment length of 16, which means for each document the model chooses 32 sentences with 16 words each for classification.

4 Result and Discussion

Each model (see Section 3.2) is trained for 3 to 5 epochs using selective text analysis approach, which is a common practice in deep learning to prevent overfitting and to achieve optimal model performance.

Model	Val Acc	Test Acc	Test Loss
RoBERTa-base	0.6464	0.5630	0.7286
BERT-base	0.6275	0.5280	0.8052
LegalBERT-base	0.6561	0.4997	0.7542
Legal-RoBERTa-base	0.6339	0.5511	0.7065

Table 1: Results for First 500 Tokens

Model	Val Acc	Test Acc	Test Loss
RoBERTa-base	0.6098	0.5412	0.8624
BERT-base	0.5777	0.5208	0.9969
LegalBERT-base	0.5682	0.5616	0.8761
Legal-RoBERTa-base	0.5995	0.5412	0.9848

Table 2: Results for Last 500 Tokens

BERT models accepts only 512 tokens at a time as input. Therefore we have used only first 500 or last 500 tokens from each document as input. The results of training the models with first 500 tokens and last 500 tokens are listed in Table 1 and Table 2 respectively. In a large document all of the tokens carry some meanings, so considering only first or last few tokens cannot give us the whole essence of the document. This issue may account for such poor performance of all these models.

As we have mentioned in Section 3.3, we have used a sliding window protocol of 512 tokens each and with a stride of 64 to take the whole document into account. Furthermore, we have taken an average of all results to ensure that the predictions are not biased towards any window. In this experiment we have got test accuracy of 0.5731 and test loss of 1.2539.

The Hierarchical Transformer models using BERT, RoBERTa and DistilBERT gave better results compared to the other models. Performance of these hierarchical models is shown in Table 3.

Model	Val Acc	Test Acc	Test Loss
RoBERTa-base	0.6693	0.6260	0.6611
BERT-base-uncased	0.6188	0.5513	0.7018
distilBERT	0.6666	0.5312	0.7466

Table 3: Results of Variations of HBM

5 Conclusion

In this report, we have analyzed the performance of different models in the legal domain on a large corpus of legal text (ILDC). The BERT or RoBERTa model can observe at most 512 or 1024 tokens, depending on the variation used, but for legal texts with a huge collection of texts, it is difficult to analyze the entire text at once.

We have mainly performed three approaches, comprising of selective text analysis, segmented text analysis and sentence-level text analysis. The first-mentioned is an analysis of prediction accuracy over a set of 500 tokens from the opening statement or preliminary remarks (first 500 tokens) and the concluding statement or the final remarks (last 500 tokens) of a court case petition. The RoBERTa-base model has generated comparatively better results. The second-mentioned is a sliding window-based approach (refer to section 3.3) using the RoBERTa-base model, and we have observed better result with a test accuracy of 0.5731. The third-mentioned is a sentence-level approach using a Hierarchical based Transformer model (refer to section 3.4) and the RoBERTa-base version of the HBM performs better than the other variations with a test accuracy of 0.6260.

However, it is important to note that AI-based legal applications cannot replace human judges and lawyers. Instead, they can assist them in their work, improving the efficiency and accuracy of the legal system. Additionally, it is crucial to ensure that any AI-based application used in the legal system adheres to ethical and legal standards and does not perpetuate biases or discrimination.

Overall, the development of AI-based legal applications has the potential to revolutionize the Indian judicial system and reduce the backlog of pending cases.

References

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. Legal-bert: The muppets straight out of law school. *arXiv preprint arXiv:2010.02559*.

Fernando A Correia, Alexandre AA Almeida, José Luiz Nunes, Kaline G Santos, Ivar A Hartmann, Felipe A Silva, and Hélio Lopes. 2022. Fine-grained legal entity annotation: A case study on the brazilian supreme court. *Information Processing & Management*, 59(1):102794.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Saibo Geng, Rémi Lebret, and Karl Aberer. 2021. Legal transformer models may not always help. *arXiv preprint arXiv:2109.06862*.

Prathamesh Kalamkar, Astha Agarwal, Aman Tiwari, Smita Gupta, Saurabh Karn, and Vivek Raghavan. 2022. Named entity recognition in indian court judgments. *arXiv preprint arXiv:2211.03442*.

Elena Leitner, Georg Rehm, and Julián Moreno-Schneider. 2020. A dataset of german legal documents for named entity recognition. *arXiv preprint arXiv:2003.13016*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Jinghui Lu, Maeve Henchion, Ivan Bacher, and Brian Mac Namee. 2021. A sentence-level hierarchical bert model for document classification with limited labelled data. In *Discovery Science: 24th International Conference, DS 2021, Halifax, NS, Canada, October 11–13, 2021, Proceedings 24*, pages 231–241. Springer.

Vijit Malik, Rishabh Sanjay, Shubham Kumar Nigam, Kripa Ghosh, Shouvik Kumar Guha, Arnab Bhattacharya, and Ashutosh Modi. 2021. Ildc for cjpe: Indian legal documents corpus for court judgment prediction and explanation. *arXiv preprint arXiv:2105.13562*.

Vasile Păiș, Maria Mitrofan, Carol Luca Gasan, Vlad Coneschi, and Alexandru Ianov. 2021. Named entity recognition in the romanian legal domain. In *Proceedings of the Natural Legal Language Processing Workshop 2021*, pages 9–18.