Transformer-Based Architecture for Judgment Prediction and Explanation in Legal Proceedings

Arooba Maqsood[1,2,3], Adnan Ul-Hasan[1], and Faisal Shafait[1,2]

[1] Deep Learning Laboratory, National Center of Artificial Intelligence (NCAI), National University of Sciences and Technology (NUST), Islamabad, Pakistan
[2] School of Electrical Engineering and Computer Science (SEECS), National University of Sciences and Technology (NUST), Islamabad, Pakistan
[3] School of Science, Edith Cowan University (ECU), Joondalup, Western Australia
{amaqsood.mscs20seecs,adnan.ulhassan,faisal.shafait}@seecs.edu.pk

Abstract. Advancements in language understanding have helped researchers develop a verdict prediction system that can assist a court judge in verdict formulation. This technological intervention can help streamline and standardize the decision-making process across all levels of courts. One key benefit of developing such a system is that the junior judges can benefit from the collective knowledge stored in the knowledge base, improving their ability to make consistent and well-informed decisions. For any such system to be practically useful, predictions should be explainable too. This research proposes a hierarchical pipeline that aims to leverage domain-specific variants of BERT to enhance the process of informed decision-making. The research is mainly divided into two modules: 'Legal Judgment Prediction (LJP)' and 'Legal Judgment Explanation Extraction (LJEE)'. The LJP task pertains to predicting the outcome of legal decisions concerning the appellant. In contrast, the LJEE refers to extracting out the phrases/clauses that led to the final decision. To promote research in developing such a system for Pakistani legal documents, this paper also introduces the VerdictVaultPK dataset. The dataset comprises around 11,943 rental-property case proceedings, each annotated with the court decisions indicating whether the appeal was allowed or dismissed. This research highlights how the use of domain-specific transformer models enriches semantic embeddings, contributing to a substantial accuracy improvement of 3-4%.

Keywords: Legal Judgment · Legal Explanation · Case Proceedings · Transformers · Legal-Transformers

1 Introduction

Making decisions on a legal issue demands reading numerous legal documents [8]. Extracting important information from legal text documents is a difficult and time-consuming task because of their distinctive characteristics, such as longer document sizes, a wide range of internal structure, and a complex pattern of relationships between documents [3]. To get a thorough judgment basis, judges and

professionals generally need to manually review a substantial amount of materials and legal documents. This technique requires a lot of time and labor [9]. It makes the task of outcome prediction difficult and makes it an open area for research. Researchers have recently begun to pay more attention to the field of anticipating court outcomes using machine learning and deep learning techniques. Various attempts in research have been made in recent years to predict judicial decisions using various machine-learning models. One drawback to machine learning-based approaches is that they are word-based approaches and do not capture the semantics of the text [1]. While the legal domain is highly context-sensitive [11]. For this research, we tend to explore domain-specific attention-based models to capture the semantics of these legal documents and hence provide a better outcome.

The main goal of this research is to propose a system that is capable of predicting unbiased judgments. In practice, it is seen that due to corruption, there will be a biased judgement [7]. Sometimes a particular judge can be inclined towards either dismissing or allowing an appeal, which again leads to biased judgments. The key advantage of our proposed system is that since we have not limited the data collection to any particular level of court, judge or year; the system is capable of capturing the decision-making style of all the judges (i.e. generalized context) from each case independent of any external biases.

This research is primarily focused on two aspects. First is the Legal Judgment Prediction (LJP) and the other is the Legal Judgment Explanation Extraction (LJEE). The LJP task refers to the prediction of the outcome of legal decisions (concerning the appellant). For any decision-making system to be practically useful, it needs to explain/give the reasoning for the predicted outcome [11]. Hence to address this, our second module is Legal Judgment Explanation Extraction which was proposed in [11] and refers to the task in which the aim is to explain the decision by extracting crucial phrases that lead to the decision, given the case proceeding and the predicted outcome. The model outputs the final verdict along with the phrases that had the most impact on the final verdict. The generic pipeline for this study is given in Figure 1.

The major contributions of this paper are as follows:

1. Creation of a new corpus of Pakistani legal proceedings, namely **VerdictVaultPK**, annotated with court decisions and explanations.
2. Use of **domain specific models** to leverage the task of legal judgment prediction as domain-specific lexicon used in court cases makes models pretrained on generally available texts ineffective on such documents.

This research provides comparisons to [11] in which the authors introduced the task of Court Judgment Prediction and Explanation (CJPE) task. They created two variants of the Indian Legal Document Corpus (ILDC) dataset: ILDC_single (contains files addressing single appeals only) and ILDC_multi (contains files addressing multiple appeals only). For the comparisons on our proposed model, we will be using the ILDC_single dataset [11] as it is similar to the dataset we created. This dataset will be referred to as the $ILDC_s$ dataset throughout this study. The key difference between the ILDC and our dataset is

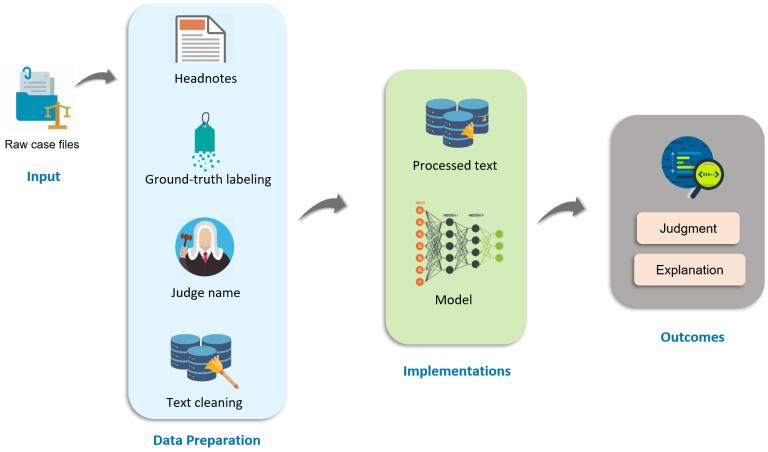

Fig. 1. The figure shows a high-level overview of the proposed pipeline. The raw case files are first fed into the data preparation module where the raw input casefiles are processed after the removal of headnotes, scanning for ground truth, removal of judge names and basic text preprocessing. After that, the processed case files are fed to the proposed hierarchical model for final prediction.

that ILDC includes appeals from various categories while our dataset focused specifically on rental property cases due to objectivity of the reasoning. We aim to further extend this dataset to contain cases from other categories of law as well.

The paper is mainly divided into 7 sections. Section 2 discusses the related work. Section 3 provides the overview of our proposed dataset. Section 4 discusses our proposed technique for the task at hand. Section 5 describes the experimental setup including preprocessing steps, hyper-parameter configurations, and our experimental design. Section 6 provides the findings and their interpretation and Section 7 concludes the study and provides future research directions.

2 Related Work

Several machine-learning approaches have been employed for text classification. In this section, we will summarize a few of them that have been used for legal text classification.

Medvedeva et al. in [4] investigated the use of NLP techniques to analyze court records. The authors presented extensive experimentation by using different sections of the case files as the input to the model. Feature extraction was done using tf-idf, n-grams, with stop words, without stop words, and the norm of frequency of word occurrences. The authors reported an accuracy of 75% on the ECtHR dataset [35]. The authors proposed to use the advanced machine learning techniques to further improve the accuracy. In [1], Liu et al. conducted experiments comparing five well-known machine learning models: k-NN, LR, Bagging,

RF, and SVM. The authors used dataset and experimental settings as stated in [4]. The results showed that the SVM model outperforms with an accuracy of 77.7%. Another study [2] used NLP techniques, particularly the bag-of-words model to represent the case text into n-grams. The best results obtained were 59% on the topic datasets using a random forest classifier. For this study, the dataset using the case records from the Philippine Supreme Court was created consisting of appeals from the Criminal Cases Category. In another study by [3], the outcome prediction is seen as a binary classification problem for classes 'Acquittal' and 'Conviction' of the accused person. The CART model outperformed all with an accuracy of 91.76%. To overcome the manual extraction of features, the authors proposed that extracting features from the text of judgment could be automated.

Strickson et al. [6] did extensive experimentation using SVM, LR, RF, K-NNs, perception, and MLPs using different word representations. The authors tested out the n-grams, topic clusters, and word embeddings. The authors were successful in producing decent results of 69.02 using the tf-idf features combined with the LR algorithm. In another study by [7], the authors used a CNN block for task at hand. The authors used the Bag of Words (BoW) to extract the keywords from the text. The proposed model gave an average accuracy of 85%. For this study, the publicly available data published by the Courts of India was used. This study was further improved by [8]. The authors employed Bi-GRUs with attention mechanisms to obtain better results. This approach gave the highest F1 score at 74.38%.

In [9,5,27,11] the authors focused more on the Transformer models. In [9], the authors used BERT proposed in [30] for feature extraction and used algorithms from deep learning based on Word2Vec [12] such as CNN, LSTM, DPCNN, and RCNN to predict judgment in judicial cases. The proposed approach was compared to the baselines for text classification in [13,14,15,17,18,19] and experimental results demonstrate that the deep learning model based on the BERT word embedding achieves 8%-10% more accuracy as compared to baseline. The dataset used was the 'CAIL2018' [21]. This idea of using BERT as a feature extractor was extended in [5] using the same dataset [21]. The authors proposed a fusion model based on BERT and LSTM-CNN for legal judgment prediction. The proposed model outperformed all the baselines with an F1-score of 96.97%. Another study by [27], proposed strong baselines that surpassed previous feature-based models in three tasks: (1) binary violation classification; (2) multi-label classification; (3) case importance prediction. The experimentation was done on 'ECHR Dataset'. Experimentally it was concluded that the hierarchical BERT outperformed with an F1-score of 82. The authors aimed to propose a better approach to explaining the outcomes of the model. Building on this limitation, the authors of [11] for the first proposed the 'Court Judgment Prediction and Explanation (CJPE)'.

In [11], the authors introduced ILDC dataset. The researchers experimented with a battery of baseline models for case predictions and proposed a hierarchical occlusion-based model for explainability. The best prediction model (XLNet

+ BiGRU) has an accuracy of 78%. As for the explanations module, the authors used the occlusion method [20] to extract the key chunks/sentences contributing to the model's final prediction. For this, they achieved 0.4445 for ROUGE-L. The findings highlight the significant discrepancy between a machine's explanation of a verdict and a legal expert's explanation. For future work, the authors proposed to train a legal transformer similar to LEGAL-BERT [28] on their Indian legal case documents. Similarly, another study by [33] attempted to investigate the impact of custom pre-trained models in the legal domain. They introduced three new variants of transformer models, namely InLegalBERT, InCaseLawBERT, and CustomInLawBERT that were retrained with a vocabulary based on Indian legal text. These models were then evaluated for primarily three tasks; Legal Statute Identification (LSI), Semantic Segmentation of Court Judgment Documents, and Court Appeal Judgment Prediction. Results reveal that the proposed variants of BERT marginally outperform the current state-of-the-art and show that there is promise in developing country-specific legal models [33].

The limitation of existing approaches is that either they are word frequency-based approaches or they use models pre-trained on a general corpus which may not yield good results in a legal setting. For instance, the tf-idf representation assumes that the counts of different words provide independent evidence of similarity. This approach lacks the capability of capturing the semantics of the text. Sequential architectures like GRUs are good at finding relationships between text sequences that are often over varying lengths of time-frames. But the GRUs do not perform equally well on larger sequences of text and are unable to capture long-term dependencies for large sequences due to vanishing/exploding gradients. In contrast to this, the transformer models employ the attention mechanism to learn the context of the larger sequences of text. This paper attempts to address these challenges and proposes to make use of the domain-specific models that are pre-trained on legal corpus as legal texts differ from the general text based on their structure and vocabulary. This study leverages the transfer learning principle. Our model incorporates a domain-specific backbone to enhance its performance and adaptability within the targeted domain as the key terms or words in the legal corpus have distinct meanings (the same term or phrase can have multiple meanings). It can be interpreted that changing the vocabulary here can mean that we are altering the semantics of the text.

3 The VerdictVaultPK Dataset

The biggest challenge of this research was the creation of a labelled dataset as we do not have a standard database for court cases. The Supreme Court of Pakistan is the highest court of Pakistan. Appeals are filed at this level. The reasoning for any case in any court of Pakistan can be of two types. One is 'subjective' and the other one is 'objective'. The subjective is mostly done for criminal cases where the courtroom environment and the gestures of the respondent/appellant can impact the case's final verdict. Unfortunately, all this information is not a part of the documented cases but plays a vital role in decision-making. On the contrary,

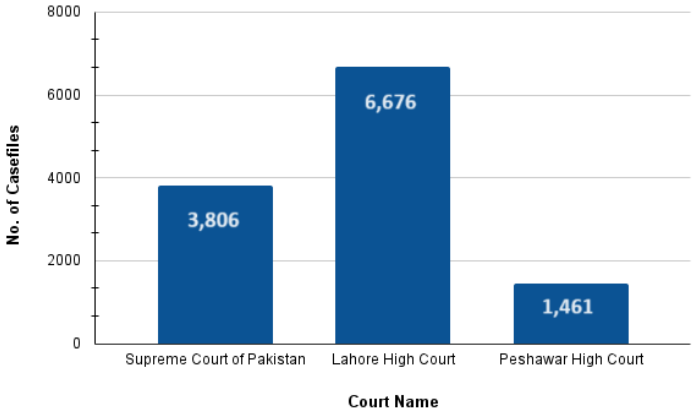

Fig. 2. Statistics of Cases per Court. From the Supreme Court to regional high courts, this graph unveils the distribution of cases filed from 1950 to 2021 in Pakistan's judicial landscape, totaling 11,943 files analyzed.

objective reasoning means giving a verdict based on facts and figures reported in the case. Civil appeals are a use-case of objective reasoning. For this research, as per the suggestion of legal experts, we have selected the 'rental-property' as they are the simplest kind of cases in the Civil Appeal (C.A.) category.

The case files were scrapped using the 'Beautiful Soup' library in '.PDF' format, later on converted to a text file for processing. Each file on average contained 5-6k tokens. The targeted courts included the Supreme Court of Pakistan, Lahore High Court, and Peshawar High Court for the years 1950 to 2021. The breakdown of several files per court is given in Figure 2. A total of 11,943 files were scrapped. The file duplicates and criminal-property-related cases were also removed. The case files were then annotated for the tasks at hand.

3.1 Legal Judgment Prediction Annotations

For data annotation, we held a series of meetings with representatives from the Supreme Court of Pakistan. Detailed discussions on what are the patterns found in case files for final decisions, led us to conclude that Legal Judgment Prediction (LJP) can be mapped as a binary classification task. This made the labeling of the dataset for the LJP an easier task. Mainly the final verdict of the case can be either 'allowed', which indicates a ruling in favor of the appellant/petitioner, or 'dismissed', which indicates a ruling in favor of the respondent. The case files were annotated by 'string matching technique', using the patterns provided by the legal team. The labeled files were then verified by the representatives for validation. Due to limited availability, we got validations from 3 experts (will be extended in future). Sample phrases that served as the basis for annotation are given in Table 1.

Table 1. Phrases for Data Annotation. (More than 600 such phrases were found in the case files and were used for data annotation.)

Sr.	Phrases for Allowed	Phrases for Dismissed
1.	appeal allowed	appeal dismissed
2.	appeal accepted	convictions set aside
3.	petition allowed	leave to appeal is refused

After the final annotation, the dataset is again scanned for the traces of these phrases. The phrases are then removed as they are the final required output from the model. The final dataset contains 11,362 cases labeled for the task of LJP. The data is also split into train, test, and validation sets using the stratification technique to maintain consistent class distribution across the subsets.

3.2 Legal Explanation Annotations

To measure the similarity between the predictions by the explanation module, we need some reference/gold annotations for the LJEE task too. It was pointed out by the legal experts that the most useful information is contained in the middle or towards the end of the case files. As one of the constraints in this research was the availability of the legal-domain experts, we also annotated a small portion of the test set (as suggested in [11]). For validation, the representatives were asked to read out the sample cases and mark the paragraphs that refer to reasoning sections in the case files which were cross-checked with our markings.

4 Methodology

This research aims to extend the work of [11] and provide a detailed review by experimenting with a variety of domain-specific transformers. Legal texts differ from the general text based on their structure and vocabulary. Applying general-purpose models like BERT to legal matters may be comparable to asking a liberal arts student to address a legal issue as opposed to a law student who has studied years' worth of legal material. This is problematic since there is a notable difference between the language used in legal documents and language used in broad open-source corpora, like Wikipedia and news articles. Legal documents frequently employ domain-specific terminologies and conventions that may not be commonly encountered in other types of text. For instance, legal terminology may include technical jargon, Latin phrases, or specialized terms that have precise legal definitions distinct from their everyday usage. Additionally, legal documents often rely on specific syntax and formatting conventions to convey legal concepts and arguments effectively. The domain-specific lexicon used in court cases makes the general-purpose models ineffective on legal documents [11]. In this paper, we have experimented with various domain-specific models, further finetuning them on custom data, to better learn the semantics

of the text due to the presence of domain-specific lexicon in legal corpora. We expected that this methodology could lead to a significant improvement in model accuracy, typically ranging between 4% and 5%.

For the legal judgment prediction, the documents need to be processed as a whole. One key challenge with the legal documents in the VerdictVaultPK dataset is that they are long and noisy. The file size normally varies from 400 to 5000+ tokens. Transformer models also have some implications when it comes to max sequence length accepted by the model as the dot product attention in transformers has a complexity of $O(n^2)$ where n is the sequence length. This computation becomes infeasible for large sequences. Transformer models like BERT can only process 512 tokens per example. To overcome this issue, we used hierarchical transformers [11]. The intuition is to process the case files as a series of chunks of length equal to 480 with overlapping windows of size n, where $n \in [70, 100, 200]$. We have selected 480 tokens for chunking the document as the BERT's tokenizer also performs WordPiece tokenization. For each chunk, we get an embedding from the transformer part of the model (as shown in Figure. 3). An important thing to note here is that there is no inter-chunk dependency at this point. To capture the inter-chunk dependencies, the embeddings against each chunk are fed into a standard Bi-GRU unit, followed by a dense layer containing a single unit for binary classification. As compared to [11]'s model, we have only used a single B-GRU layer to avoid overfitting. This hierarchical model is first trained for the task of judgment prediction and then is used to extract the phrases/chunks that lead to the decision-making.

For the explanation extraction, for each document, we mask each complete chunk embedding one at a time. The masked input is passed through the trained Bi-GRU and the output probability (masked probability) of the label obtained by the original unmasked model is calculated. The masked probability is compared with the unmasked probability to calculate the chunk explainability score as suggested by [11]. Formally, for a chunk c, if the sigmoid outputs (of the Bi-GRU) are α_m (when the chunk was not masked) and $\alpha_m^{'}$ (when the chunk was masked) and the predicted label is y, then the chunk scores given as $s_c = p_m - p_m^{'}$. The final explanations are obtained by tracing the top 3 chunks from the transformer part of the models. For final evaluations, we compare the performance of occlusion method explanations with the ground-truth explanations by measuring the overlap between the two.

We conducted a comprehensive comparison of our findings with those presented in [11] as this research served as a baseline for our work.

5 Experiments

This section describes the data preprocessing steps, our experimental setup, and details of the hyper-parameters that have been used for the experimentation.

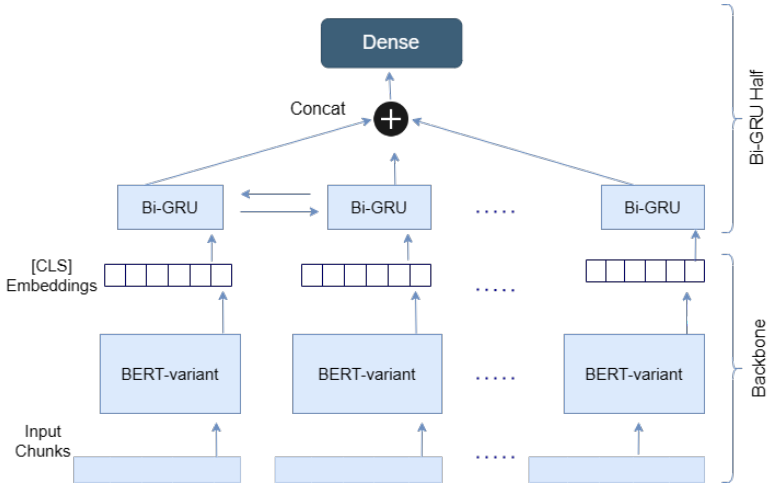

Fig. 3. Hierarchical Model Pipeline. The model is first trained for the task of legal judgment prediction using the binary labels and then the same model is used to extract the important chunks from the case files that led to final decision-making. The given pipeline is tested for multiple variants of the BERT. For each experiment, the backbone part of the pipeline is replaced keeping the BiGRU layer intact.

5.1 Data Preprocessing

The legal text requires some additional pre-processing. For instance, the judge's name is an important predictor while predicting the decisions on the cases [4]. This makes the anonymization of the judge's names in each case an essential step. Legal experts pointed out that a judge's identity can sometimes be a strong indicator of the case outcome [4,11]. Additionally, the case files have a specific format (as shown in Figure. 4) and the legal experts also recommended removing this meta-data as this information (i.e. headnote section) contained in the case file can also influence the final decision [11]. To avoid any bias being introduced into the dataset, we have removed the 'head-notes' from the case files. Other than this additional preprocessing, conventional NLP data preprocessing techniques like the removal of special characters, URLs, and white spaces were also applied.

5.2 Experimental Setup

In order to carry out the legal text classification, different experiments were carried out to provide a comparison with the baselines and the current state-of-the-art. In our exploration of text classification models, we tested out the baseline models including both classical machine learning models like Support Vector Machine (SVM), Logistic Regression (LR), Random Forest (RF), Naive Bayes (BN) and Decision Trees (DT) using the tf-idf word embeddings, as well as state-of-the-art transformer models such as BERT and it's domain specific variants. Given the token limitations inherent in transformer models, we devised

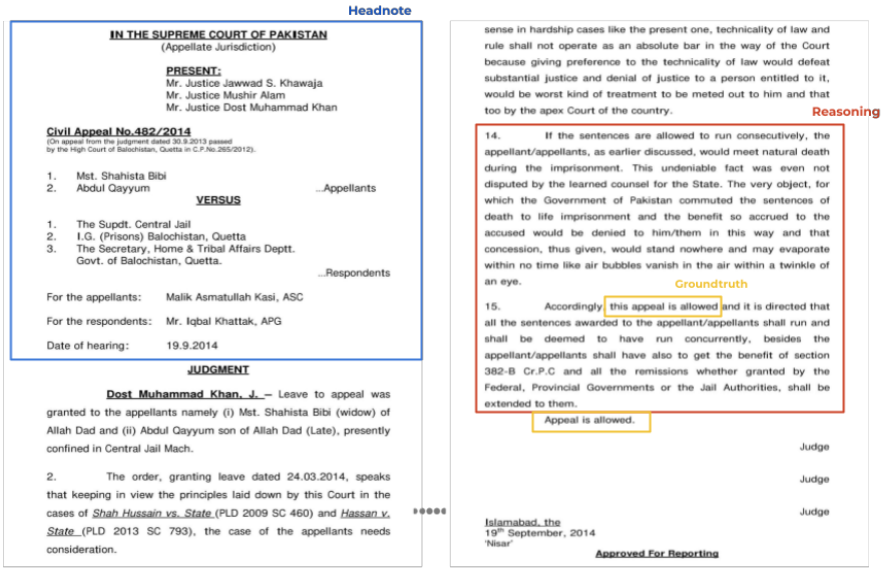

Fig. 4. Sample Case File. The case file contains two main sections: the **head-note** (i.e. meta-data of the case) and the **judgment**. The blue box represents the headnote section of the case file, the yellow boxes refer to the phrases that are used to label this case file for the Legal Judgment Prediction (LJP) task and lastly, the red box refers to the ground truth for the Legal Judgment Explanation Extraction task.

a strategic approach: breaking down each document into manageable 512-token chunks with varying overlaps of neighboring chunks (ranging from 70 to 200 tokens), thereby maintaining document context. Each chunk was then assigned the same label as the original document, and the transformer model was fine-tuned with each chunk treated as an individual input during training.

The transformer followed by any LSTM, CNN or fusion model can not only quickly obtain effective information, but also obtain contextual information and global feature information of the text [5]. For validation of this statement, we used the embeddings files from the transformers were then passed into a CNN, Bi-GRU and Bi-GRU with attention models. We divide the documents into chunks of 512-tokens with overlapping tokens as already mentioned. Each chunk is them fed into the transformer model to extract the embeddings, then embeddings for each document are fed into the sequential model to capture inter-chunk dependency. A final dense layer then classifies the document either as allowed or dismissed.

Our hypothesis posits that the results will significantly improved with the use of domain specific models. The proposed models are finetuned to better understand legal language, which may result in more accurate analysis of legal documents. We expect the improvement because these models are trained to

grasp the unqiue and complex aspects of legal text, making them more effective for legal text classification.

5.3 Hyperparameters

The proposed architecture was implemented using PyTorch. We carried out the training of our architecture on different single GPUs including NVIDIA RTX-3080 and A5000 GPU. For the transformer, we used 12 encoders as this setting gave us the best results. We kept the batch size equal to 8 for fine-tuning the transformers. As for the training of other models, the batch size 32 was consistent. We tested with an overlap of n-tokens $n \in [70, 100, 200]$ with a neighbouring chunk to test if this parameter has an impact on the results. The Adam optimizer was used to train the model. Additionally, different learning rates were tested including 1e-3, 1e-5, 2e-5. We got optimal results with the learning rate equal to 2e-5. Other settings of learning rate diminished the training by either diverging the loss for a higher learning rate or slow convergence for a lower learning rate.

6 Results, Analysis, and Discussion

In this section, we discuss in detail the model performance from different aspects, where it is better and how can it be improved. The results have significantly improved with the use of domain-specific models like Legal-BERT [28], Legal-RoBERTa [26], CaseLawBERT [34], InLegalBERT [33], and InCaseLawBERT [33]. Due to the limited size of our dataset, we focused on fine-tuning these pre-existing pre-trained models rather than training a transformer model from scratch.

In particular, we did extensive experimentation as provided by Malik et al.'s study [11] and replicated the different types of models: Classical Models, Transformer Models, and Transformer + Sequential Models. Table 2 summarizes the performance of baseline models on our VerdictVaultPK Dataset.

From the results in Table 2, it can be interpreted that classical and sequential models did not perform so well. The tf-idf computes document similarity directly in the word-count space and is slower for large vocabularies. Similarly, GRUs are good at caputuring semantics but are unable to capture long-term dependencies for large sequences due to vanishing/exploding gradients.

The use of embeddings learned using a Transformer model like BERT achieves a significant improvement of 8%-10% in the accuracy of prediction [9]. The main characteristics of Transformers are that they are non-sequential meaning sentences are processed as a whole rather than word by word. Transformers make use of self-attention to capture the semantic relations between the words. Additionally, due to their non-recursive property, they do not suffer long dependency issues and can retain information for larger time stamps. Moreover, multi-head attention and positional embeddings both provide information about the relationship between different words. Transformer architecture performs best when capturing the context and semantics of the text. Building upon this intuition,

Table 2. Results for Legal Judgment Prediction (LJP) on VerdictVaultPK Dataset using the approaches proposed in Malik et al.'s study [**Comparisons to BASELINES**]

Model	Precision	Recall	F1-Score	Accuracy
Classical Models on VerdictVaultPK Dataset				
Support Vector Machine (tf-idf)	52.78	52.59	49.86	50.15
Logistic Regression (tf-idf)	66.34	62.63	62.16	66.18
Random Forest (tf-idf)	70.95	57.30	52.26	63.44
Naive Bayes (tf-idf)	70.67	50.49	37.83	58.16
Decision Trees (tf-idf)	59.73	59.66	59.69	60.80
word2vec + BiGRU + att.	60.82	50.47	55.16	58.06
Hierarchical Attention Network (HAN)	62.25	50.14	55.54	57.86
Transformers on VerdictVaultPK Dataset				
BERT	79.66	79.68	79.67	80.15
RoBERTa	82.65	81.87	82.17	82.79
XL-Net	84.16	83.00	83.40	**84.06**
Transformers + Seq Models on VerdictVaultPK Dataset				
BERT + CNN	83.26	83.20	83.23	83.65
RoBERTa + CNN	84.19	84.70	84.38	84.63
XL-Net + CNN	85.26	85.99	85.44	**85.61**
BERT + BiGRU	82.96	80.99	81.57	82.48
RoBERTa + BiGRU	83.69	84.49	83.79	83.95
XL-Net + BiGRU	84.33	85.11	84.47	**84.63**
BERT + BiGRU + att.	83.03	81.46	81.97	82.77
RoBERTa + BiGRU + att.	84.44	85.23	84.57	84.73
XL-Net + BiGRU + att.	84.70	85.28	84.89	**85.12**

we experimented with different variants of BERT models that were further fine-tuned or pre-trained on legal datasets (as can be seen in Table 3).

For the initial set of experiments with BERT variants and mainly due to the limitation on the number of input tokens to BERT and other transformer models, we followed the approach [11] given below:

1. For each case, the document was divided it into chunks of 480 tokens each with an overlap of n-tokens $n \in [70, 100, 200]$ with a neighboring chunk.
2. Each chunk was assigned the same label as the original case document.
3. Model was trained to treat each chunk as a distinct example.

For testing, we only used 'last 512 tokens' (as they contain the most meaningful information [11]) of the document). The transformer models outperformed classical and sequential models as can be seen in Table 2 and 3.

Table 3. Comparison of **Proposed Models** for Legal Judgment Prediction (LJP) on our VerdictVaultPK Dataset and $ILDC_s$ Dataset.

Model	Precision	Recall	F1-Score	Acc	Precision	Recall	F1-Score	Acc
	VerdictVaultPK Dataset				$ILDC_s$ **Dataset**			
Transformers								
Legal-BERT [28]	86.81	87.04	86.19	87.19	76.81	76.74	76.71	76.73
Legal-RoBERTa [26]	86.51	86.73	86.61	86.93	74.99	74.71	74.62	74.68
CaseLawBERT [34]	85.71	85.32	85.49	85.92	72.86	72.47	72.33	72.44
InLegalBERT [33]	**87.9**	**88.16**	**88.02**	**88.26**	**79.01**	**78.96**	**78.95**	**78.95**
InCaseLawBERT [33]	87.33	87.16	87.24	87.58	74.43	74.18	74.09	74.15
Transformers+Seq								
Legal-BERT+BiGRU	86.08	85.73	85.89	86.31	77.03	76.79	76.73	76.78
Legal-RoBERTa+BiGRU	85.83	85.94	85.87	86.20	76.63	75.56	75.28	75.52
CaseLawBERT+BiGRU	85.69	85.36	85.47	85.91	72.72	71.11	72.34	72.42
InLegalBERT+BiGRU	**87.12**	**87.39**	**87.23**	**87.48**	**78.04**	**77.96**	**78.04**	**77.98**
InCaseLawBERT+BiGRU	85.67	86.16	85.86	86.11	74.83	74.82	74.82	74.83

Table 3 shows the results of the proposed models on our dataset (i.e. VerdictVaultPK Dataset) and the $ILDC_s$ dataset [11]. The results clearly show that the use of domain-specific variants like Legal-BERT [28], Legal-RoBERTa [26], CaseLawBERT [34], InLegalBERT [33], and InCaseLawBERT [33] improved the results with a margin of 3-4% in comparison to the baseline XL-Net. This is because InLegalBERT is a variant of BERT especially designed for understanding the legal text making it aware of nuances and intricacies of legal language, whereas the XL-NET is a more general-purpose language model. Even though both models were fine-tuned for legal tasks, InLegalBERT's extensive pertaining on legal corpora may have been more suitable for capturing the legal-specific

information for extracting the explanations. The fine-tuning process could have emphasized the importance of certain features or contextual cues specific to legal documents, leading to better explanation quality. The same models were then tested using the proposed hierarchical configuration. Table 3 reports that the domain-specific hierarchical pipeline beats the baseline/sota 'XL-Net + BiGRU' by a margin of 2-3%.

For our secondary task, Legal Judgment Explanation Extraction, the best hierarchical configuration (i.e. InLegalBERT + BiGRU) was used following the masking procedure explained earlier. Various metrics such as ROUGE-1, ROUGE-2, ROUGE-L, BLEU, Overlap-Min, Overlap-Max, and Jaccard Similarity are commonly employed to gauge the quality and similarity of generated explanations against reference texts. ROUGE family compares machine-generated text and the reference text using overlapping n-grams, word sequences that appear in both texts. Whereas the BLEU score evaluates the text using the precision of n-grams. Overlap-Min and Overlap-Max determine the minimum and maximum overlap ratios between the generated and reference texts. Lastly, the Jaccard Similarity quantifies the similarity between sets of words in generated and reference texts. Higher values for all metrics means greater overlap, meaning the model is able to generate text that is close to reference text. Table 4 gives a summary of comparison of the model outputs with reference text.

Table 4. Comparison of Proposed Models for Legal Judgment Explanation Extraction (LJEE) on our VerdictVaultPK Dataset and $ILDC_s$ Dataset.

	VerdictVaultPK Dataset		$ILDC_s$ Dataset [11]	
Metric	XLNet + BiGRU [11]	**InLegalBERT + BiGRU**	XLNet + BiGRU [11]	**InLegalBERT + BiGRU**
Jaccard Similarity	0.5652	**0.5726**	0.4627	**0.4777**
Overlap-Min	0.8859	**0.8951**	0.7181	**0.7457**
Overlap-Max	0.5471	**0.5721**	0.5582	**0.5670**
ROUGE-1	0.6771	**0.6890**	0.5876	**0.6061**
ROUGE-2	**0.6146**	0.5897	0.4561	**0.4804**
ROUGE-L	0.6761	**0.6841**	0.5728	**0.5950**
BLEU	0.3783	**0.4283**	0.4209	**0.4244**

The ROUGE scores indicate relativley higher overlap between model generated text and reference text as compared to baseline model. However, the BLEU score is comparatively lower, indicating a less precise match in terms of n-gram precision between the generated and reference texts. Similarly, the Jaccard Similarity also lies in moderate range suggesting moderate level of similarity between the sets of words in model generated text and reference text. The Overlap-Min and Overlap-Max also suggest substantial agreement. It can be inferred that the model performance has been improved with transfer learning if a model is

fine-tuned on a dataset with similar domain. Although the results needs improvement but it sets a promising research direction to explore and develop explainable models that can not only capture the decision-making chunks from the given text but will also be able to generate explanation on it's own after understanding the context of the case file.

The proposed approach along with it's advantages has some limitations too. One potential limitation is that the explanation task is heavily dependent on the accuracy of prediction task, accurate predictions by the model enable more reliable explanations. If the model's predictions are biased or inaccurate, the explanations generated using this method may also be flawed or misleading. Additionally, if the original model struggles with certain types of legal documents or cases, it may produce unreliable explanations for those instances. Table 5 somes of the cases where the proposed model couldn't perform well. We aim to improve these results in future.

Table 5. Model Failure Cases: Lowest scores in the VerdictVaultPK dataset for our proposed model's performance on individual documents.

Metric	Example 1	Example 2	Example 3	Example 4
Jaccard Similarity	0.2158	0.2628	0.2906	0.2789
Overlap-Min	0.4965	0.6602	0.4968	0.9594
Overlap-Max	0.2763	0.3039	0.4118	0.2823
ROUGE-1	0.3378	0.3918	0.4206	0.4364
ROUGE-2	0.1274	0.2122	0.1744	0.3189
ROUGE-L	0.3108	0.3599	0.3644	0.4304
BLEU	0.1781	0.1154	0.3441	0.1407

7 Conclusion and Future Work

This research introduces the VerdictVaultPK corpus of Pakistani rental property cases for the LJPEE task. From the experiments, it can be seen that the domain-specific models outperform the baseline models with an increase of $3-4\%$ in accuracy. The usage of a domain-specific model in the explanation module also showed an improvement in the results. One limitation of this research is that we are extracting the information that is already a part of the case files. One future direction of this research can be the generation of explanation i.e. Legal Judgment Explanation Generation (LJEG) based on raw case text. The proposed model can be used to annotate the dataset for the task of LJEG and then train another model that is capable of generating the reasoning. Legal Judgment Explanation Generation (LJEG) can be treated as a question-answering task. Another future direction is to overcome the token limit of Transformer models by the Longformers [29] (a variant of Transformers) for the task of Legal Judgment Prediction and Explanation (LJPE).

References

1. Z. Liu, H. Chen: "A Predictive Performance Comparison of Machine Learning Models for Judicial Cases," in *2017 IEEE Symposium Series on Computational Intelligence (SSCI) (pp. 1-6). IEEE*, (2017)
2. M. B. L. Virtucio, J. A. Aborot, J. K. C. Abonita, R. S. Avinante, R. J. B. Copino, M. P. Neverida, V. O. Osiana, E. C. Peramo, J. G. Syjuco, G. B. A. Tan: "Predicting Decisions of the Philippine Supreme Court using Natural Language Processing and Machine Learning." in *2018 IEEE 42nd annual computer software and applications conference (COMPSAC) (Vol. 2, pp. 130-135). IEEE*, (2018)
3. R. A. Shaikha, T. P. Sahua, V. Anandb: "Predicting Outcomes of Legal Cases based on Legal Factors using Classifiers.," in *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC) (Vol. 2, pp. 130-135). IEEE*, (2018)
4. M. Medvedeva, M. Vols, M: "Using Machine Learning to Predict Decisions of the European Court of Human Rights," in *Artif Intell Law 28, 237–266*, (2020)
5. L. Liu, D. An, Y. Wang, X. Ma, C. Jiang: "Research on Legal Judgment Prediction Based on Bert and LSTM-CNN Fusion Model," in *2021 3rd World Symposium on Artificial Intelligence (WSAI) (pp. 41-45). IEEE*, (2021)
6. B. Strickson, B. D. L. Iglesia: "Legal Judgement Prediction for UK Courts," in *Proceedings of the 2020 the 3rd International Conference on Information Science and System (pp. 204-209)*, (2020)
7. V. G. Pillai, L. R. Chandran: "Verdict Prediction for Indian Courts Using Bag of Words and Convolutional Neural Network," in *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT) (pp. 676-683). IEEE)*, (2020)
8. K. Kowsrihawat, P. Vateekul, P. Boonkwan: "Predicting Judicial Decisions of Criminal Cases from Thai Supreme Court Using Bi-directional GRU with Attention Mechanism.," in *2018 5th Asian Conference on Defense Technology (ACDT) (pp. 50-55). IEEE*, (2018)
9. Y. Wang, J. Gao, J. Chen: "Deep learning algorithm for judicial judgment prediction based on BERT," in *2020 5th International Conference on Computing, Communication and Security (ICCCS) (pp. 1-6). IEEE*, (2020)
10. I. Chalkidis, I. Androutsopoulos, N. Aletras: "Neural legal judgment prediction in English," in *English. arXiv preprint arXiv:1906.02059*, (2019)
11. V. Malik, R. Sanjay, S. K. Nigam, K. Ghosh, S. K. Guha, A. Bhattacharya, A. Modi: "ILDC for CJPE: Indian Legal Documents Corpus for Court Judgment Prediction and Explanation," in *arXiv preprint arXiv:2105.13562*, (2021)
12. Cahyani, D. E., Patasik, I.: "Performance Comparison of tf-idf and word2vec Models for Emotion Text Classification' in *Bulletin of Electrical Engineering and Informatics, 10(5), 2780-2788*, (2021)
13. C. Yahui, "Convolutional Neural Network for Sentence Classification.," in *(Master's thesis, University of Waterloo)*, (2015)
14. P. Zhou, Z. Qi, S. Zheng, J. Xu, H. Bao, B. Xu: "Text Classification Improved by Integrating Bidirectional LSTM with Two-dimensional Max Pooling.," in *arXiv preprint arXiv:1611.06639*, (2016)
15. J. Xie, B. Chen, X. Gu, F. Liang, X. Xu:"Self-Attention-Based BiLSTM Model for Short Text Fine-Grained Sentiment Classification.," in *IEEE Access, 7, pp.180558-180570*, (2019)
16. Dietterich, T. G: "Ensemble Methods in Machine Learning' in *International workshop on multiple classifier systems (pp. 1-15). Berlin, Heidelberg: Springer Berlin Heidelberg*, (2000)

17. S. Lai, L. Xu, K. Liu, J. Zhao: "Recurrent Convolutional Neural Networks for Text Classification." in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, (2015)
18. A. Joulin, E. Grave, P. Bojanowski, T. Mikolov: "Bag of Tricks for Efficient Text Classification," in *arXiv preprint arXiv:1607.01759*, (2016)
19. R. Johnson, T. Zhang: "Deep Pyramid Convolutional Neural Networks for Text Categorization.," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 562-570)*, (2017)
20. J. Li, W. Monroe, D. Jurafsky: "Understanding Neural Networks through Representation Erasure' in *arXiv preprint arXiv:1612.08220*, (2016)
21. C. Xiao, H. Zhong, Z. Guo, C. Tu, Z. Liu, M. Sun, Y. Feng, X. Han, Z. Hu, H. Wang, J. Xu: "CAIL2018: A Large-Scale Legal Dataset for Judgment Prediction.," in *arXiv preprint arXiv:1807.02478*, (2018)
22. Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, E. Hovy: "Hierarchical Attention Networks for Document Classification" in *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies, pp. 1480-1489*, (2016)
23. Y. Liu, M. Ott, N. Goyal, J. Du, M.Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov: "Roberta: A Robustly Optimized BERT Pretraining Approach" in *arXiv preprint arXiv:1907.11692*, (2019)
24. He, P., Liu, X., Gao, J., Chen, W: "DeBERTa: Decoding-enhanced BERT with Disentangled Attention' in *arXiv preprint arXiv:2006.03654*, (2020)
25. Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, Q. V. Le: "Xlnet: Generalized Autoregressive Pretraining for Language Understanding" in *Advances in Neural Information Processing Systems 32*, (2019)
26. S. Geng, R. Lebret, K. Aberer: "Legal transformer models may not always help." in *arXiv preprint arXiv:2109.06862*, (2021)
27. I. Chalkidis, I. Androutsopoulos, N. Aletras: "Neural Legal Judgment Prediction in English," in *arXiv preprint arXiv:1906.02059*, (2019)
28. I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, I. Androutsopoulos: "LEGAL-BERT: The muppets straight out of law school," in *arXiv preprint arXiv:2010.02559*, (2019)
29. I. Beltagy, M. E. Peters, A. Cohan: "Longformer: The long-document transformer." in *arXiv preprint arXiv:2004.05150*, (2020)
30. J. Devlin, M. Chang, K. Lee, K. Toutanova: "Bert: Pre-training of deep bidirectional transformers for language understanding." in *arXiv preprint arXiv:1810.04805*, (2018)
31. A. Gasparetto, M. Marcuzzo, A. Zangari, A. Albarelli: "A Survey on Text Classification Algorithms: From Text to Predictions' in *Information, 13(2), p.83*, (2022)
32. S. Long, C. Tu, Z. Liu, M. Sun: "Automatic Judgment Prediction via Legal Reading Comprehension' in *China National Conference on Chinese Computational Linguistics (pp. 558-572). Springer, Cham*, (2019)
33. Paul, S., Mandal, A., Goyal, P., Ghosh, S: "Pre-trained Language Models for the Legal Domain: A Case Study on Indian Law' in *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law (pp. 187-196)*, (2023)
34. Zheng, L., Guha, N., Anderson, B. R., Henderson, P., Ho, D. E: "When does pretraining help? assessing self-supervised learning for law and the casehold dataset of 53,000+ legal holdings.' in *In Proceedings of the eighteenth international conference on artificial intelligence and law (pp. 159-168)*, (2021)
35. Zheng, L., Guha, N., Anderson, B. R., Henderson, P., Ho, D. E: "Paragraph-level rationale extraction through regularization: A case study on European court of

human rights cases' in *In Proceedings of the eighteenth international conference on artificial intelligence and law (pp. 159-168)*, (2021)