ClearCase : Legal Text Classification with Explainable AI

Arun Karthik Sengottuvel
sengottu@usc.edu

Ashwinkumar Venkatnarayanan
venkatna@usc.edu

Indira Swaminathan
iswamina@usc.edu

Bhavya Avuthu
avuthu@usc.edu

Mona Teja Kurakula
mkurakul@usc.edu

Mohan Sai Ganesh Kanna
mkanna@usc.edu

Abstract

Classifying legal documents is essential for organizing case law and assisting legal professionals in managing vast volumes of legal information. However, the nuanced and complex nature of legal language presents significant challenges for both experts and non-experts. To address these challenges, ClearCase introduces a Transformer-based legal text classification system that leverages state-of-the-art models such as LegalBERT and RoBERTa, enhanced with Explainable AI (XAI) techniques such as SHAP and LIME. The system not only delivers high accuracy in classifying legal documents but also identifies and visualizes the specific portions of case text that influence each decision. By combining performance with interpretability, ClearCase makes AI-driven legal analysis more transparent, trustworthy, and actionable for practitioners.

1 Introduction

1.1 Problem Statement

Exponential growth of legal texts has made manual classification time intensive and prone to errors. While AI may offer a scalable solution to automate this process, it introduces new challenges. Legal documents contain specialized jargon, and require high interpretability due to the critical nature of legal decisions. Existing AI solutions lack transparency, making their decision-making process difficult to trust in sensitive domains such as law.

1.2 ClearCase: Our Contribution

Legal professionals often spend considerable time manually sorting and analyzing vast amounts of legal documents. Automating this process with transformer-based models increases efficiency, and reduces human error. What sets this approach apart is its emphasis on explainability, allowing legal professionals to trust and validate AI-generated classifications with confidence. This not only improves the effectiveness of the technology, but also makes it ideal for real-world legal settings, where transparency, trust, and accountability are essential.

1.3 How is it interesting?

The intersection of deep learning, legal technology, and explainability presents unique challenges due to the complexity of legal language. Legal documents include specialized jargon, citations, and intricate structures that demand domain expertise. ClearCase investigates how fine-tuned transformer models improve classification accuracy over general-purpose models. By incorporating XAI techniques, it offers insights into the models' decision-making processes.

2 Related Work

This section reviews prior works that have contributed to the fields of legal text classification, transformer-based models, and XAI, providing the foundation for the proposed approach.

Transformer-based architectures have revolutionized text classification tasks by leveraging self-attention mechanisms to capture contextual relationships. Chalkidis et al. (2020) introduced LEGAL-BERT, a domain-specific variant of BERT pre-trained on legal corpora, demonstrating its superior performance in multi-label classification and named entity recognition tasks compared to general-purpose models. Similarly, Limsopatham (2021) explored how BERT can be effectively adapted for legal NLP tasks using datasets such as datasets for the European Convention on Human Rights (ECHR) violations and overruling tasks. Their findings revealed that truncating long documents negatively impacts performance, underscoring the need for specialized architectures.

To address the limitations of standard transformers in processing long sequences, Beltagy et al. (2020) proposed Longformer, which employs a linear attention mechanism capable of handling sequences up to thousands of tokens. Building on this, Mamakas et al. (2022) modified Longformer and LegalBERT to accommodate sequences as long as 8,192 tokens, achieving state-of-the-art results in legal document classification. While these advancements focus on improving accuracy and scalability for lengthy texts, they often overlook interpretability and explainability that are critical requirements for real-world legal applications.

Explainability in AI systems is essential for fostering trust and transparency, particularly in sensitive domains like law. Several XAI techniques have emerged to address this need. Lundberg and Lee (2017) introduced SHAP (SHapley Additive exPlanations), a game-theoretic approach to quantifying feature contributions to predictions, ensuring local accuracy and consistency. Ribeiro et al. (2016) proposed LIME (Local Interpretable Model-agnostic Explanations), which approximates black-box models with interpretable surrogates to explain individual predictions. Attention visualization tools, such as BertViz by Vig (2019), provide insights into token-level attention weights within transformer models. However, Jain and Wallace (2019) cautioned that attention weights do not always correlate with causal influence, underscoring the need for more robust interpretability methods.

Despite their success in domains like healthcare and finance, XAI techniques remain underutilized in legal NLP. Legal professionals require explanations aligned with legal reasoning, such as identifying key citations or phrases driving predictions. Current general-purpose XAI methods often fail to capture the nuanced structure of legal argumentation, highlighting the need for domain-specific adaptations.

3 Data Preparation

3.1 Dataset Description

The proposed Legal Text Classification dataset[1] by Mohankumar (2023) consists of 24,985 records, where each record is assigned a unique `case_id` in the format `CaseX`, with `X` indicating the case number. This `case_id` serves as a identifier for each case and does not contain duplicate or missing values.

[1] https://www.kaggle.com/datasets/amohankumar/legal-text-classification-dataset

Key Fields:

- `case_id`: A unique identifier for each case.
- `case_outcome`: The verdict or outcome of the case, which serves as the class label for classification tasks.
- `case_title`: The headline or name of the case referred.
- `case_text`: The textual content of the case, which provides detailed information regarding the case.

3.2 Exploratory Data Analysis (EDA)

Upon initial inspection, it is found that the `case_id`, `case_title`, and `case_outcome` columns have no missing values. However, the `case_text` i.e., our key input column for classification contains 176 missing values. As a result, drop these 176 records, leaving 24,809 complete records for further analysis and model training. The distribution of the `case_outcome` field is highly imbalanced, with some categories more prevalent than others.The distribution is summarized in Table 1.

Case Outcome	Count (%)	
Affirmed	106	(0.43%)
Applied	2438	(9.83%)
Approved	108	(0.44%)
Cited	**12110**	**(48.81%)**
Considered	1699	(6.85%)
Discussed	1018	(4.10%)
Distinguished	603	(2.43%)
Followed	2252	(9.08%)
Referred To	4363	(17.59%)
Related	112	(0.45%)

Table 1: Case Outcome Distribution in the Dataset

From Table 1, it is clear that the `cited` outcome dominates the dataset, comprising 48.8% of the total records. The `referred to` outcome follows, making up approximately 17.6% of the data. Several other outcomes, such as `applied`, `considered`, and `followed`, are less frequent. The dataset is imbalanced, which will need to be carefully handled during model training through techniques like stratified sampling, resampling, or class-weight adjustments.

A key part of our EDA focused on quantifying the length of legal case texts after tokenization, as this directly impacts model selection and preprocessing strategies. To ensure a robust and conservative approach, we analyzed each document's length using both Byte Pair Encoding (BPE) and WordPiece tokenization algorithms - two widely used subword tokenization methods in modern Natural Language Processing (NLP). For each case, we recorded the maximum token count observed across these two algorithms, thereby safeguarding against underestimating sequence length for downstream modeling.

The distribution of tokenized `case_text` lengths is visualized in the below bar chart:

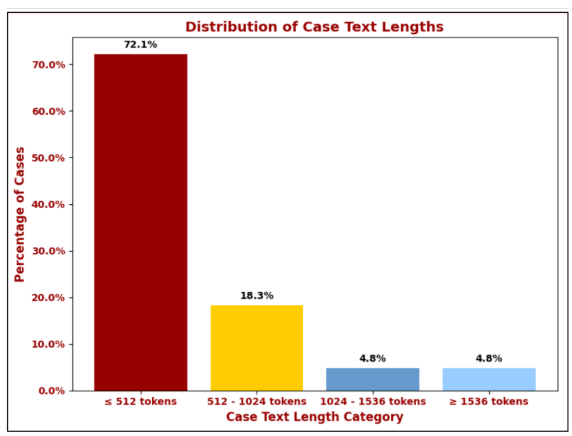


Figure 1: Data Distribution by Token Size.

Length of `case_text`:

- ≤ 512 tokens: 72.1% of cases
- 512 –1024 tokens: 18.3% of cases
- 1024 –1536 tokens: 4.8% of cases
- ≥ 1536 tokens: 4.8% of cases

This analysis highlights that while most legal documents (over 70%) fit within the standard 512-token input limit of transformer models like LegalBERT and RoBERTa, a significant minority (nearly 28%) exceed this threshold. This long-tail distribution underscores the necessity for careful truncation or segmentation strategies, as these longer documents risk losing important context if not handled appropriately.

By using the maximum token length across both Byte-Pair Encoding (BPE) and WordPiece tokenizers, we ensured our preprocessing pipeline was robust to the nuances of different tokenization schemes, thereby preventing inadvertent information loss during model input preparation. This conservative approach informed our subsequent choice of truncation strategies (such as head-tail truncation) and highlights the ongoing need for models capable of handling longer input sequences in legal NLP.

3.3 Data Deduplication

As part of our data cleaning pipeline, we implemented a critical pre-processing step to remove exact duplicates from the dataset, where both the `case_text` and `case_outcome` were identical. These exact duplicates represented redundant information that could potentially bias our model training process.

- **Training Bias:** Duplicated examples receive disproportionate weight during training, potentially biasing the model toward these specific patterns.
- **Artificial Performance Inflation:** When duplicates appear across training and evaluation sets, they create a false impression of model generalization ability.
- **Inefficient Resource Utilization:** Processing redundant data wastes computational resources during training.

Data Deduplication process involved grouping records by their `case_text` and `case_outcome` values, then retaining only the first occurrence of each unique combination. This approach eliminated redundant exposures of identical examples to the model and reduced the class imbalance to a great extent as most of such duplicates were found in majority classes.

3.4 Data Sampling

To start, the target variable, `case_outcome`, was encoded into numeric labels using the `LabelEncoder`. This encoding process transformed the `case_outcome` into numerical values (0-9 as there are 10 classes), allowing for seamless integration.

Following this, a stratified sampling approach was employed to ensure balanced splits across the target variable. The dataset was divided into three subsets: `Train` (80%), `Validation` (10%) and `Test` (10%) sets, preserving the distribution of `case_outcome` labels within each subset. This stratified split ensures that all splits maintain the class proportions, addressing potential class imbalances in the dataset. These final splits of training, validation, and testing were saved in separate CSV files, ensuring the dataset is properly prepared for the next steps in model development. This marks an end to the data preparation phase, ensuring that the dataset is optimized for training or fine-tuning transformer models with well-distributed outcomes.

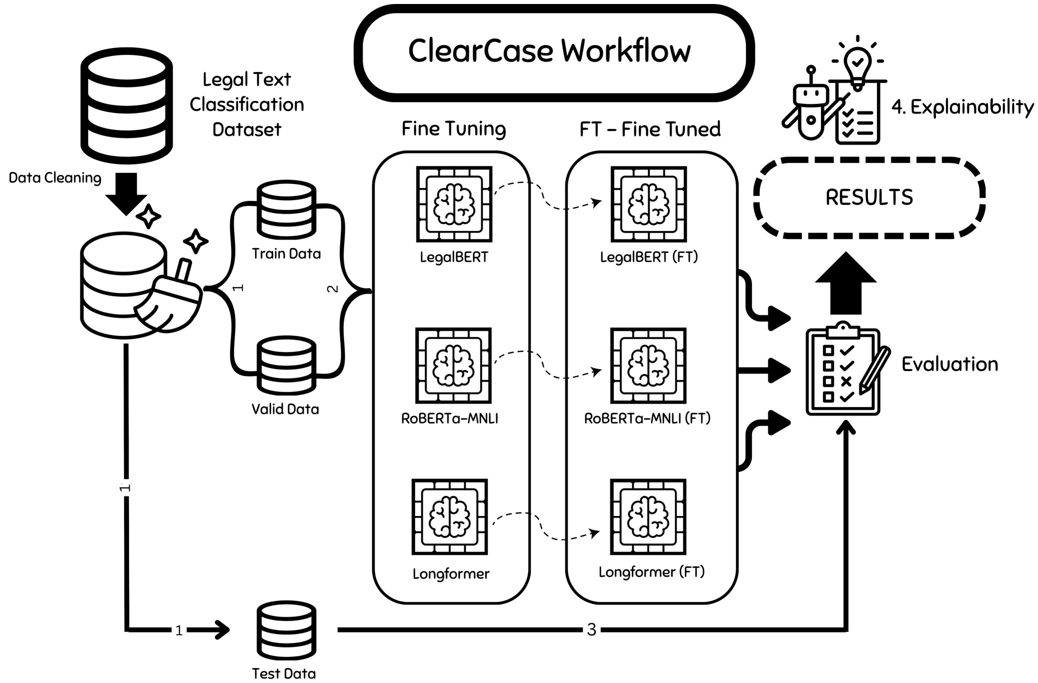


Figure 2: ClearCase Workflow.

4 Methodology

The workflow implements a comprehensive framework for legal text classification leveraging advanced transformer architectures. The approach begins with zero-shot classification using RoBERTa-large-MNLI as a baseline, establishing fundamental performance benchmarks without domain-specific training.

4.1 Baseline

To establish a baseline, we utilized the `roberta-large-mnli`[2] from Hugging Face's Transformer library for zero-shot classification. This model was trained on the Multi-Genre Natural Language Inference (MNLI) tasks and supports textual entailment-based inference. Each legal case's text (`case_text`) was treated as the "premise", while candidate labels such as `affirmed`, `cited`, and `applied` were hypothesized in the form: "`This case was [label].`" The model evaluated the likelihood of each hypothesis being entailed by the premise and selected the label with the highest entailment probability as its prediction. This phase provided insights into how general-purpose models perform on domain-specific tasks like legal text classification. While it offered a quick baseline, it underscored the need for supervised fine-tuning tailored to legal content for improved accuracy.

[2] https://huggingface.co/roberta-large-mnli

4.2 Supervised Training or Fine-Tuning

To improve legal text classification performance, we employed supervised fine-tuning on two state-of-the-art transformer models. Each model was fine-tuned to classify legal cases into one of ten predefined class labels (`case_outcomes`).

Model Selection:

- **LegalBERT**[3]: A domain-specialized model pre-trained on legal corpora.
- **RoBERTa**: A general-purpose model known for robust language understanding.

`case_texts` were tokenized using model-specific tokenizers to determine their token lengths. RoBERTa, which uses Byte-Pair Encoding (BPE), and LegalBERT, which relies on WordPiece tokenization, both support a maximum input length of 512 tokens. However, many legal documents exceed this limit, making it necessary to apply truncation strategies to ensure effective processing. To address this, we evaluated two truncation strategies.

4.3 Truncation

End Truncation (Head-Only): This approach retained only the first 512 tokens of each document, discarding the rest. While this strategy assumes that introductory context is sufficient for classification, it risks losing critical information located near the middle or end of documents.

Start+End Truncation (Head-Tail): To mitigate the shortcomings of end truncation, this strategy preserved both the first 256 and last 256 tokens of each document. This balanced retention allowed the model to capture introductory context as well as concluding insights both crucial for legal case analysis.

Despite implementing truncation strategies, a significant proportion of legal texts still suffer from information loss due to the 512-token limitation of standard transformer architectures.

[3] https://huggingface.co/nlpaueb/legal-bert-base-uncased

As evident from our token distribution analysis (Figure 1), approximately 27.9% of documents exceed this threshold, with 9.6% containing more than 1024 tokens. To address this challenge during model training, we implemented a dynamic chunking mechanism.

4.4 Chunking

For documents exceeding the 512-token limit, we segmented the text into multiple chunks of approximately equal length. Each chunk was processed independently, maintaining the same class label as its parent document. This approach enabled the model to process the entirety of lengthy documents during training, theoretically capturing complete information that would otherwise be lost through truncation. The chunking mechanism was applied dynamically based on document length, with longer documents producing more chunks.

4.5 Chunking Limitations

While chunking improves data utilization compared to truncation strategies, it still represents a compromise rather than a comprehensive solution for handling long legal documents. During training, chunking allows models to process more content from each document by dividing texts exceeding 512 tokens into multiple segments of manageable length. However, this segmentation inherently fragments the document's logical flow and semantic cohesion.

The chunking approach introduces two significant limitations:

1. **Contextual Fragmentation:** Dividing documents into chunks disrupts the natural progression of legal reasoning. Arguments in legal texts typically build on premises established earlier, with conclusions that depend on this progressive development.

Chunking breaks these connections, preventing models from learning the complete argumentative structure.

2. **Inference Inconsistencies:** During testing, chunked segments from the same document may receive contradictory classifications based on their partial content. As each legal case can have only one definitive outcome in our classification schema, reconciling these potentially conflicting predictions becomes problematic.

This approach represents only an incremental improvement over simple truncation. It captures more content but still losing critical inter-segment relationships and the document's holistic legal reasoning. These limitations further highlight the need for architectures specifically designed for long-document processing, such as **Longformer** [4] with its extended context window of 4,096 tokens.

To address the persistent challenges of context fragmentation from truncation and chunking, we implemented Longformer's extended 4,096-token context window solution alongside standard transformer architectures. While this architectural enhancement improves holistic document processing, maximizing classification performance across all models (including Longformer) required a careful initialization and rigorous optimization of training parameters.

The following section details the training approach and hyperparameter tuning, which systematically calibrates learning dynamics to extract the best performance from conventional and long-context transformer architectures.

[4] https://huggingface.co/allenai/longformer-base-4096

4.6 Training and Hyperparameter Tuning

Initial training began with all transformer layers frozen, allowing only the classification head to be trained. This approach preserved the valuable linguistic knowledge encoded in the pre-trained weights while adapting only the task-specific layers to legal classification. After performance plateaued (around 2-3 epochs), we gradually unfroze deeper transformer layers in a top-to-bottom sequence, starting with the top layer.

We applied a significantly reduced learning rate to prevent catastrophic forgetting of pre-trained information. This gradual unfreezing strategy mitigates the risk of destroying valuable general language understanding while allowing the model to adapt its representations to the legal domain. Each subsequent layer was unfrozen only after the previous layers showed convergence in validation performance.

Hyperparameters:

- **Optimization:** Models were fine-tuned using the AdamW optimizer with a learning rate of 2×10^{-5}, epsilon of 1×10^{-8}, and weight decay of 0.01.
- **Learning Rate Scheduler:** A linear scheduler with warm-up steps set to 10% of total training steps was applied.
- **Gradient Clipping:** Gradient clipping with a max norm of 1.0 was used to ensure stable training.
- **Loss Function:** CrossEntropy Loss was employed to update the weights for the multi-class classification problem.

Training was conducted over 6 epochs with a batch size of 32. Validation accuracy was calculated after each epoch, and the best-performing model was saved for evaluation on the test set.

4.7 Explainability Techniques (XAI)

ClearCase integrates two explainability techniques, each offering unique insights into model decision-making.

SHAP (SHapley Additive exPlanations):

SHAP values quantify the contribution of each word or phrase to the final classification decision using game theory principles. For legal texts, this highlights which specific legal terms, citations, or factual statements most strongly influence the model's outcome prediction.

SHAP:

- Calculates feature attribution scores that sum to the model's output
- Generates consistent and locally accurate explanations
- Identifies both supporting and contradicting evidence within the text

LIME (Local Interpretable Model-agnostic Explanations):

LIME generates explanations by creating simplified, interpretable surrogate models that approximate the behavior of our complex transformer models for individual predictions.

LIME:

- Highlights important words and phrases with color-coded visualizations
- Provides intuitive, text-based explanations accessible to non-technical users
- Offers localized insights specific to each document's classification.

5 Results and Discussion

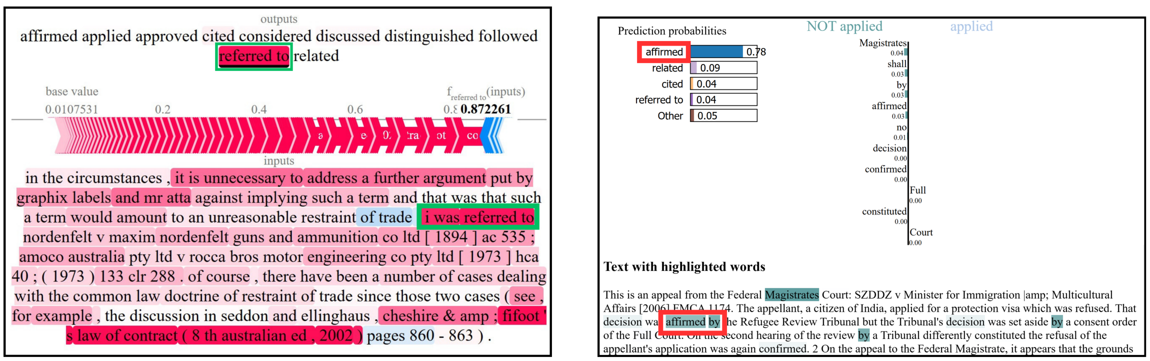
In this study, we fine-tuned three transformer-based models: Legal-BERT, RoBERTa, and Longformer, to classify legal case texts into one of ten predefined outcomes. All models were trained using identical experimental settings, with consistent training data, learning rate, batch size, and number of epochs. The primary differences lie in their architectures and pretraining corpora, allowing us to evaluate the effect of model design and domain-specific pretraining. A detailed performance comparison is shown in Table 2.

The baseline model, `roberta-large-mnli (0-Shot)`, performed the worst, with an accuracy of 29.15%. This suggests that zero-shot learning is not sufficient for this task, lacking the domain-specific adaptation needed for reliable results. Fine-tuning improved performance significantly, especially with models like `legal-bert-base-uncased (Head only)`, which achieved an accuracy of 53.35%. Fine-tuning allowed models to better adapt to specific task requirements, leading to better precision, recall, and overall effectiveness.

Applying Head-Tail truncation strategy to `legal-bert-base-uncased (Head-Tail)` and `roberta-large-mnli (Head-Tail)` improved accuracy and F1-scores, demonstrating that focusing on the core parts of the input can enhance model performance by reducing irrelevant information. Applying chunking strategy to `legal-bert-base-uncased (Chunking)` resulted in a notable performance boost with an accuracy of 62.26%. Chunking helps process longer documents more effectively, maintaining relevant context and improving results by reducing data loss.

MODEL	ACCURACY	PRECISION	RECALL	F1-SCORE
roberta-large-mnli (0-Shot)	29.15%	16.86%	14.61%	15.65%
legal-bert-base-uncased (Head only)	53.35%	49.79%	52.26%	51.00%
roberta-large-mnli (Head only)	53.67%	51.38%	52.12%	51.74%
legal-bert-base-uncased (Head-Tail)	54.76%	52.96%	54.73%	53.83%
roberta-large-mnli (Head-Tail)	54.84%	50.45%	54.82%	52.54%
legal-bert-base-uncased (Chunking)	62.26%	64.27%	62.18%	63.21%
longformer-base-4096	**68.35%**	**67.59%**	**66.75%**	**67.17%**

Table 2: Comparative Results on Test Data.



(a) SHAP

(b) LIME

Figure 3: Visual explanations using SHAP and LIME.

Longformer achieved the highest accuracy of 68.35% and an F1-score of 67.17%. Its ability to handle longer sequences and preserve context through efficient attention mechanisms allowed it to outperform other models, making it ideal for tasks involving large, complex documents.

In SHAP, the phrase `"I was referred to"` received a high positive contribution toward the model's prediction, suggesting it was a strong indicator for the predicted class `referred to`. In LIME, the word `"affirmed"` was given high weight for the prediction of the class `affirmed`, showing how individual tokens influence class-specific probabilities. Both tools demonstrate which terms the model relies on most for its legal reasoning.

5.1 Conclusion

This study provides a conclusive evidence on how fine-tuning, alongside input handling techniques like truncation and chunking, equips models to navigate the complexity of legal texts. Longformer emerged as the most effective, leveraging its extended attention span to retain critical context across lengthy documents.

As illustrated in Figure 3, the model's predictions were guided by legally salient phrases and words rather than superficial keywords. SHAP and LIME were used to identify specific phrases and words influencing model predictions, improving transparency and supporting informed evaluation of AI decisions in legal contexts.

6 Limitations

Legal text classification presents unique challenges that impact both model training and performance. This section highlights two primary challenges encountered during the study.

6.1 Computation Bottleneck

A key limitation of this work is the availability of computational resources. Experiments were conducted using limited access to A100 GPUs on USC's CARC (52 hours across 2 GPUs) and a local NVIDIA GeForce RTX 4060 GPU (64 hours). While sufficient for prototyping, these constraints limited our ability to scale to larger models and perform more extensive hyperparameter tuning.

6.2 Label Ambiguity

A minor challenge arises from the presence of 1,024 records where the `case_text` is repeated across different entries but assigned conflicting `case_outcome` labels. This label ambiguity introduces noise into the training process, as identical inputs correspond to different outputs, potentially confusing the model and affecting its ability to learn consistent patterns.

7 Future Scope

While ClearCase demonstrates strong performance, several enhancements can further improve fairness, and interpretability.

Class Balancing: Leveraging class-weighted and focal loss functions can address class imbalance by emphasizing minority classes during training.

Legal Data Augmentation: Techniques such as domain-aware text transformation and legal-specific back-translation can enhance data diversity while preserving legal semantics.

Explainability: Incorporating attention visualization and multi-level explanation systems can offer deeper insight for both technical and legal audiences.

References

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

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

Sarthak Jain and Byron C. Wallace. 2019. Attention is not explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Nut Limsopatham. 2021. Effectively leveraging BERT for legal document classification. In *Proceedings of the Natural Legal Language Processing Workshop (NLLP)*.

Scott M. Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Dimitris Mamakas, Ilias Chalkidis, Ion Androutsopoulos, and Nikolaos Aletras. 2022. Processing long legal documents with pre-trained transformers: Modding legalbert and longformer. In *Proceedings of the Natural Legal Language Processing Workshop (NLLP)*.

A. Mohankumar. 2023. Legal text classification dataset. Kaggle.

Marco T. Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Jesse Vig. 2019. BertViz: Visualize attention in NLP models. GitHub Repository.

A Best Models for Classification

To facilitate reproducibility and further research, we have made our final models available at the following link:
Fine-Tuned Models

The above repository best checkpoint models:

- RoBERTa (Head-Tail)
- LegalBERT (Head-Tail)
- LegalBERT (Chunking)

B SHAP and LIME Demo Files

- **SHAP**
- **LIME**