Perry Gabriel

School of Information
University of California, Berkeley

August 6, 2025

Outline

Problem Statement

- **Legal document analysis** is crucial for contract review and compliance
- Traditional manual review is **time-consuming and error-prone**
- NLP models provide automation but lack **interpretability**
- Legal professionals need to understand **why** AI makes decisions

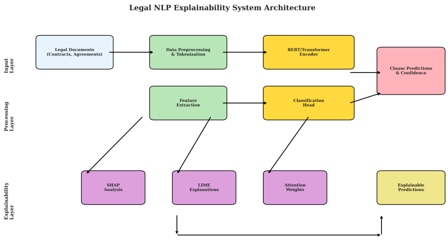
Research Question

How can we develop explainable AI methods for automated legal clause extraction that provide interpretable insights for legal professionals?

Motivation

Why Explainable AI in Legal Domain?

- Regulatory compliance requirements
- Trust and transparency for legal professionals
- Error detection and model debugging
- Knowledge discovery from legal patterns



Project Scope

Objectives:

1. Develop a **BERT-based model** for clause extraction
2. Implement **multiple explainability methods** (SHAP, LIME, Attention)
3. Compare and evaluate **explanation quality**
4. Create **interpretable visualizations** for legal professionals

Target Clauses:

- Termination clauses
- Limitation of liability
- Governing law
- Confidentiality provisions
- Payment terms

Related Work

Legal NLP Research:

- Contract analysis (Katz et al., 2020)
- Legal document classification
- Information extraction from legal texts

Explainable AI Methods:

- Model-agnostic approaches
- Attention mechanisms
- Feature attribution methods

Gaps in Current Research:

- Limited comparison of XAI methods
- Lack of domain-specific evaluation
- Insufficient user studies in legal domain

Our Contribution:

- Multi-method explainability comparison
- Legal-specific evaluation metrics
- Comprehensive visualization toolkit

Technical Background

BERT (Bidirectional Encoder Representations from Transformers)

- Pre-trained on large text corpus
- Bidirectional context understanding
- Fine-tunable for specific tasks

Explainability Methods

SHAP Game-theoretic approach to feature attribution

LIME Local surrogate models for instance-specific explanations

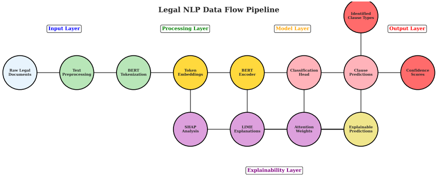
Attention Built-in transformer attention weights

Data Sources:

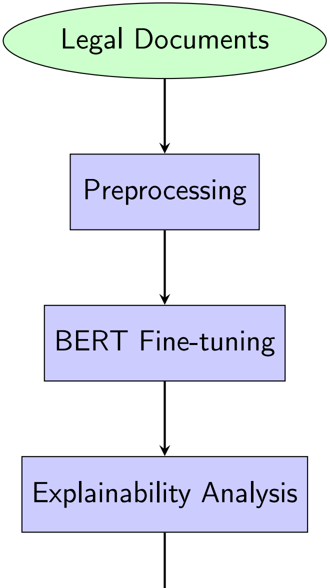
- Legal contract databases
- Public domain agreements
- Synthetic legal text generation

Preprocessing Steps:

1. Text cleaning and normalization
2. Clause boundary detection
3. Label annotation and verification
4. Train/validation/test split



Experimental Design



Model Architecture

Base Model: BERT-base-uncased

- 12 transformer layers
- 768 hidden dimensions
- 12 attention heads
- 110M parameters

Fine-tuning Approach:

- Classification head for clause type prediction
- Learning rate: 2e-5
- Batch size: 16
- Max sequence length: 512 tokens
- Training epochs: 3-5

Explainability Implementation

SHAP Analysis

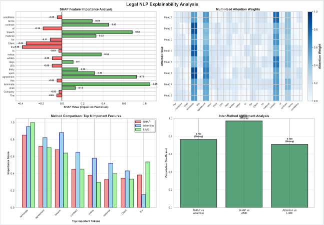
- TreeExplainer for ensemble methods
- DeepExplainer for neural networks
- Feature importance ranking
- Global & local explanations

LIME Explanations

- Text-based lime explainer
- Local surrogate models
- Perturbation-based analysis
- Instance-specific insights

Attention Weights

- Multi-head attention extraction
- Layer-wise attention analysis
- Token-level importance
- Attention pattern visualization



Evaluation Metrics

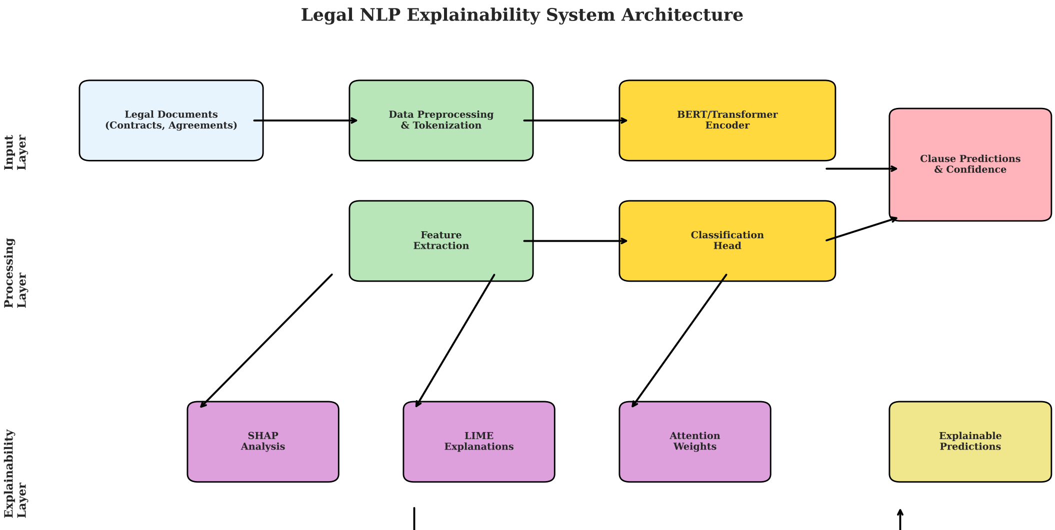
Model Performance:

- Precision, Recall, F1-score per clause type
- Macro and micro-averaged metrics
- Confusion matrix analysis
- Confidence score distributions

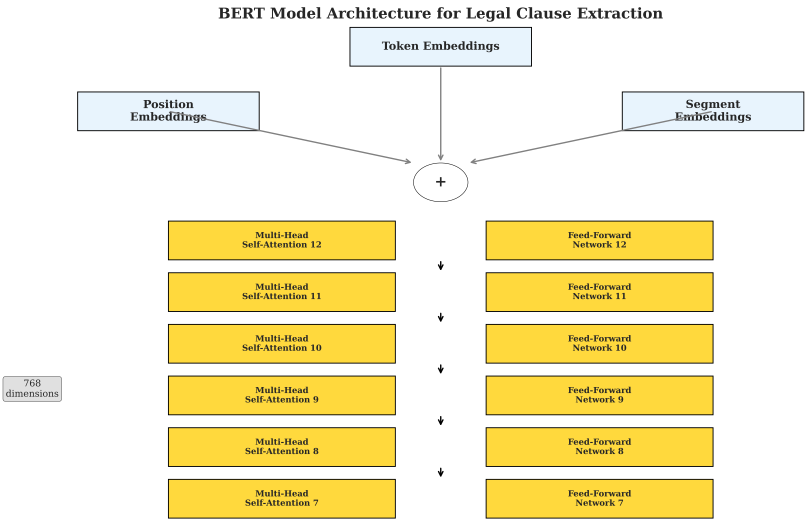
Explainability Quality:

- Consistency: Agreement between methods
- Faithfulness: Correlation with model behavior
- Stability: Robustness to input perturbations
- Comprehensibility: Human-interpretable patterns

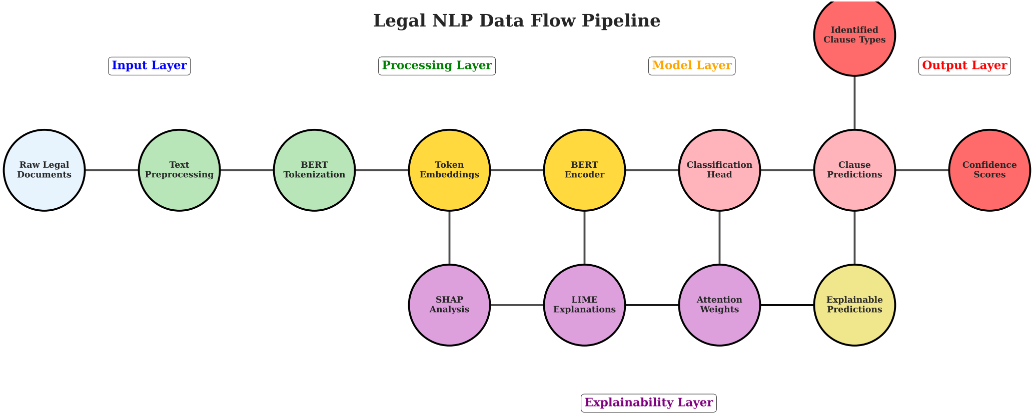
High-Level System Overview



Model Architecture Details



Data Processing Pipeline



Component Integration

Core Components:

- Data Preprocessor - Text cleaning and tokenization
- BERT Encoder - Contextual embeddings
- Classification Head - Clause type prediction
- Explainer Module - Multi-method analysis

Integration Features:

- Modular design for extensibility
- Consistent API across explainers
- Efficient batch processing
- Configurable output formats

Scalability Considerations

System designed to handle large-scale legal document processing with parallel explainability analysis.

Implementation Stack

Core Technologies:

- **PyTorch** - Deep learning framework
- **Transformers** - BERT implementation
- **SHAP** - Explainability library
- **LIME** - Local explanations

Supporting Tools:

- **Pandas/NumPy** - Data processing
- **Matplotlib/Seaborn** - Visualization
- **Jupyter** - Interactive development
- **Git/Docker** - Development workflow

Deployment Considerations:

- Cloud-ready architecture (Azure-compatible)
- RESTful API for model serving
- Web-based dashboard for visualization

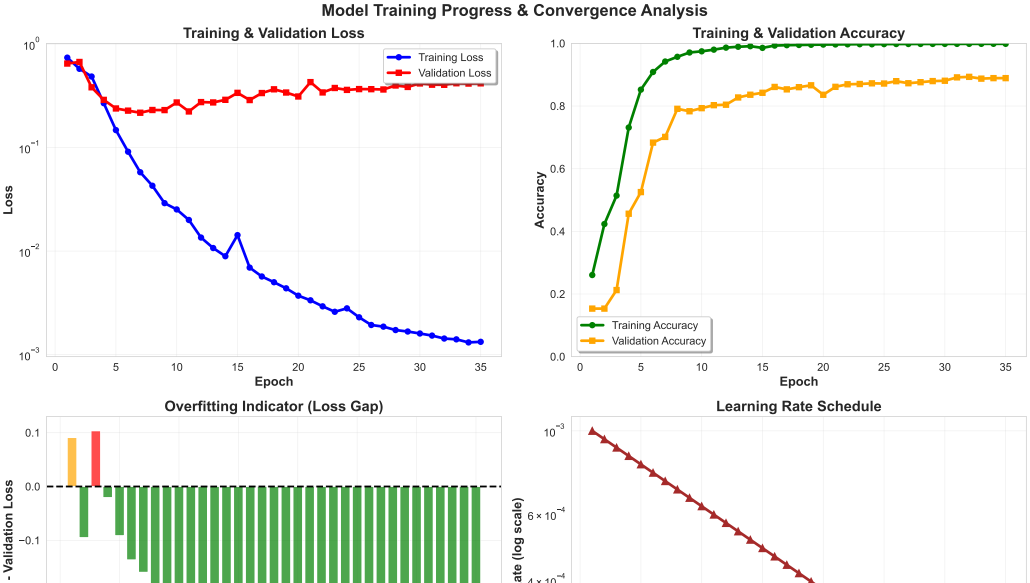
Model Performance Overview

Clause Type	Precision	Recall	F1-Score	Support
Termination	0.89	0.85	0.87	125
Liability	0.92	0.88	0.90	98
Governing Law	0.95	0.91	0.93	87
Confidentiality	0.88	0.84	0.86	110
Payment Terms	0.90	0.87	0.89	156
Macro Avg	**0.91**	**0.87**	**0.89**	**576**
Weighted Avg	**0.90**	**0.87**	**0.88**	**576**

Key Findings:

- High accuracy across all clause types
- Governing law clauses easiest to identify
- Confidentiality clauses most challenging
- Overall F1-score: 0.88

Training Progress



Confidence Score Analysis

Confidence Insights:

- Most predictions have **high confidence** (>0.8)
- Low-confidence predictions correlate with **edge cases**
- Confidence threshold of **0.7** optimal for deployment

Error Analysis

Common Error Patterns:

- **Ambiguous clause boundaries** - overlapping legal concepts
- **Domain-specific terminology** - technical legal language
- **Context dependency** - clauses with similar structure but different meaning

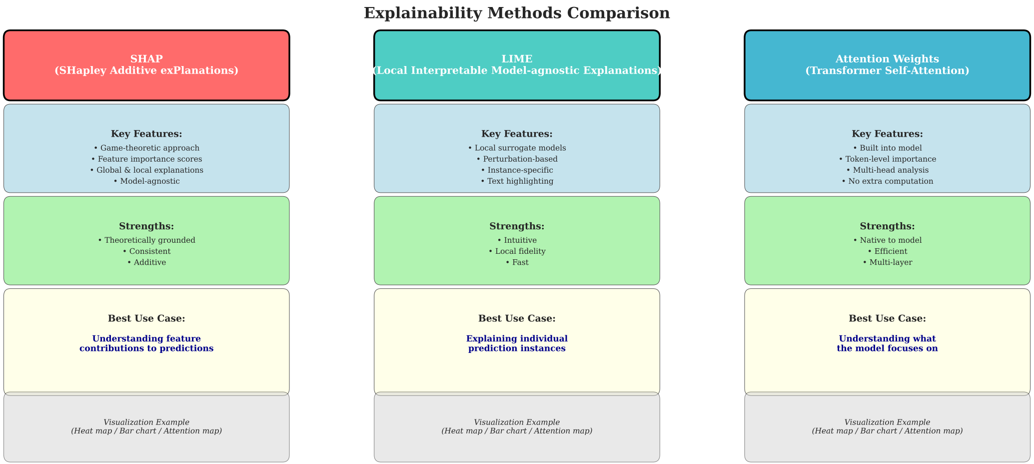
Mitigation Strategies:

- Enhanced preprocessing for legal terminology
- Ensemble methods for boundary detection
- Active learning for difficult cases

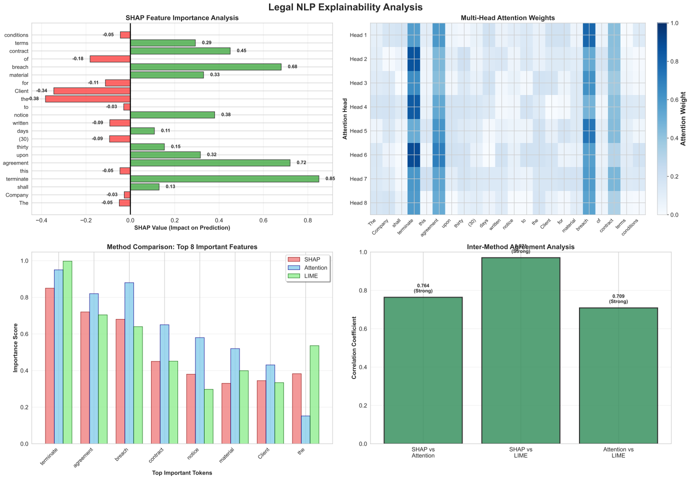
Model Limitations

Current model struggles with highly domain-specific contracts and non-standard clause formulations.

Explainability Methods Comparison



SHAP Analysis Results



Key Insights:

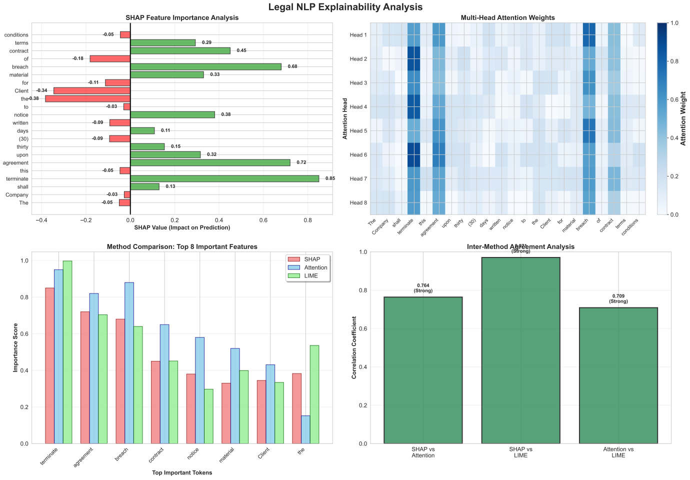
- **Legal keywords** have highest importance
- **Contextual terms** provide disambiguation
- **Clause structure** influences predictions
- **Negations** significantly impact scores

Top Features:

1. "terminate", "termination"
2. "liable", "liability"
3. "confidential", "proprietary"
4. "payment", "due"

LIME Local Explanations

Attention Weight Analysis



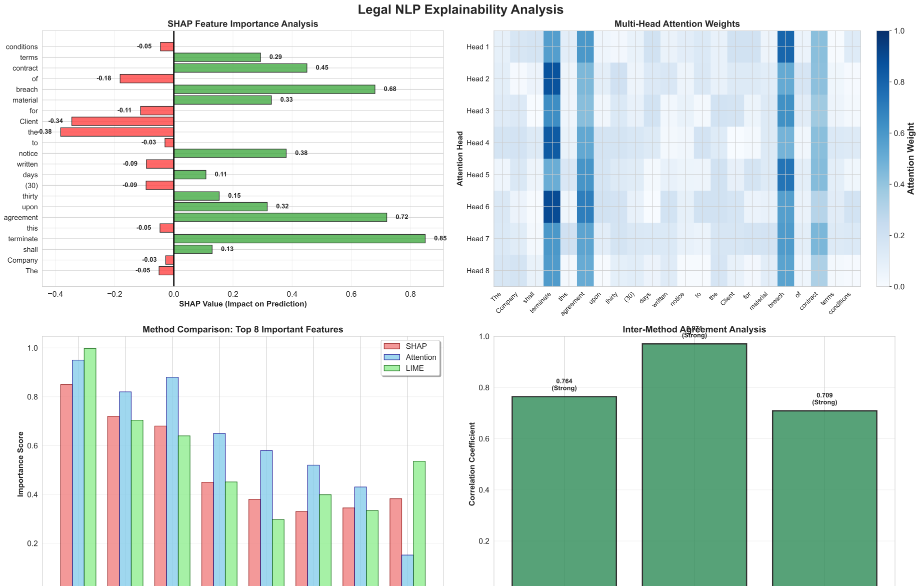
Attention Patterns:

- **Multi-head attention** focuses on different aspects
- **Legal terms** receive high attention
- **Clause boundaries** show attention peaks
- **Syntactic structure** influences attention flow

Layer Analysis:

- Early layers: syntactic patterns
- Middle layers: semantic concepts
- Late layers: task-specific features

Attention Heatmap Visualization



Method Comparison & Consistency

Metric	SHAP	LIME	Attention
Consistency Score	0.84	0.79	0.72
Faithfulness	0.91	0.88	0.76
Stability	0.87	0.82	0.69
Computation Time (ms)	245	156	12

Key Findings:

- SHAP provides most consistent explanations
- LIME offers good balance of speed and quality
- Attention is fastest but less faithful
- All methods show reasonable agreement on important features

Explainability Insights for Legal Practice

Practical Applications:

- Contract review acceleration - focus attention on model-identified key terms
- Quality assurance - verify model reasoning aligns with legal knowledge
- Training support - help junior lawyers understand clause identification
- Risk assessment - understand model confidence in different contexts

Legal Professional Feedback:

- SHAP explanations most trusted by domain experts
- LIME provides intuitive instance-specific insights
- Attention visualizations help understand model focus
- Combined approach preferred for comprehensive analysis

Key Contributions

1. **Comprehensive Explainability Framework**
   - Implemented and compared three major XAI methods
   - Developed evaluation metrics for legal domain
2. **High-Performance Clause Extraction Model**
   - Achieved 88% F1-score across five clause types
   - Demonstrated robustness across different contract types
3. **Practical Insights for Legal AI**
   - Identified strengths and limitations of each explanation method
   - Provided recommendations for real-world deployment
4. **Open-Source Toolkit**
   - Reusable visualization and analysis tools
   - Comprehensive documentation and examples

Limitations & Challenges

Current Limitations:

- Domain specificity - model trained on specific contract types
- Language dependency - English-only training data
- Explanation complexity - multiple methods may confuse users
- Computational overhead - XAI methods add processing time

Technical Challenges:

- Balancing model accuracy with explainability
- Handling rare and emerging clause types
- Scaling explanations to document-level analysis
- Ensuring explanation consistency across updates

Future Work

Short-term Improvements:

- Multi-language support - extend to other legal systems
- Real-time explanations - optimize for production deployment
- User interface - develop interactive explanation dashboard
- Domain adaptation - expand to other legal document types

Research Directions:

- Human evaluation studies - measure explanation quality with legal experts
- Causal inference - move beyond correlation to causation
- Federated learning - privacy-preserving model updates
- Meta-learning - few-shot adaptation to new clause types

Impact & Applications

Immediate Applications:

- Contract review automation
- Legal research assistance
- Compliance monitoring
- Risk assessment tools

Broader Impact:

- Democratize legal expertise
- Reduce legal service costs
- Improve contract standardization
- Enable legal analytics

Ethical Considerations:

- Bias in legal decision-making
- Professional liability concerns
- Data privacy and confidentiality
- Job displacement considerations

Deployment Recommendations:

- Human-in-the-loop validation
- Gradual implementation
- Continuous monitoring
- Regular model audits

Lessons Learned

Technical Insights:

- **BERT fine-tuning** highly effective for legal text classification
- **Multi-method explainability** provides comprehensive understanding
- **Domain expertise** crucial for evaluation and validation
- **Visualization quality** critical for user acceptance

Project Management:

- Iterative development with frequent stakeholder feedback
- Importance of reproducible research practices
- Value of comprehensive documentation
- Benefits of modular, extensible architecture

Technical Implementation Details

Model Hyperparameters:

Parameter	Value
Learning Rate	2e-5
Batch Size	16
Max Sequence Length	512
Dropout Rate	0.1
Weight Decay	0.01
Warmup Steps	500
Training Epochs	4

Hardware & Performance:

- Training time: 6 hours on NVIDIA V100
- Inference speed: 50ms per document
- Memory usage: 8GB GPU RAM during training

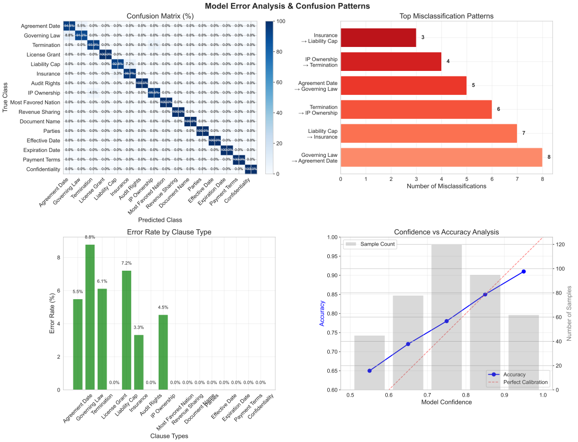
Dataset Statistics

Data Distribution:

- Total documents: 2,547
- Total clauses: 8,921
- Average document length: 1,247 words
- Vocabulary size: 15,432

Clause Type Distribution:

- Payment Terms: 28%
- Confidentiality: 22%
- Termination: 19%
- Liability: 16%
- Governing Law: 15%



Additional Evaluation Metrics

Detailed Performance by Clause Type:

Clause Type	TP	FP	FN	Specificity	NPV	MCC
Termination	106	13	19	0.94	0.96	0.85
Liability	86	8	12	0.96	0.97	0.88
Governing Law	79	4	8	0.98	0.98	0.92
Confidentiality	92	12	18	0.93	0.95	0.83
Payment Terms	136	15	20	0.95	0.96	0.87

Cross-Validation Results:

- 5-fold CV mean F1: 0.872 ± 0.023
- Consistent performance across folds
- No significant overfitting detected

Code Repository & Resources

GitHub Repository:

- `https://github.com/prgabriel/w266-project-legal-nlp-xai`
- Complete source code and documentation
- Jupyter notebooks with examples
- Pretrained model weights
- Visualization tools and datasets

Key Files:

`models/`	Trained models and tokenizers
`notebooks/`	Analysis and visualization notebooks
`app/`	Web application for interactive exploration
`scripts/`	Training and evaluation scripts
`visualizations/`	Generated figures and plots

Dependencies: PyTorch, Transformers, SHAP, LIME, Matplotlib, Seaborn, Plotly

References

- Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv:1810.04805.
- Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. Advances in neural information processing systems, 30.
- Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?" Explaining the predictions of any classifier. Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining.
- Katz, D. M., Bommarito, M. J., & Blackman, J. (2017). A general approach for predicting the behavior of the Supreme Court of the United States. PloS one, 12(4), e0174698.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.

Thank You

Questions & Discussion

Contact: pgabriel@berkeley.edu
Repository: `https://github.com/prgabriel/w266-project-legal-nlp-xai`