



Perry Gabriel

School of Information  
University of California, Berkeley

August 6, 2025

# Outline

- 1 Introduction
- 2 Background
- 3 Methodology
- 4 System Architecture
- 5 Results
- 6 Explainability Analysis
- 7 Conclusion
- 8 Appendix

# Problem Statement

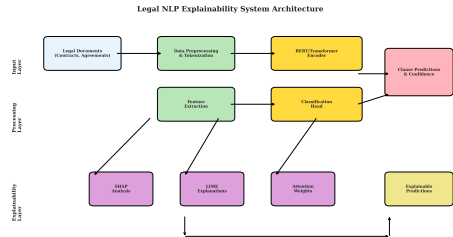
- Legal document analysis is crucial for contract review and compliance
- Traditional manual review is time-consuming and error-prone
- NLP models provide automation but lack interpretability
- Legal professionals need to understand why AI makes decisions

## Research Question

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

## 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



## 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

## 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

## 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

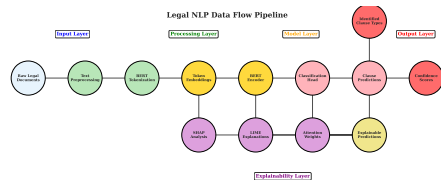
# Dataset Overview

## Data Sources:

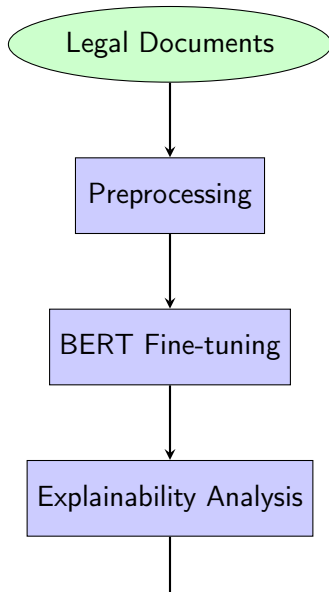
- Legal contract databases
- Public domain agreements
- Synthetic legal text generation

## Preprocessing Steps:

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







## **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

## extbfSHAP Analysis:

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

## extbfLIME Explanations:

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

## extbfAttention Weights:

- Multi-head attention analysis
- Token-level importance
- Layer-wise attention patterns
- Visualization of focus areas

## Model Performance:

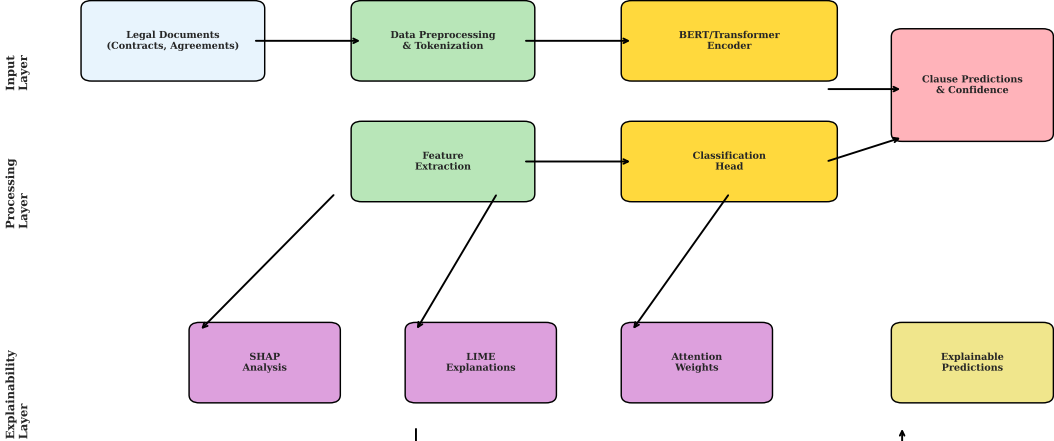
- Precision, Recall, F1-score per clause type
- Macro and micro-averaged metrics
- Confusion matrix analysis
- Confidence score distributions

## Explainability Quality:

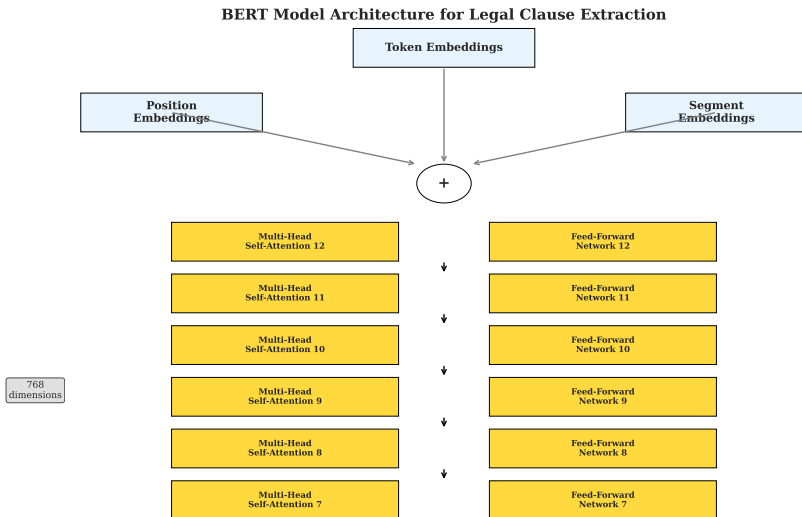
- **Consistency:** Agreement between methods
- **Faithfulness:** Correlation with model behavior
- **Stability:** Robustness to input perturbations
- **Comprehensibility:** Human-interpretable patterns

# High-Level System Overview

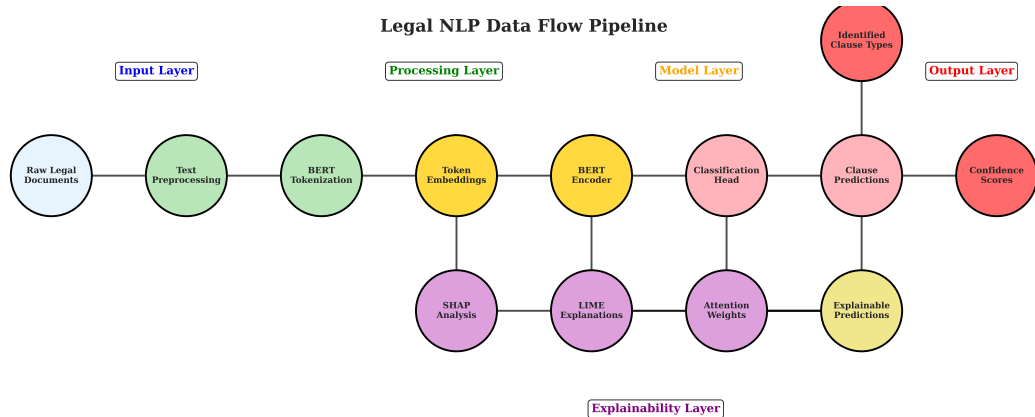
Legal NLP Explainability System Architecture



# 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:

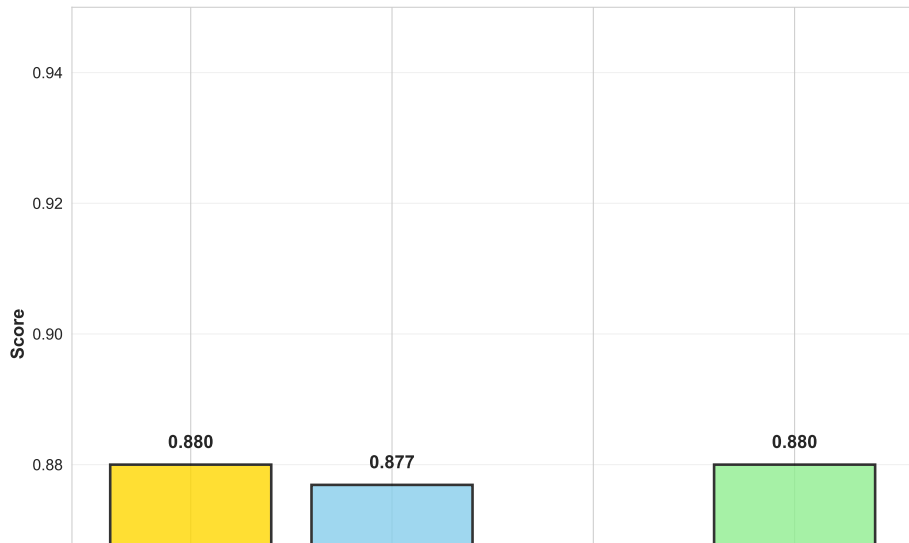
- **Pandas/NumPy** - Data processing
- **Matplotlib/Seaborn** - Visualization
- **Jupyter** - Interactive development
- **Git/Docker** - Development workflow

## Deployment Considerations:

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

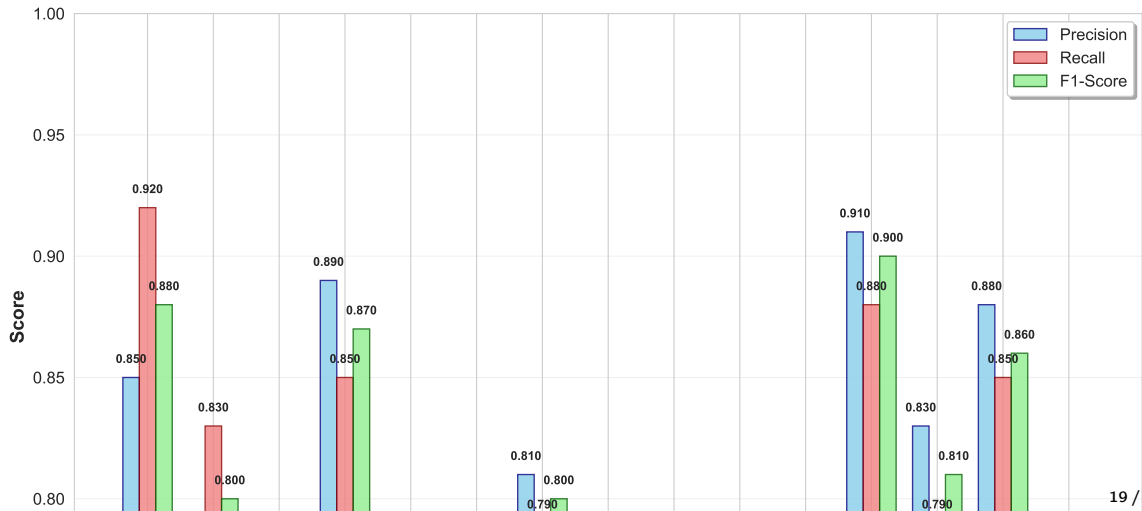
# Overall Model Performance

Overall Model Performance Summary



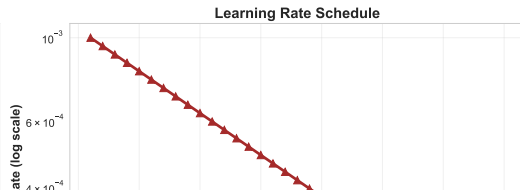
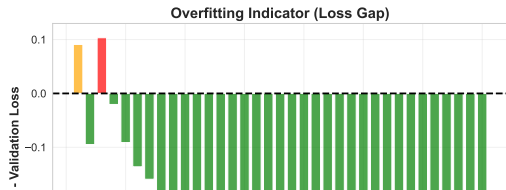
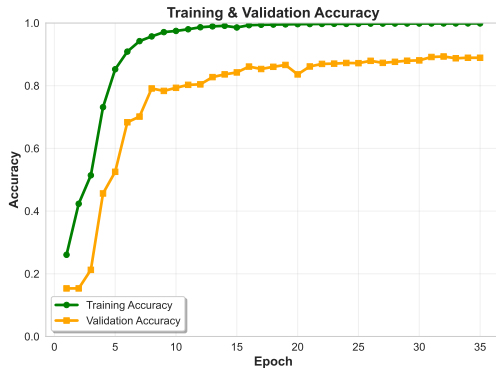
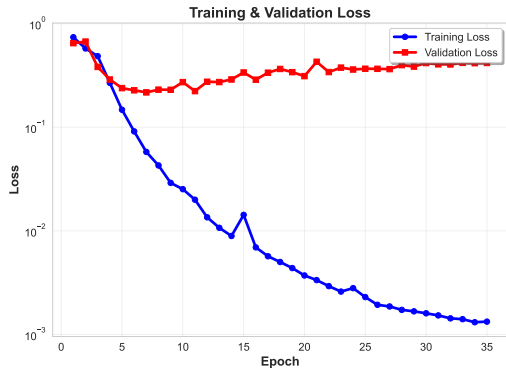
# Performance by Clause Type

Performance Metrics by Clause Type



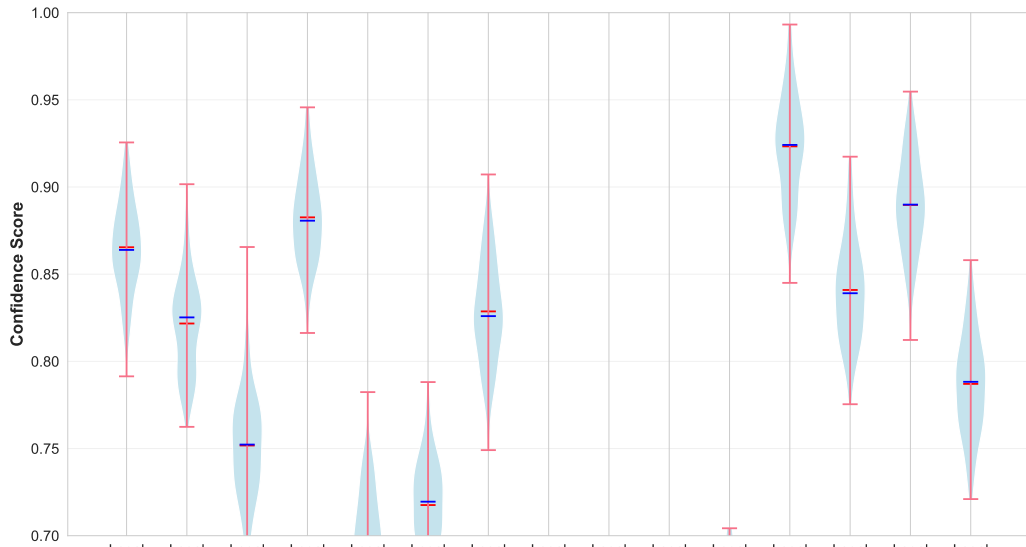
# Training Progress

Model Training Progress & Convergence Analysis



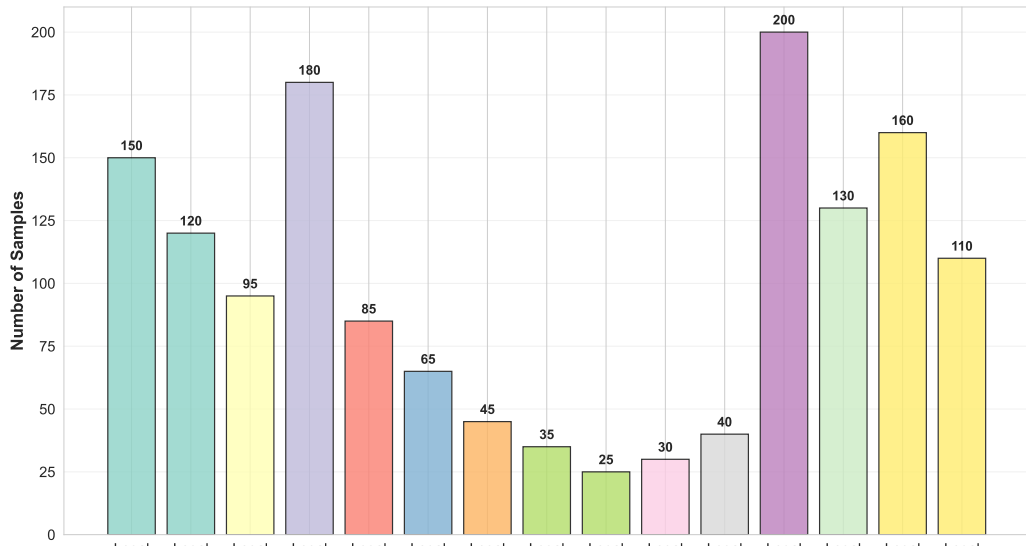
# Confidence Score Analysis

Prediction Confidence Distribution



# Dataset Distribution

Dataset Distribution by Clause Type



# 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:

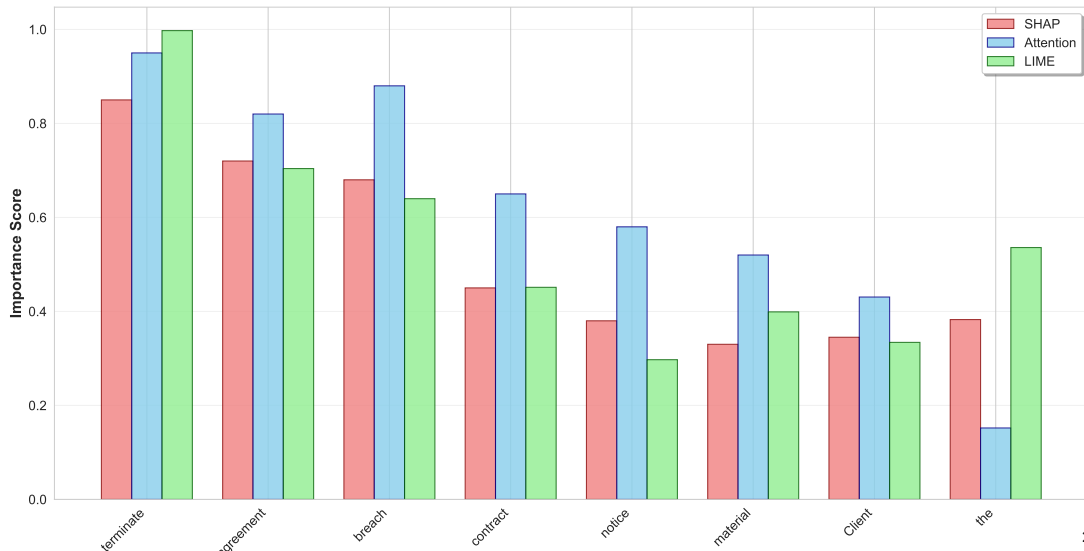
- Enhanced preprocessing for legal terminology
- Ensemble methods for boundary detection
- Active learning for difficult cases

## Model Limitations

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

# Feature Importance Method Comparison

Method Comparison: Top Important Features





# Explainability Methods Comparison

## Explainability Methods Comparison

**SHAP**  
(SHapley Additive exPlanations)

**Key Features:**

- Game-theoretic approach
- Feature importance scores
- Global & local explanations
  - Model-agnostic

**Strengths:**

- Theoretically grounded
  - Consistent
  - Additive

**Best Use Case:**  
**Understanding feature contributions to predictions**

Visualization Example  
(Heat map / Bar chart / Attention map)

**LIME**  
(Local Interpretable Model-agnostic Explanations)

**Key Features:**

- Local surrogate models
- Perturbation-based
- Instance-specific
- Text highlighting

**Strengths:**

- Intuitive
- Local fidelity
  - Fast

**Best Use Case:**  
**Explaining individual prediction instances**

Visualization Example  
(Heat map / Bar chart / Attention map)

**Attention Weights**  
(Transformer Self-Attention)

**Key Features:**

- Built into model
- Token-level importance
- Multi-head analysis
- No extra computation

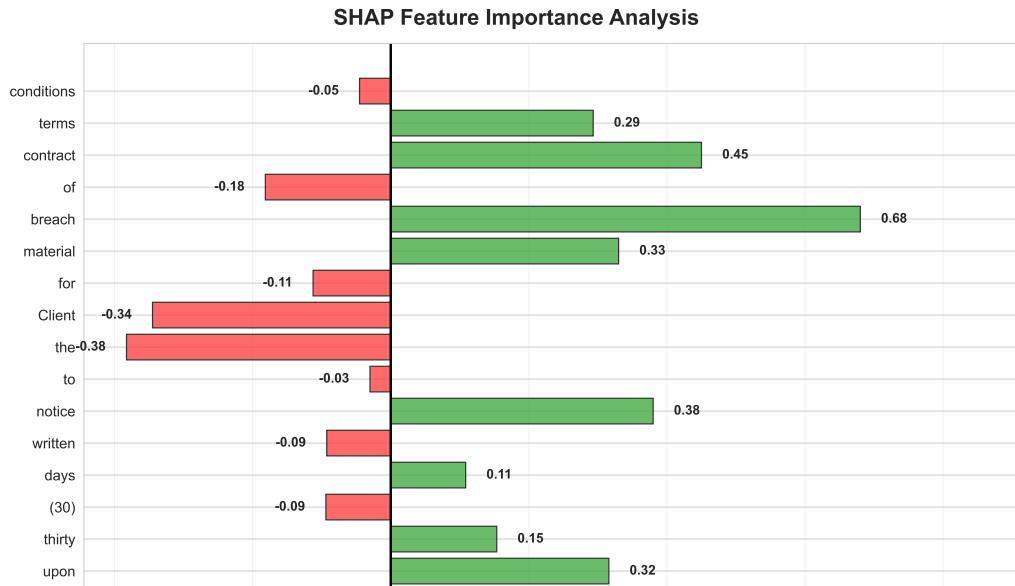
**Strengths:**

- Native to model
  - Efficient
  - Multi-layer

**Best Use Case:**  
**Understanding what the model focuses on**

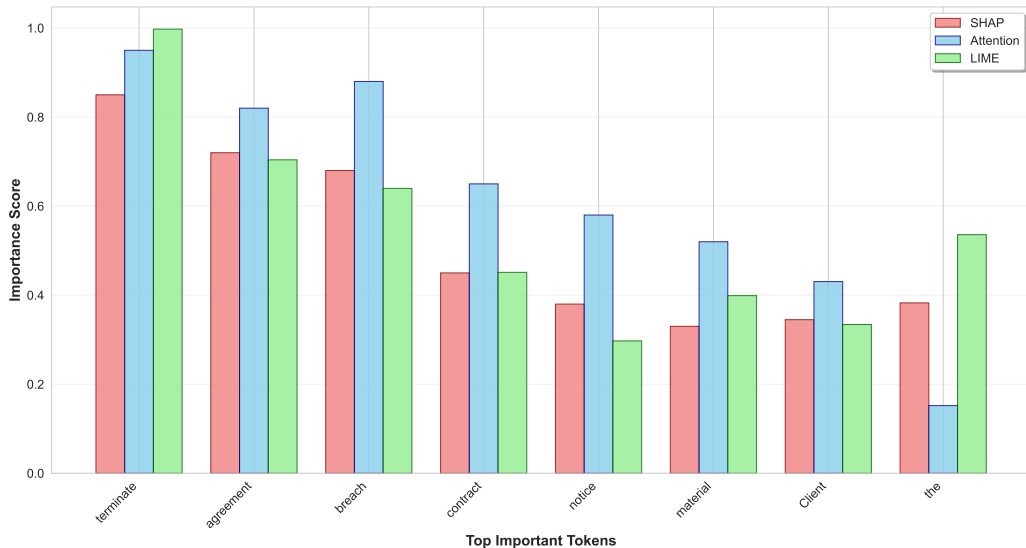
Visualization Example  
(Heat map / Bar chart / Attention map)

# SHAP Feature Importance Analysis

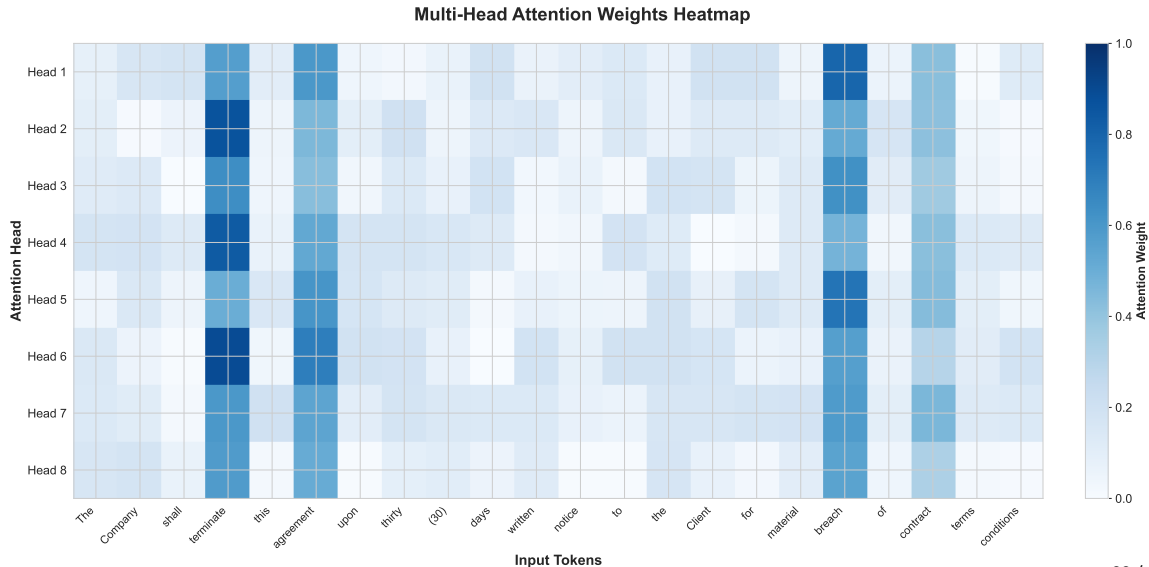


# LIME Local Explanations

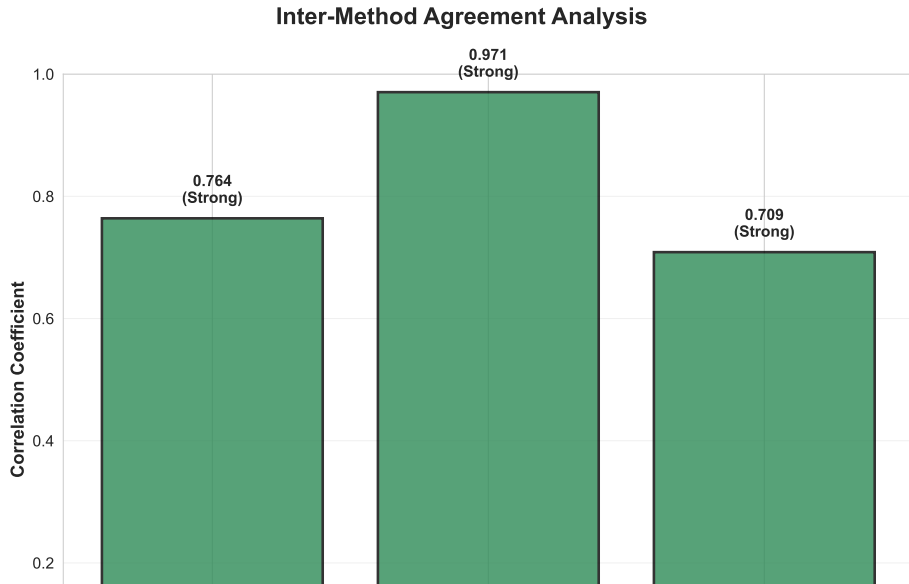
Method Comparison: Top Important Features



# Multi-Head Attention Analysis



# Inter-Method Agreement Analysis



## 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

## 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

## 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

## 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

## 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

## 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 Distribution Analysis

extbfKey  
Statis-  
tics:

• Total  
sam-  
ples:  
1,350

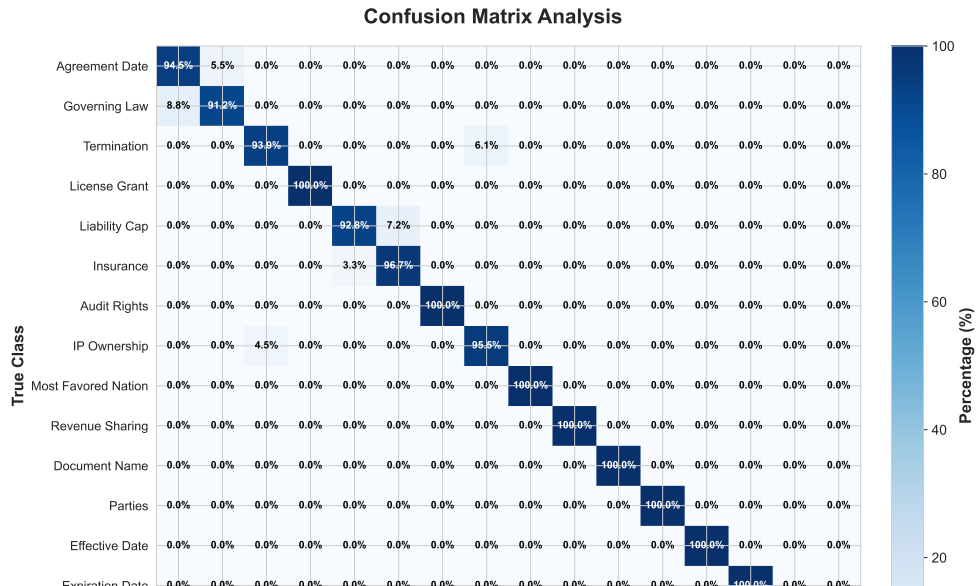
• Payment  
Terms:  
20%

twidhtwidth

• Termination:  
18%

twidhtwidth

# Confusion Matrix Analysis



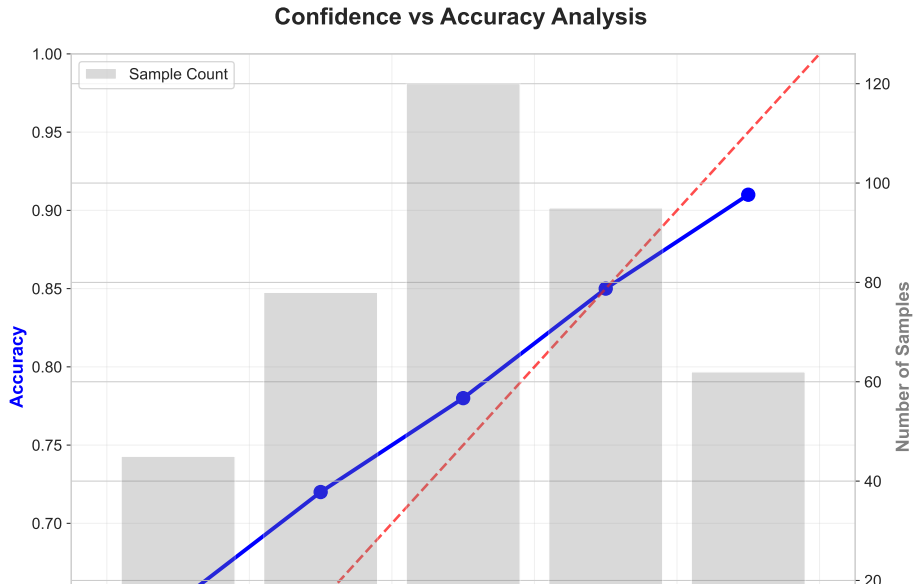
twidhtwidth

extbfCommon Misclassifications:

- **Governing Law** ↔ **Agreement Date** confusion reflects similar legal language
- **Liability** ↔ **Insurance** overlap due to risk-related terminology
- **IP Ownership** ↔ **Termination** shows boundary challenges
- Error patterns guide targeted **model improvements**



# Model Calibration Analysis



# 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:  $0.872 \pm 0.023$
- 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`