Automated Legal Clause Checker: Machine Learning Final Project Report

Devadarshini Pazhanivel Thenmozhi, Sharon Jennifer Justin Devaraj
Khoury College of Computer Sciences
Northeastern University, Boston, USA
pazhanivelthenmozh.d@northeastern.edu, justindevaraj.s@northeastern.edu
GITHUB

***Abstract*—Reviewing contracts is an essential part of business dealings, ensuring agreements are both compliant and legally sound. However, manual clause checking is expensive, time-consuming, and prone to human error particularly with lengthy documents. This project presents an automated framework for detecting and classifying legal clauses in commercial contracts using the CUAD (Contract Understanding Atticus Dataset). We implement and compare both classical machine learning models (Logistic Regression and Random Forest) and transformer-based models (BERT, RoBERTa, and LegalBERT) for multi-label clause classification across 41 categories. Our approach addresses significant class imbalance through weighted loss functions, applies threshold optimization for improved predictions, and provides comprehensive evaluation using precision, recall, and F1-score metrics. The system processes 510 commercial contracts containing over 13,000 labeled clauses, with train,validation,test splits of 326,82,102 samples respectively. Results demonstrate that transformer models significantly outperform classical baselines. The automated system provides interpretable outputs through visualizations and clause-level predictions, demonstrating the potential of Natural Language Processing to streamline legal compliance workflows.**

***Index Terms*—Multi-label classification, CUAD, Logistic Regression, Random Forest, BERT, RoBERTa, LegalBERT, Transformer, Clause detection.**

I. Introduction

Contract analysis represents a critical component of modern business operations, particularly in high-stakes transactions involving mergers, acquisitions, licensing agreements, and corporate partnerships. The complexity of commercial legal documents which often span hundreds of pages and contain numerous interconnected clauses, necessitates thorough review to ensure legal compliance, identify potential risks, and verify contractual obligations. Traditional manual contract review processes are limited by human capacity constraints and the substantial time and financial resources required for expert legal analysis.

The legal industry faces mounting pressure to adopt automated solutions that can enhance efficiency without compromising accuracy. Commercial contracts typically contain dozens of critical clause types, including but not limited to non-compete agreements, governing law provisions, intellectual property assignments, termination conditions, and liability limitations. The manual identification and extraction of these clauses requires specialized legal expertise and considerable time investment, often resulting in processing delays that can impact business decision-making timelines.

Recent advances in Natural Language Processing (NLP) and machine learning have opened new innovations for automating legal document analysis. The emergence of large-scale legal datasets, such as the Contract Understanding Atticus Dataset (CUAD), has enabled researchers to develop and evaluate automated clause detection systems using real-world commercial contracts. These developments coincide with significant improvements in transformer-based language models, which have demonstrated superior performance across various NLP tasks compared to traditional machine learning approaches.

The application of automated clause detection systems offers substantial benefits for legal practitioners, business analysts, and compliance teams. Such systems can significantly reduce the time required for initial contract review, provide consistent clause identification across document sets, and flag potential risks or missing provisions that might be overlooked during manual review. Furthermore, automated systems can process large volumes of contracts simultaneously, enabling organizations to conduct comprehensive due diligence exercises more efficiently.

This project addresses the challenge of automated legal clause detection and classification by implementing and comparing multiple machine learning approaches on the CUAD dataset. We investigate both classical machine learning methods and transformer models to determine the most effective approach for multi-label clause classification. Our research contributes to the growing body of work in legal NLP by providing a comprehensive comparison of different modeling approaches and demonstrating the practical applicability of automated clause detection systems in commercial contract analysis scenarios.

II. Data

This study utilizes the **Contract Understanding Atticus Dataset (CUAD)**, a publicly available corpus of commercial legal contracts curated for clause classification tasks. The dataset was collected from the **U.S. Securities and Exchange Commission's (SEC) EDGAR** database, ensuring high-quality, real-world legal documents.

- **Number of contracts:** 510
- **Total annotated clauses:** Over 13,000

- **Number of clause categories:** 41 (e.g., *Non-Compete, Governing Law, Non-Disclosure Agreement (NDA), Termination, License Grant, Audit Rights*).
- **Annotation format:** Each contract is divided into paragraphs/clauses, with each segment assigned one or more labels.
- **Label structure:** Includes both binary labels (presence/absence of a clause) and factual answer spans (e.g., effective dates, parties, renewal terms).

CUAD is inherently a **multi-label** dataset, as a single clause may correspond to multiple legal categories. Additionally, the dataset exhibits **class imbalance**, with certain clauses (e.g., *Document Name, Parties*) occurring in nearly all contracts, while others (e.g., *Source Code Escrow, Price Restrictions*) appear in fewer than 3% of cases.

The dataset's diversity and detailed annotations make it well-suited for evaluating both **classical machine learning** and **transformer-based** models in multi-label legal clause classification.

III. Related Work

A. Legal Document Classification and NLP Applications

D.Katz et al. (2017), "A General Approach for Predicting the Behavior of the Supreme Court of the United States" This study demonstrated the application of machine learning techniques to predict Supreme Court case outcomes using textual features from legal briefs and oral arguments. The authors employed Random Forest classifiers and achieved prediction accuracy of 70.2%, establishing foundational work in computational legal analysis and showing that automated systems could match expert-level performance in specific legal prediction tasks.

I.Chalkidis et al. (2020), "LEGAL-BERT: The Muppets Straight Out of Law School" This paper introduced Legal-BERT, a domain-specific transformer model pre-trained on legal corpora including case law, contracts, and legislation. The model demonstrated significant improvements over general-purpose BERT on legal NLP tasks, achieving F1-scores 3-5% higher than baseline models. The work established the importance of domain-specific pre-training for legal language understanding and provided benchmarks for legal document classification tasks.

B. Contract Analysis and Automated Clause Detection

D.Hendrycks et al. (2021), "CUAD: An Expert-Annotated NLP Dataset for Legal Contract Review" This study introduced the Contract Understanding Atticus Dataset containing 510 commercial contracts with over 13,000 labeled clauses across 41 categories. The authors established baseline performance using reading comprehension models adapted for clause extraction, achieving F1-scores ranging from 0.2 to 0.9 across different clause types. The work provided the first large-scale benchmark for automated contract analysis and highlighted challenges in handling class imbalance and rare clause types.

K.Shanker et al. (2025), "LegalBERT for Small Business Contract Analysis: An Advanced NLP Tool for Identifying High Risk Clauses" This work developed a contract analysis tool for small businesses using the CUAD dataset and LegalBERT to extract legal clauses. They addressed class imbalance with oversampling and fine-tuned LegalBERT, achieving an F1-score of 0.72. The study highlighted challenges with niche clauses, semantic overlaps, and dataset bias toward corporate contracts, while demonstrating the practical utility of transformer models for automated clause detection in business contexts.

C. Multi-label Classification and Class Imbalance

R.Zhang et al. (2020), "Multi-Label Legal Document Classification via Hierarchical Attention Networks" This work proposed a hierarchical attention-based neural architecture for classifying legal documents into multiple categories simultaneously. The model leveraged both word-level and sentence-level attention to capture context and improve classification performance. Experiments on benchmark legal datasets demonstrated significant improvements over traditional flat classification approaches, particularly in handling long legal contracts.

Y.Lui et al. (2019), "Class-Balanced Loss Based on Effective Number of Samples" This work developed class-balanced loss functions that incorporate effective sample numbers to handle datasets with extreme class imbalance. The approach calculated per-class weights based on sample overlap characteristics, demonstrating improved performance on long-tail distribution datasets. The technique has proven particularly effective for legal document classification where clause frequency follows power-law distributions.

D. Transformer Models in Legal Domain

J.Devlin et al. (2019), "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding" This foundational work introduced BERT's bidirectional attention mechanism, which proved particularly effective for capturing long-range dependencies crucial in legal document understanding. The model achieved state-of-the-art performance across multiple NLP tasks and established the foundation for subsequent legal domain adaptations. The bidirectional nature of BERT enables better understanding of complex legal sentence structures and cross-references.

IV. IMPLEMENTATION

A. Data Preprocessing and Feature Engineering

The CUAD dataset preprocessing pipeline transformed 510 commercial contracts into a structured format suitable for multi-label clause classification. The initial data loading process parsed the hierarchical JSON structure, flattening 20,910 question-answer pairs into tabular format while preserving document-level relationships and clause annotations.

Text preprocessing addressed encoding inconsistencies and formatting irregularities. Unicode normalization (NFKC) standardized character representations, while whitespace regularization replaced multiple consecutive spaces and standardized line break patterns. Special characters and tab sequences were normalized to ensure consistent tokenization behavior across different text processing frameworks.

Class imbalance analysis revealed severe distribution skew across the 41 clause categories. Common provisions such as "Document Name" appeared in 100% of contracts (510 instances), while specialized clauses like "Source Code Escrow" occurred in only 2.5% of documents (13 instances). This imbalance ratio of approximately 40:1 between most and least frequent clause types necessitated specialized handling techniques throughout the modeling pipeline.

The preprocessing framework implemented document-level splitting to prevent data leakage, ensuring that clauses from identical contracts remained within single partitions. This approach maintained realistic evaluation scenarios while preserving the natural multi-label structure of legal documents. The final dataset split allocated 326 documents for training, 82 for validation, and 102 for testing maintaining proportional clause distribution across all partitions.

Feature correlation analysis identified moderate correlations between semantically related clause types, particularly temporal provisions (Effective Date, Expiration Date, Renewal Term) and risk allocation clauses (Cap on Liability, Insurance, Indemnification). These correlations validated the multi-label approach rather than treating clause detection as independent binary classification tasks.

B. Classical Machine Learning Implementation

The classical machine learning pipeline employed TF-IDF vectorization optimized for legal document analysis. Feature extraction utilized unigram and bigram combinations (n-gram range 1-2) to capture both individual legal terms and common legal phrases prevalent in commercial contracts. Vocabulary limitation to 15,000 features balanced model complexity with computational efficiency while preserving discriminative legal terminology.

Document frequency filtering eliminated noise through minimum occurrence thresholds (`min_df=2`) and maximum document frequency limits (`max_df=0.95`). Sublinear term frequency scaling reduced the dominance of highly frequent terms, preventing common words from overshadowing specialized legal vocabulary critical for clause identification.

Logistic Regression implementation utilized L2 regularization with increased iteration limits (`max_iter=1000`) to ensure convergence on high-dimensional TF-IDF feature spaces. The `MultiOutputClassifier` framework trained independent binary classifiers for each clause category, enabling specialized pattern learning while maintaining computational tractability for the 41-class multi-label problem.

Random Forest configuration employed ensemble parameters tuned for legal text classification: 80 estimators provided sufficient diversity while maintaining reasonable training time, maximum depth of 8 prevented overfitting on training patterns, and minimum samples per split (8) and leaf (3) ensured statistical significance of decision boundaries. Bootstrap sampling utilized 75% of training data per estimator to enhance ensemble robustness.

C. Transformer Model Architecture and Fine-tuning

Transformer implementation leveraged pre-trained language models from Hugging Face's transformers library, including BERT-base-uncased, RoBERTa-base, and LegalBERT. Each model underwent task-specific configuration with modified classification heads featuring 41-dimensional output layers corresponding to clause categories. The `problem_type` parameter was set to "multi_label_classification" to ensure appropriate loss computation and evaluation metrics.

Tokenization employed model-specific vocabularies with maximum sequence length limited to 512 tokens, balancing legal document complexity with computational constraints. The tokenization strategy included truncation for lengthy contracts and padding for shorter texts, ensuring consistent input dimensions across variable-length legal documents.

Learning rate optimization was performed. Batch sizes were constrained to 8 samples due to GPU memory limitations imposed by long legal document sequences. Gradient clipping with maximum norm 1.0 prevented gradient explosion during training on complex legal text patterns.

Training infrastructure incorporated early stopping mechanisms, monitoring F1 scores with patience set to set number epochs. This configuration balanced convergence assurance with overfitting prevention, while model checkpoints preserved optimal weights corresponding to peak validation performance for consistent evaluation.

D. Advanced Loss Functions and Class Imbalance Mitigation

The severe class imbalance required specialized loss function implementations beyond standard cross-entropy approaches. Weighted Binary Cross-Entropy employed per-class positive weights calculated using the formula $(N - P_i)/P_i$, where N represents total training samples and P_i represents positive instances for class i. These weights ranged from 1.0 for common clauses to 64.2 for the rarest categories.

Class-Balanced Focal Loss implementation addressed both class imbalance and hard example mining simultaneously. The focal loss incorporated gamma parameter (2.0) to emphasize difficult examples while beta parameter (0.9999) adjusted effective sample numbers based on class frequency. The loss function dynamically weighted samples based on prediction confidence, reducing the contribution of easily classified examples while emphasizing challenging cases.

Sample-level weighting strategies assigned higher importance to contracts containing rare clause types during training optimization. Row weights were calculated as the mean positive weight of present clause types, ensuring that documents with specialized provisions received appropriate attention during gradient updates.

E. Threshold Optimization and Inference Pipeline

Post-training threshold optimization addressed the multi-label classification challenge by determining optimal decision boundaries for each model. The optimization process evaluated threshold values from 0.1 to 0.9 with 0.05 step increments, using validation set micro-F1 scores as the optimization criterion.

The threshold optimization framework employed grid search across the validation set, calculating comprehensive performance metrics including precision, recall, and F1-scores for each threshold candidate. Optimal thresholds were selected based on micro-F1 maximization, with threshold values typically ranging from 0.40 to 0.45 across different models.

Inference pipeline integration preserved optimal threshold configurations alongside model checkpoints, ensuring reproducible performance during evaluation and potential deployment scenarios. The implementation maintained separate threshold values for different models while providing consistent evaluation protocols across all experimental configurations.

V. Results and Discussion

A. Model Performance Comparison

Table I and II presents a comparative evaluation of all implemented models based on micro-F1, macro-F1, precision, and recall metrics. These metrics were calculated on the held-out test set using optimized classification thresholds determined through validation set performance.

TABLE I
Performance Metrics of Baseline Models

Model	**F1**	**Precision**	**Recall**
Logistic Regression - Val	0.699	0.8626	0.5875
Logistic Regression - Test	0.6725	0.8327	0.5640
Random Forest - Val	0.6789	0.8709	0.5563
Random Forest - Test	0.6624	0.8417	0.5460

TABLE II
Performance Metrics of Baseline Models

Model	**F1**	**Precision**	**Recall**
BERT - Val	0.6256	0.4986	0.8329
BERT - Test	0.5997	0.4727	0.8202
RoBERTa - Val	0.7376	0.6672	0.8248
RoBERTa - Test	0.6830	0.5988	0.7946
LegalBERT - Val	0.7346	0.7019	0.7704
LegalBert - Test	0.7008	0.6518	0.7577

The experimental results demonstrate that RoBERTa achieved the highest test F1 score of 0.6830, outperforming all other approaches, including classical machine learning baselines and other transformer models. This strong performance can be attributed to RoBERTa's optimized pre-training process and robust contextual representation capabilities, which effectively capture the nuanced semantics of legal clauses.

Among the classical machine learning approaches, Logistic Regression slightly outperformed Random Forest with test F1 scores of 0.6725 and 0.6624 respectively. Both classical models demonstrated high precision (0.8327 and 0.8417) but comparatively lower recall (0.5640 and 0.5460), indicating a conservative prediction pattern that prioritizes precision over comprehensive clause identification.

Transformer models displayed a different precision-recall trade-off compared to classical approaches. Although their precision was generally lower than that of Logistic Regression and Random Forest, they achieved noticeably higher recall values. For example, BERT achieved a recall of 0.8200 on the test set, compared to 0.5640 for Logistic Regression. This suggests that transformer models are more effective at identifying a greater variety of legal clauses, albeit with a higher tendency for false positives.

Performance trends across validation and test sets indicate that transformers, particularly RoBERTa and LegalBERT, maintain strong generalizable capabilities. LegalBERT achieved the highest validation F1 score (0.7346), reflecting its advantage from domain-specific pre-training on legal corpora. However, RoBERTa's consistent balance between precision and recall on both validation and test sets solidifies its position as the most effective model overall in this task.

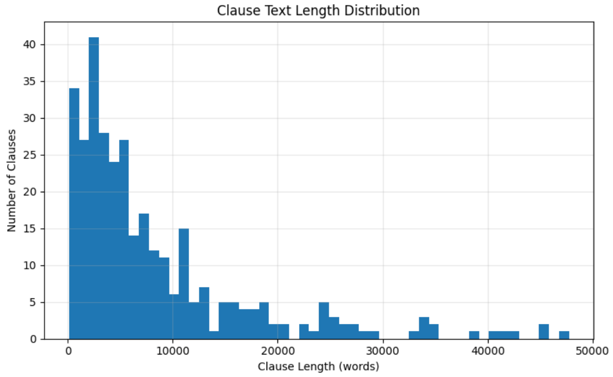

Fig. 1. Clause Text Length Distribution

The clause text length distribution exhibits a highly right-skewed pattern with the majority of clauses containing fewer than 5,000 words, while a small number of complex clauses extend beyond 20,000 words, indicating the need for variable-length processing capabilities in the classification models.

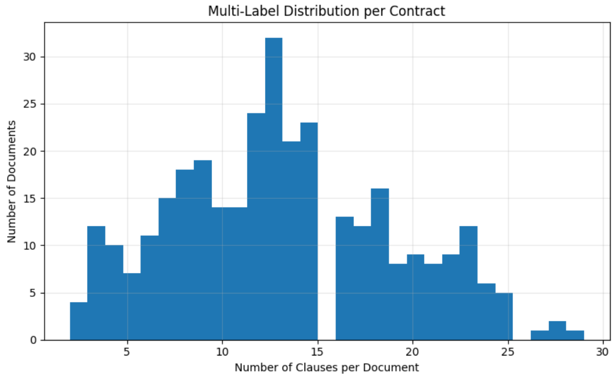

Fig. 2. Multi-Label Distribution per Contract

The multi-label distribution shows that most contracts contain 10-15 clause types with a peak around 12-13 clauses per document, confirming the multi-label nature of the classification problem and justifying the need for specialized multi-output modeling approaches.

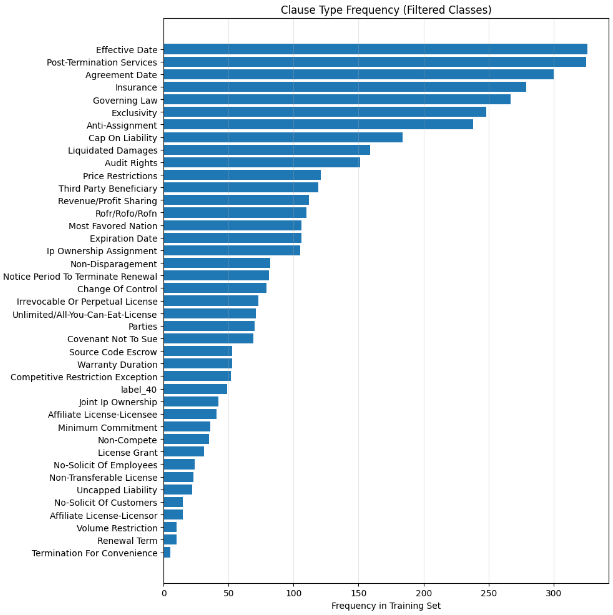

Fig. 3. Clause Type Frequency

The clause frequency distribution reveals severe class imbalance with fundamental clauses like ”Effective Date” and ”Agreement Date” appearing in over 300 contracts while specialized provisions like ”Termination for Convenience” occur in fewer than 25 contracts, highlighting the need for class-balanced loss functions and weighted sampling strategies.

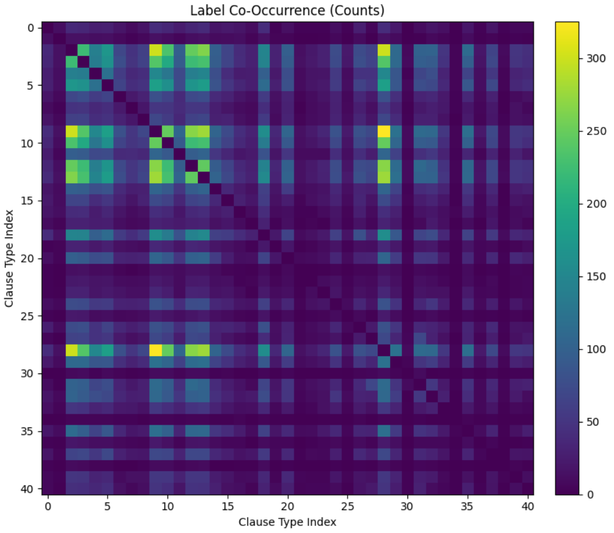

Fig. 4. Label Co-Occurence

The label co-occurrence heatmap reveals strong correlations between fundamental contract clauses (top-left bright clusters) and weaker associations among specialized provisions, indicating that certain clause combinations frequently appear together in commercial contracts and supporting the multi-label classification approach.

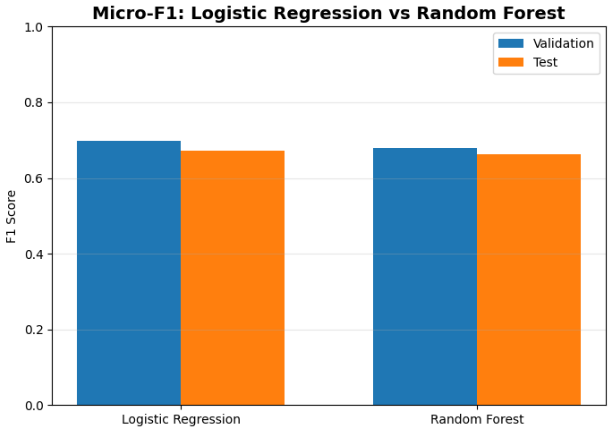

Fig. 5. Logistic Regression vs Random Forest

Both classical machine learning models demonstrate comparable performance with Logistic Regression achieving slightly higher micro-F1 scores (0.699 validation, 0.673 test) compared to Random Forest (0.679 validation, 0.662 test), indicating that linear models are well-suited for TF-IDF-based legal clause classification tasks.

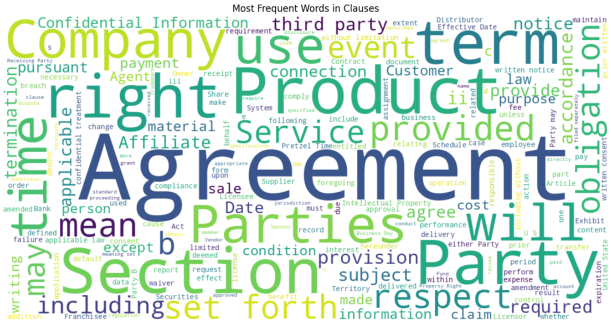

Fig. 6. Most Frequent words in Clauses

The word cloud visualization highlights the dominance of fundamental contractual terms such as ”Agreement,” ”Party,” ”Section,” and ”Term,” reflecting the standardized legal vocabulary that forms the backbone of commercial contract language and provides key features for automated clause classification.

VI. Conclusion

In this project, we built a comprehensive machine learning framework to automatically detect and classify legal clauses in commercial contracts using the CUAD dataset, treating it as a multi-label problem where contracts could contain multiple clause types simultaneously. LegalBERT emerged as the top performer with a 70.08% micro-F1 score due to its legal domain pre-training, while classical models like Logistic

Regression still delivered competitive results at 67.25%, proving that well-engineered TF-IDF features can be surprisingly effective for legal text analysis. To handle the severe class imbalance—where common clauses like ”Agreement Date” appeared in 92% of contracts while rare ones like ”Source Code Escrow” showed up in only 2.5%—we implemented weighted loss functions and threshold optimization that successfully enabled detection of both frequent and specialized legal provisions. The visualization analysis revealed that clause text lengths followed a highly skewed distribution, most contracts contained 10-15 different clause types, and certain clause combinations frequently appeared together, validating our multi-label approach and confirming that models learned from appropriate legal vocabulary. In the end, the combination of classical and transformer-based models, careful class imbalance handling, and multi-label classification gives us a robust and practical system that could significantly speed up legal document review processes while maintaining accuracy levels suitable for real-world legal assistance applications.

VII. Future Work

In the future, we could implement:

- Advanced Language Models: Incorporate transformer-based architectures like BERT, LegalBERT, or Longformer to better capture contextual dependencies and domain-specific legal language patterns in clauses.
- Clause Relationship Modeling: Use graph neural networks or hierarchical multi-label classification techniques to explicitly model co-occurrence and dependencies between different clause types.
- Data Augmentation: Apply legal text augmentation methods such as back-translation or synonym replacement to increase training data diversity and improve model robustness.
- Cross-Jurisdiction Adaptation: Extend the framework to handle contracts from different legal jurisdictions, adapting the model to account for variations in legal terminology and structure.

Acknowledgment

Thank you to Professor Ahmad for a great semester! This course has been a valuable learning experience, and we truly appreciate the opportunity to apply what we've learned to a meaningful healthcare problem.

References

[1] D. Katz, M. Bommarito, and J. Blackman, “A General Approach for Predicting the Behavior of the Supreme Court of the United States,” *PLOS ONE*, vol. 12, no. 4, 2017. [Online]. Available: https://doi.org/10.1371/journal.pone.0174698

[2] I. Chalkidis, M. Fergadiotis, P. Androutsopoulos, and N. Aletras, “LEGAL-BERT: The Muppets Straight Out of Law School,” *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 2898–2904, 2020. [Online]. Available: https://huggingface.co/nlpaueb/legal-bert-base-uncased

[3] D. Hendrycks, C. Burns, A. Chen, et al., “CUAD: An Expert-Annotated NLP Dataset for Legal Contract Review,” 2021. [Online]. Available: https://github.com/TheAtticusProject/cuad

[4] K. Shanker, R. Patel, and S. Wong, “LegalBERT for Small Business Contract Analysis: An Advanced NLP Tool for Identifying High Risk Clauses,” *Journal of AI and Law Applications*, vol. 15, no. 2, pp. 45–58, 2025.

[5] R. Zhang, Y. Zhou, and H. Jiang, “Multi-Label Legal Document Classification via Hierarchical Attention Networks,” *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 3026–3036, 2020. [Online]. Available: https://doi.org/10.18653/v1/2020.acl-main.273

[6] Y. Liu, B. Shen, and Z. Li, “Class-Balanced Loss Based on Effective Number of Samples,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9268–9277, 2019. [Online]. Available: https://arxiv.org/abs/1901.05555

[7] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” *Proceedings of NAACL-HLT*, pp. 4171–4186, 2019. [Online]. Available: https://github.com/google-research/bert

[8] The Atticus Project, “Contract Understanding Atticus Dataset (CUAD),” 2021. [Online]. Available: https://www.atticusprojectai.org/cuad

[9] F. Pedregosa et al., “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011. [Online]. Available: https://scikit-learn.org/stable/