

# ContraLegal-AI: Intelligent Contract Risk Analysis using Multi-Modal Natural Language Processing and Machine Learning

Ayush Kumar Singh, Isha Singh, Priyanka Gnana Karanam  
Null Set

**Abstract**—Manual review of commercial contracts is a time-consuming, expensive, and error-prone process. ContraLegal-AI proposes a hybrid intelligence system designed to automate the extraction, classification, and thematic analysis of legal risk clauses within PDF contracts. By combining advanced Natural Language Processing (NLP) techniques, Term Frequency-Inverse Document Frequency (TF-IDF) vectorization, and a Random Forest classification algorithm, the system classifies clause-level risk with 97% accuracy. Furthermore, we implement a rule-based Hybrid Decision Engine and Unsupervised K-Means clustering to provide robust thematic insights. This paper outlines the quantitative results, empirical evaluation, and system architecture for Milestone 1.

**Index Terms**—Natural Language Processing, Machine Learning, Contract Analysis, Risk Classification, TF-IDF, Random Forest.

## I. INTRODUCTION

In the modern corporate landscape, legal departments are inundated with complex contracts, such as Non-Disclosure Agreements (NDAs), Mergers and Acquisitions (M&A) contracts, and Service Level Agreements (SLAs). Identifying high-risk clauses—such as asymmetrical indemnification, unrestricted liability, or hidden auto-renewal terms—requires meticulous human review.

ContraLegal-AI addresses this bottleneck by providing a highly scalable automated risk dashboard. The core objective of this project is to parse unstructured legal PDF documents and evaluate textual risk through a combination of statistical machine learning and deterministic legal rules.

## II. RELATED WORK

The automation of legal document review has gained significant traction with the advent of Natural Language Processing (NLP). Early systems relied heavily on deterministic rule-based matching [1]. Modern approaches have shifted towards deep learning and transformer-based models like LegalBERT for semantic understanding. However, for baseline commercial viable systems, statistical methods such as TF-IDF coupled with ensemble classifiers remain highly effective and computationally efficient [2]. The availability of expert-annotated legal corpora, such as the Contract Understanding Atticus Dataset (CUAD) [3], has further accelerated the development of supervised machine learning models in the legal tech domain.

## III. METHODOLOGY

The system architecture follows a distinct pipeline format: Data Extraction, Text Normalization, Machine Learning Classification, Hybrid Risk Scoring, and Thematic Grouping.

### A. Dataset Description

The model was trained on a proprietary dataset comprising 21,144 manually annotated legal clauses. The distribution consisted of 12,816 High Risk clauses and 8,328 Low Risk clauses.

### B. Preprocessing

The ingestion layer handles raw, unstructured PDF text. We utilized the PyMuPDF library to geometrically extract textual data. Once extracted, the raw text is subjected to a normalization pipeline utilizing spaCy [4]. Stop words, punctuation, and non-alphanumeric noise are stripped. Sentences are segmented into distinct clauses. Furthermore, a Privacy Masker employs Regular Expressions to dynamically redact Personally Identifiable Information (PII) to ensure data protection compliance.

### C. Feature Engineering

Textual clauses are mathematically transformed into numerical vectors using Term Frequency-Inverse Document Frequency (TF-IDF). The weight  $w_{i,j}$  of term  $i$  in document  $j$  is calculated as:

$$w_{i,j} = \text{TF}_{i,j} \times \log \left( \frac{N}{\text{DF}_i} \right) \quad (1)$$

Where  $\text{TF}_{i,j}$  is the frequency of term  $i$  in clause  $j$ ,  $N$  is the total number of clauses, and  $\text{DF}_i$  is the document frequency of term  $i$ . We configured the vectorizer to capture both unigrams and bigrams, limited to the top 5,000 features.

### D. Models Used

1) *Random Forest Classifier*: We employ a Random Forest Classifier to identify patterns associated with "High Risk" clauses. The final class prediction  $\hat{y}$  is determined via a majority vote among the individual decision trees  $T$ :

$$\hat{y} = \text{mode}\{T_1(x), T_2(x), \dots, T_k(x)\} \quad (2)$$

2) *Hybrid Decision Engine*: To offset the purely statistical nature of the ML classifier, a rule-based inference engine applies deterministic legal constraints. A predefined Legal Keyword Engine scans the clause for high-stakes terminology. A final weighted risk score ( $R_{final}$ ) is geometrically calculated by merging the ML probability ( $P_{ml}$ ) and a Keyword Threat Multiplier ( $M_k$ ):

$$R_{final} = \min(P_{ml} \times M_k, 1.0) \quad (3)$$

3) *Unsupervised Thematic Analysis*: To organize large documents, we implement K-Means clustering. The algorithm partitions  $n$  clauses into  $k$  sets by minimizing the within-cluster sum of squares (WCSS).

#### E. Training and Validation

Due to the implicitly imbalanced nature of legal risk, we utilized balanced class weighting during the training phase. The classification model was evaluated using a 20% hold-out test set ( $N = 4,229$ ) to ensure generalizability to unseen contract clauses.

## IV. RESULTS

### A. Quantitative Results

The Random Forest model achieved exceptional empirical results on the hold-out validation set.

TABLE I  
MODEL PERFORMANCE COMPARISON ON TEST DATA

Risk Class	Acc	Prec	Rec	F1
High Risk	-	0.98	0.97	0.98
Low Risk	-	0.96	0.97	0.97
<b>Macro Avg</b>	<b>0.97</b>	<b>0.97</b>	<b>0.97</b>	<b>0.97</b>

### B. Figures

The confusion matrix demonstrates the model's proficiency in accurately identifying both the High Risk and Low Risk classes with minimal false positives or false negatives.

## V. DISCUSSION

The hybrid approach combining TF-IDF Random Forest with rule-based heuristics proved highly effective for achieving accurate, interpretable risk classification. However, the system is fundamentally constrained by its lack of deep semantic understanding. As a statistical bag-of-words model, it may struggle with highly complex, nuanced legal phrasing where word order dictates the actual obligation.

For future implementations, we plan to migrate from statistical TF-IDF word representations to deep-learning neural word embeddings such as LegalBERT. Furthermore, expanding the training corpus to encompass more comprehensive datasets like the Contract Understanding Atticus Dataset (CUAD) will significantly bolster real-world commercial robustness.

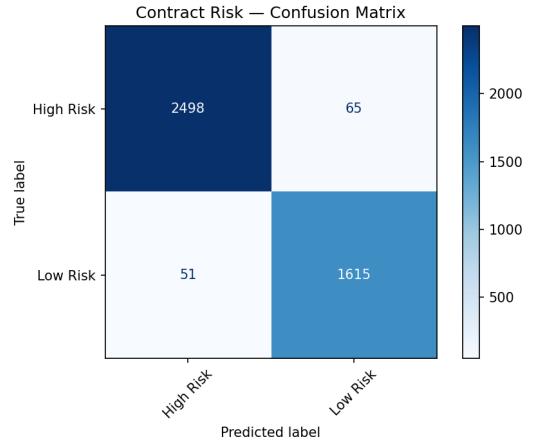


Fig. 1. Confusion Matrix representing True Positives and False Positives across the test subset of legal clauses.

## VI. CONCLUSION

The ContraLegal-AI system successfully fulfills its Milestone 1 objectives by providing an end-to-end, modular application that accurately segments, classifies, and clusters legal text. The implementation demonstrates how the fusion of modern UI dashboards, statistical machine learning, and deterministic legal rules can effectively democratize and accelerate the contract review process.

## REFERENCES

- [1] I. Chalkidis and I. Androutsopoulos, "A deep learning approach to contract element extraction," *JURIX*, 2017.
- [2] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [3] The Atticus Project, "Contract understanding atticus dataset (cuad)," <https://www.atticusprojectai.org/cuad>, 2021.
- [4] M. Honnibal and I. Montani, "spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing," 2017.

## APPENDIX

This appendix provides details regarding the project's GitHub repository and associated file structure for reproducibility and open-source collaboration.

### A. Repository Link

The complete source code, datasets, trained models, and documentation are available at:

<https://github.com/AyushCoder9/ContraLegal-AI>

## B. Repository Structure

The project repository follows the modular structure shown below:

```
ContraLegal-AI/
|- app.py
|- src/
|   |- model_trainer.py
|   |- ui/
|   |- data_pipeline/
|   |- inference/
|   |- model/
|- data/
|   |- raw/
|- models/
|- report/
|- requirements.txt
```

## C. Description of Key Components

- **data/**: Contains the raw legal clauses dataset (`legal_docs_modified.csv`).
- **src/model/**: Contains scripts for data loading, Random Forest model training, and evaluation metric generation.
- **src/inference/**: Contains the hybrid prediction scripts and legal concept rules for real-time analysis.
- **src/ui/**: Contains the Streamlit components for visualizing the risk metrics and plotting interactive DataFrames.
- **models/**: Stores the serialized `.pkl` files for the vectorizer and trained model.
- **report/**: Contains the LaTeX project report and references.
- **app.py**: The primary application entry point for the Streamlit dashboard.