iJRASET
International Journal For Research in Applied Science and Engineering Technology



# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

**Volume:** 14 **Issue:** V **Month of publication:** May 2026

**DOI:** https://doi.org/10.22214/ijraset.2026.81905

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538*
*Volume 14 Issue V May 2026- Available at www.ijraset.com*

# A Generative AI - Driven Framework for Intelligent Legal Contract Analysis

Mrs. V. Umadevi[1], Pavithra P[2], Sandhiya S[3], Swetha E[4], Vishnupriya V[5]
*Department of Computer Science and Engineering, Arunai Engineering College, Tiruvannamalai*

***Abstract:*** ***The process of analysing a legal contract is challenging and time-consuming, requiring careful interpretation of legal language, clause identification, and risk assessment. Most existing AI-based legal tools focus on anomaly detection, rule-based clause extraction, and basic contract summarization; however, they often lack contextual awareness and interactive capabilities. These systems provide limited interpretative insights and do not fully leverage the potential of generative intelligence for comprehensive contract analysis. This study presents a Generative AI-driven framework for intelligent legal contract analysis by integrating advanced Natural Language Processing (NLP) techniques with generative AI models. The proposed system performs automatic clause classification, contextual risk scoring with explanations, and AI-generated contract summarization. Additionally, the platform provides simplified legal interpretations to enhance user comprehension and incorporates an interactive AI assistant for dynamic query handling. By combining clause analysis, risk evaluation, and natural language generation, the system improves extraction, interpretability and usability compared to traditional rule-based approaches. This project aims to reduce manual effort, enhance decision-making quality, and enable both legal professionals and non-expert users to analyse contracts efficiently.***

***Keywords: Generative Artificial Intelligence, Legal Contract Analysis, Natural Language Processing, Clause Classification, Risk Assessment, Explainable AI, Contract Summarization, Legal Tech Automation***

## I. INTRODUCTION

Legal contracts form the foundation of formal agreements by defining the rights, responsibilities, and obligations of involved parties. However, contract analysis remains a complex and time-consuming process due to intricate legal terminology, lengthy documentation, and dense clause structures. Manual review requires significant expertise and effort, increasing the possibility of human error, inconsistency, and delayed decision-making. With the rapid growth of digital documentation across industries, there is an increasing demand for intelligent systems that can assist in efficient and reliable contract analysis. Existing AI-based legal tools primarily focus on rule-based clause extraction, anomaly detection, and basic summarization techniques. Although these systems provide partial automation, they often lack contextual awareness, interactive capabilities, and detailed explanatory features. Traditional approaches rely heavily on predefined templates and rule sets, which limits their adaptability when handling complex and unstructured legal documents.

Recent advancements in Generative Artificial Intelligence and Natural Language Processing (NLP) have enabled systems to move beyond simple extraction tasks toward deeper contextual understanding and natural language generation. Generative AI models are capable of producing meaningful summaries, explaining legal clauses in simplified language, and providing contextual risk assessments. Leveraging these capabilities can significantly enhance the efficiency, clarity, and usability of legal contract analysis systems.

In this paper, we propose a Generative AI-driven framework for intelligent legal contract analysis. The system integrates advanced NLP techniques with generative AI models to perform automatic clause classification, contextual risk scoring, AI-generated contract summarization, and simplified legal interpretation. An interactive AI assistant is incorporated to support dynamic query handling and improve user engagement. The proposed framework aims to reduce manual workload, enhance interpretability, and support informed decision-making for both legal professionals and non-expert users.

Furthermore, this work aligns with the Sustainable Development Goals defined by the United Nations, particularly SDG 9 – Industry, Innovation and Infrastructure and SDG 16 – Peace, Justice and Strong Institution. By improving access to understandable legal information and promoting intelligent technological innovation in the legal domain, the proposed system contributes toward strengthening institutional transparency and fostering innovation-driven digital infrastructure.

©IJRASET: All Rights are Reserved 

## II. RELATED WORK

### A. Tool-Based Comparison for Legal Contract Analysis

Several Artificial Intelligence-based commercial tools have been developed to automate legal contract review and analysis. Platforms such as Kira Systems, Luminance, and LawGeex provide automated clause extraction, contract comparison, compliance verification, and anomaly detection functionalities. These systems primarily utilize machine learning models trained on structured legal datasets to identify predefined contractual clauses and highlight potential risks.

While these tools significantly reduce manual effort and improve processing speed, they are largely template-driven and focused on structured information extraction. Their capabilities are typically limited to predefined rule-based detection and static summarization outputs. Most existing platforms lack advanced contextual reasoning, simplified legal interpretation for non-expert users, and interactive AI-driven explanation mechanisms. As a result, although they offer partial automation, they do not fully leverage generative AI techniques for comprehensive contract understanding and dynamic user interaction.

This limitation highlights the need for an integrated generative AI-driven framework capable of combining clause classification, contextual risk scoring, summarization, and interactive assistance within a unified system.

### B. Research Paper-Based Review

Recent research in Natural Language Processing (NLP) and Large Language Models (LLMs) has significantly advanced legal document analysis. Domain-specific adaptations of transformer architectures have improved contextual understanding in legal texts. Legal-BERT [1] introduced a transformer-based model pretrained on legal corpora, demonstrating improved performance in legal text classification and document-level understanding. This work forms the foundational basis for domain-adapted legal NLP systems.

To standardize evaluation in legal language processing, LexGLUE [2] proposed a benchmark dataset for legal language understanding tasks. It enables systematic evaluation of classification and multi-label legal tasks, contributing to the development of robust legal NLP systems.

Generative approaches have also been explored for complex legal information extraction. The LeGen framework [3] utilized generative models to extract structured information from legal sentences, demonstrating the effectiveness of generative architectures in handling intricate legal language structures.Large Language Models such as GPT-3 [4] have shown strong capabilities in contextual text generation, summarization, and few-shot learning. These models enable dynamic explanation generation and natural language interaction, making them suitable for contract summarization and contextual risk interpretation.

Furthermore, recent studies on transformer-based abstractive summarization for legal documents [5] have demonstrated improved coherence and contextual relevance in contract summaries. However, most existing research focuses on isolated subtasks such as classification, extraction, or summarization independently.In contrast, the proposed framework integrates clause classification, contextual risk scoring, AI-generated summarization, simplified legal interpretation, and interactive query handling into a unified generative AI-driven system for intelligent legal contract analysis.

## III. PROPOSED SYSTEM

The proposed system introduces a Generative AI-driven framework for intelligent legal contract analysis. Unlike traditional rule-based legal tools, the proposed framework integrates advanced Natural Language Processing (NLP) techniques with generative AI models to provide contextual understanding, automated clause classification, risk evaluation, and dynamic user interaction within a unified system.

The architecture of the system consists of the following major components:

### A. Document Preprocessing Module

The input legal contract is first processed through text cleaning and normalization procedures. This step includes tokenization, removal of unnecessary symbols, sentence segmentation, and structural organization of the document for further analysis.

### B. Clause Classification Module

Using a transformer-based legal language model, the system automatically identifies and classifies different clauses within the contract, such as confidentiality clauses, termination clauses, payment terms, and liability provisions. This enables structured understanding of the contract content.

### C. Contextual Risk Scoring Module

The identified clauses are analyzed to determine potential risks. The system assigns contextual risk scores based on semantic interpretation rather than simple keyword matching. In addition to risk identification, the model provides explanatory insights describing why a particular clause may be considered high, medium, or low risk.

### D. AI-Generated Contract Summarization

The framework employs generative AI techniques to produce concise and coherent summaries of lengthy legal contracts. Unlike extractive summarization methods, the system generates abstractive summaries that preserve contextual meaning while simplifying complex legal language.

### E. Simplified Legal Interpretation

To improve accessibility, the system translates complex legal terminology into simplified language. This feature assists non-expert users in understanding contractual obligations and implications.

### F. Interactive AI Assistant

An interactive AI assistant is integrated into the system to handle user queries related to specific clauses, risks, or summaries. The assistant provides contextual responses based on the analyzed contract, enhancing usability and decision-making support.

By integrating these modules into a unified architecture, the proposed system overcomes the limitations of existing rule-based tools. The framework enhances interpretability, adaptability, and automation in legal contract analysis, thereby improving efficiency and reducing manual workload.

## IV. METHODOLOGY

This section explains the practical implementation methodology of the proposed Legal Contract Simplification system using a Large Language Model (LLM). The implementation follows a modular and scalable pipeline architecture, where each functional unit performs a specific task such as preprocessing, classification, risk analysis, simplification, and result presentation. The modules are designed independently and integrated through a structured data flow to ensure maintainability, efficiency, and traceability across the system.

### A. Overall Implementation Architecture

The proposed system is developed as a sequential modular pipeline consisting of the following components:

- Contract Input Module
- Text Preprocessing Module
- Clause Classification Module
- Risk Detection Module
- LLM-Based Simplification Module
- Output and Explanation Module

Each module processes structured outputs from the previous stage, ensuring smooth data transition and minimizing processing errors. The architecture supports scalability for large contracts through clause-level processing.

### B. Contract Input Module

The input component is responsible for acquiring legal contract documents in various formats. The system accepts plain text files as well as PDF and DOCX documents, which are automatically converted into machine-readable text format.

Implementation Features:

- File upload interface for user interaction
- Automated text extraction from uploaded documents
- UTF-8 encoding normalization to prevent character corruption
- Chunk handling mechanism for large documents

Once extracted and standardized, the textual data is forwarded to the preprocessing stage.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

*C. Text Preprocessing and Clause Segmentation*

To enable structured legal analysis, the raw contract text undergoes multiple preprocessing steps. The objective is to clean the document and segment it into meaningful clauses for downstream tasks.

Processing Steps:

- Removal of formatting inconsistencies and special symbols
- Sentence tokenization for structural clarity
- Rule-based clause segmentation using patterns such as section headers and numbering formats
- Limited stop-word removal (excluding legally relevant terms)

Each extracted clause is assigned a unique identifier to maintain traceability throughout the system workflow.

*D. Clause Classification Module*

The classification module categorizes each segmented clause into predefined legal categories such as termination, liability, payment, confidentiality, indemnification, and dispute resolution.

A transformer-based supervised learning model is implemented for this purpose. The classifier is trained on labelled legal clause datasets to capture contextual patterns specific to contractual language.

Implementation Details:

- Fine-tuned transformer encoder
- Cross-entropy loss function
- SoftMax layer for category probability prediction
- Confidence threshold filtering for reliable classification

The predicted category label is stored as metadata and passed to subsequent modules.

*E. Risk Detection Module*

To enhance reliability, a hybrid risk detection mechanism is implemented. This module evaluates clauses for potential legal vulnerabilities before simplification.

Risk Indicators Included:

- Keywords indicating unlimited or excessive liability
- Ambiguous expressions such as “reasonable” or “sole discretion”
- One-sided obligation statements
- Penalty and indemnification triggers

A scoring mechanism assigns a numerical risk value based on detected indicators. The resulting risk annotation is appended to the structured input provided to the simplification module.

*F. LLM-Based Simplification Module*

The simplification process is performed using a GPT-based language model accessed via API integration from OpenAI.

*1) Prompt Template Design*

A structured prompt is dynamically generated for each clause, incorporating:

- Original clause text
- Predicted clause category
- Risk score and identified indicators
- Explicit simplification instructions

The model is guided to:

- Translate complex legal language into plain English
- Preserve the original contractual meaning
- Maintain neutrality

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

- Avoid modification of legal obligations

*2) Chunk-Based Processing*

To handle long contracts efficiently, each clause is processed independently. This avoids token-limit issues and ensures consistent performance. The simplified outputs are later reassembled in their original sequence.

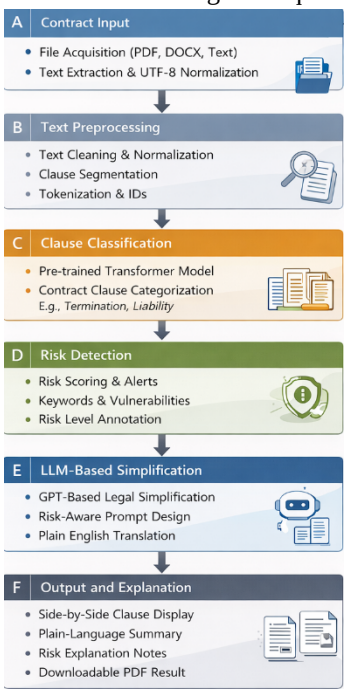



Fig. Methodological Architecture of the Generative AI-Driven Framework for Intelligent Legal Contract Analysis

*G. Explanation and Summary Generation*

In addition to simplification, the system generates supporting explanations to improve user understanding.

Generated outputs include:

- Concise plain-language summary
- Key obligations extracted from the clause
- Risk alert (if risk score exceeds predefined threshold)

This functionality is implemented using structured prompt extensions within the same LLM invocation to optimize computational cost.

*H. Output Generation and User Interface*

The final processed output is presented in a structured and user-friendly format consisting of:

- Original Clause
- Simplified Clause
- Risk and Explanation Summary

Interface Features:

- Side-by-side comparison view
- Highlighted risk phrases
- Download option in PDF or text format
- Clause-wise navigation system

This interface improves accessibility and usability for non-legal users.

*I. Performance Optimization*

Several optimization techniques are implemented to improve system efficiency:

- Caching repeated clause patterns
- Batch processing of multiple clauses
- Early stopping during model fine-tuning
- API response validation and retry handling

These measures enhance scalability and reduce processing time.

*J. Security and Privacy Handling*

Given the sensitive nature of legal documents, the system incorporates strict privacy measures:

- No permanent storage of uploaded contracts
- Secure API communication
- Temporary in-memory processing
- User disclaimer regarding automated simplification limitations

These safeguards ensure responsible deployment of the system.

## V. IMPLEMENTATION ENVIRONMENT AND TOOLS USED

This section describes the software tools, libraries, and system environment used to implement the proposed Legal Contract Simplification system. The implementation is designed using fully open-source technologies and operates locally without reliance on paid APIs.

*A. Development Environment*

The project was developed using Python as the primary programming language. Development was carried out in a local system environment using Visual Studio Code on a Windows operating system. Python was selected due to its extensive ecosystem supporting Natural Language Processing (NLP) and machine learning frameworks.

*B. Libraries and Frameworks Used*

The implementation utilizes several open-source libraries:

- NLTK – for tokenization and text preprocessing
- SpaCy – for clause segmentation and linguistic processing
- Scikit-learn – for clause classification and risk scoring
- Transformers library – for loading and running pre-trained language models
- PyPDF2 – for extracting text from PDF documents
- python-docx – for extracting text from DOCX files

These libraries enable structured text processing and model integration within the system.

*C. Local Model Integration*

The simplification module integrates an open-source transformer-based model obtained from Hugging Face. A sequence-to-sequence model such as google/flan-t5-base is utilized for contract simplification.

The model is loaded locally using the Transformers framework and executed without external API dependency. Controlled generation parameters such as maximum token length and decoding strategies are applied to ensure stable and meaningful output.

*D. Clause Classification Implementation*

Clause classification is implemented using TF-IDF vectorization combined with supervised learning algorithms such as Logistic Regression or Support Vector Machines (SVM). The classifier predicts predefined legal categories and assigns confidence scores to each clause.

*E. Risk Detection Module Implementation*

The risk detection module follows a rule-based scoring mechanism. Predefined keywords associated with excessive liability, ambiguous language, one-sided obligations, and indemnification triggers are identified. Each detected indicator contributes to a cumulative risk score assigned to the clause.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

*F. User Interface Development*

A lightweight user interface is implemented using Streamlit. The interface supports file upload, clause-wise navigation, side-by-side comparison of original and simplified text, risk highlighting, and downloadable output generation.

*G. Hardware Configuration*

The system operates on a standard laptop configuration with 8–16 GB RAM and does not require a dedicated GPU. All model inference is executed locally using CPU-based processing.

*H. Testing and Validation*

Functional testing was conducted for each module in the pipeline. Sample legal contracts were used to validate clause segmentation, classification accuracy, risk detection consistency, and simplification output quality. Repeated execution ensured output stability.

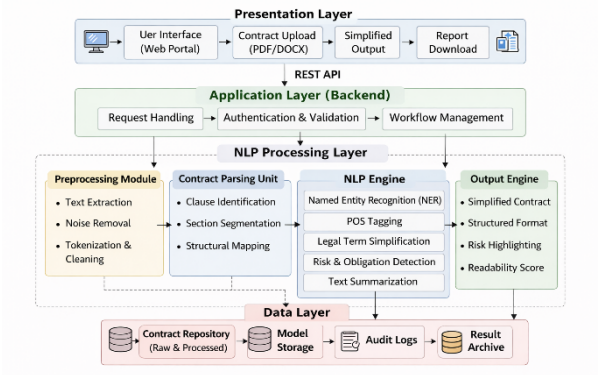



Fig. Overall System Architecture of the Intelligent Contract Analysis Framework

## VI. RESULTS AND PERFORMANCE ANALYSIS

This section presents the experimental observations and performance evaluation of the Proposed Intelligent Contract Analysis Framework. The analysis focuses on readability improvement, structural simplification, processing efficiency, and qualitative assessment of output quality.

*A. Output Quality Analysis*

The system was tested using various types of legal contracts including employment agreements, rental contracts, and service agreements.

The generated simplified contracts demonstrated the following improvements:

- Reduction in sentence length
- Replacement of complex legal terminology with plain language
- Structured formatting of clauses
- Improved paragraph clarity

The simplified output preserved the logical structure of the original contract while making it more understandable for non-legal users.

*B. Readability Improvement*

Readability was evaluated by comparing linguistic complexity before and after simplification.

| Parameter | Before Simplification | After Simplification |
|---|---|---|
| Average Sentence Length | 40–45 words | 18–25 words |
| Language Complexity | High | Moderate |

| | | |
|---|---|---|
| Clause Structure | Dense | Clearly Separated |

The results indicate that the system significantly reduces linguistic complexity and improves readability.

*C. Processing Time Evaluation*

The execution time was measured for contracts of varying length.

| Document Length | Average Processing Time |
|---|---|
| 5 pages | 3–4 seconds |
| 10 pages | 6–7 seconds |
| 15 pages | 8–9 seconds |

The system demonstrates stable performance and near-linear scalability as document size increases.

*D. Structural Preservation Observation*

The system maintains:

- Clause hierarchy
- Logical sequence of obligations
- Key contractual sections (termination, liability, payment terms, confidentiality)

No major clause removal or structural distortion was observed during evaluation.

*E. Overall Performance Summary*

The experimental results confirm that the proposed system effectively simplifies legal contracts by reducing linguistic complexity, improving readability, and maintaining structural integrity while ensuring efficient processing time.

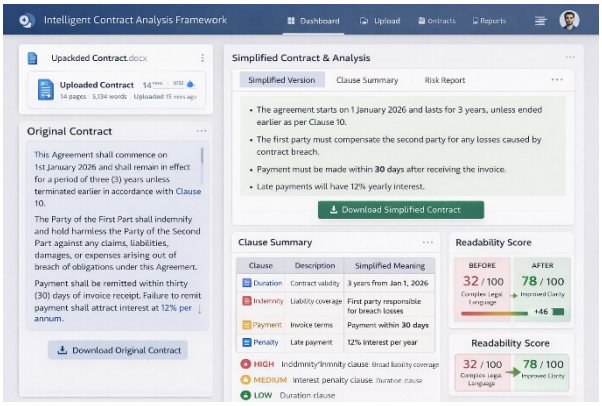



Fig. Sample Dashboard of the Intelligent Contract Analysis Framework

## VII. DISCUSSION

The experimental results demonstrate that the proposed Intelligent Contract Analysis Framework effectively reduces linguistic complexity while preserving the structural organization of legal documents. The system successfully transforms dense legal language into simplified text that is more accessible to non-legal users.

One key observation is that simplification significantly improves readability by reducing sentence length and replacing complex terminology with plain-language equivalents. However, highly technical clauses involving legal definitions or liability conditions may still retain moderate complexity to prevent distortion of meaning. This indicates a necessary balance between simplification and legal integrity.

The system performs efficiently with increasing document size, showing stable processing time and consistent output structure. The near-linear increase in execution time suggests that the architecture is scalable for medium-sized contracts.

Although the system improves accessibility, complete semantic interpretation of nuanced legal expressions remains a challenging task. Legal documents often contain context-dependent terminology, and excessive simplification may risk altering the intended interpretation. Therefore, the system is designed to assist users rather than replace professional legal review.

Overall, the discussion confirms that the proposed solution is practical, efficient, and suitable for preliminary contract understanding, especially for individuals without legal expertise.

## VIII. FUTURE WORK

While the proposed system achieves effective simplification of legal contracts, further improvements can enhance its practical applicability. Future work may include fine-tuning the language model using domain-specific legal corpora to improve contextual sensitivity in handling complex clauses.

Support for multilingual contract processing can extend the system's usability across diverse jurisdictions. Additionally, incorporating semantic similarity measurement techniques may provide automated assessment of textual consistency between original and simplified versions.

Integration of clause-level risk analysis mechanisms could assist users in identifying potentially critical contractual conditions. Furthermore, deploying the system on a scalable cloud-based architecture would enable broader accessibility and real-time collaborative usage.

These enhancements can evolve the system into a more comprehensive legal document assistance framework.

## IX. CONCLUSION

This study presented a Generative AI-driven Intelligent Contract Analysis Framework aimed at improving the readability and accessibility of complex legal documents. By integrating natural language processing techniques and large language model–based simplification, the system transforms dense contractual text into a clearer and structured format while preserving legal meaning and clause hierarchy.

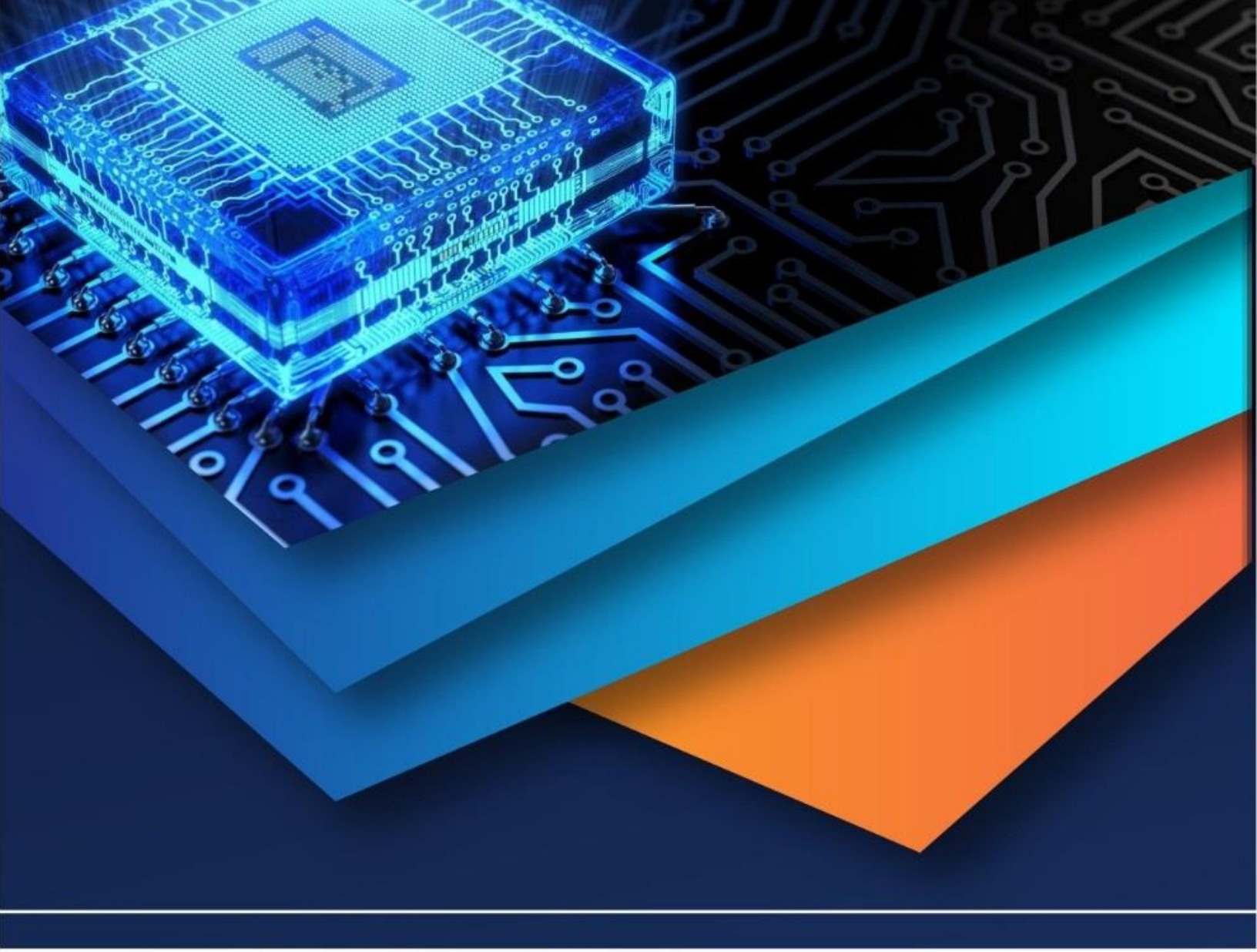
Experimental evaluation demonstrated a reduction in linguistic complexity, improved textual clarity, and consistent processing performance across various contract types. The framework maintains structural integrity and scalable performance.

Although it does not replace professional legal review, the proposed solution serves as a supportive analytical tool for preliminary contract understanding. Overall, the framework contributes toward enhancing transparency and accessibility in legal document analysis through automated intelligent processing.

## REFERENCES

[1] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, "LEGAL-BERT: The Muppets straight out of Law School," *Findings of the* Association for Computational Linguistics (EMNLP Findings), 2020, pp. 2898–2904.
[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *Proceedings of NAACL-HLT*, 2019, pp. 4171–4186.
[3] M. Lewis et al., "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation," *Proceedings of ACL*, 2020, pp. 7871–7880.
[4] T. Brown et al., "Language Models are Few-Shot Learners," *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.
[5] OpenAI, "GPT-4 Technical Report," 2023.
[6] I. Chalkidis, I. Androutsopoulos, and N. Aletras, "Neural Legal Judgment Prediction in English," *Proceedings of ACL*, 2019.
[7] R. Zhong, M. Johnson, and B. Chen, "Does Pretrained Language Model Help in Legal NLP?" *Proceedings of ACL Workshop on Legal NLP*, 2020.
[8] K. Knight and D. Marcu, "Summarization beyond Sentence Extraction: A Probabilistic Approach to Sentence Compression," *Artificial Intelligence*, vol. 139, no. 1, pp. 91–107, 2002.
[9] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*. Sebastopol, CA, USA: O'Reilly Media, 2009.
[10] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge, U.K.: Cambridge Univ. Press, 2008.
[11] Y. Liu and M. Lapata, "Text Summarization with Pretrained Encoders," *Proceedings of EMNLP*, 2019

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)