# Analyzing Legal Texts Using Legal Longformer for Document Summarization

Abhishek Ram[1], Atharv Limje[2], Aman Sahani[3], Dr. Sarita Patil[4]
[1,2,3,4]*B. Tech Computer Engineering, G. H. Raisoni College of Engineering and Management Pune, India*

***Abstract*—Reviewing lengthy and complex legal documents is a predominantly manual, time-consuming process that is highly susceptible to human error. To address this bottleneck, this paper presents an automated, end-to-end Natural Language Processing (NLP) pipeline designed to streamline the extraction, semantic search, and summarization of legal contracts. The proposed architecture ingests digital PDFs using PyMuPDF and processes the unstructured text through a sequence of domain-specific language models. Specifically, Legal-ALBERT, evaluated against the Contract Understanding Atticus Dataset (CUAD), is utilized to accurately classify and extract critical legal clauses. Concurrently, Legal-Longformer generates dense document embeddings to enable context-aware semantic search, bypassing traditional keyword limitations. To synthesize the retrieved information, Long-T5 performs abstractive summarization, distilling multi- page documents into concise overviews. Furthermore, a translation layer is integrated to support multilingual accessibility. Evaluated using F1, ROUGE, and BLEU metrics, the system demonstrates high accuracy and consistency, effectively reducing document review time from hours to minutes while successfully handling documents up to 100 pages in length.**

***Index Terms*—LegalTech, Natural Language Processing, Abstractive Summarization, Semantic Search, Legal-ALBERT.**

## I. INTRODUCTION

The legal domain is inherently reliant on extensive, com- plex documentation. Contracts, court judgments, and statutory agreements frequently span tens to hundreds of pages, saturated with dense terminology and domain-specific jargon. Traditionally, reviewing and comprehending these documents has been a manual, labor-intensive process. This manual approach is not only time-consuming and financially burdensome but also highly susceptible to human error, particularly when identifying critical clauses or assessing legal obligations.

While the rapid advancement of Artificial Intelligence (AI) and Natural Language Processing (NLP) has introduced automated summarization tools, standard Large Language Models (LLMs) face significant limitations in the legal sector. General- purpose models frequently struggle with restricted context windows (token limits) when processing lengthy documents, leading to truncated analyses. More critically, they are prone to "hallucinations" generating factually incorrect text which is an unacceptable risk in legal applications where a single altered word can change the legal intent. Furthermore, in linguistically diverse landscapes such as India, there is a profound disconnect between basic English literacy and true legal comprehension. Ordinary citizens, government agencies, and business owners often find themselves alienated by the language barrier of legal English, even if they can read the words.

To bridge this gap, this paper proposes an end-to-end Legal Document Analyzer designed to automate the extraction, search, and summarization of complex legal texts. Accessible via a user-friendly web interface built on the Flask framework, the system leverages a pipeline of domain-specific, state-of- the-art NLP models. By utilizing models explicitly trained on legal data and integrating advanced translation layers, the proposed system transforms opaque legal jargon into simple, sensible, and multilingual summaries.

The primary contributions of this paper are as follows:

- Domain-Specific Clause Extraction: Implementation of Legal-ALBERT, fine-tuned on the Contract Understand- ing Atticus Dataset (CUAD), to accurately identify and classify critical contract elements such as deadlines, lia- bilities, and termination clauses.

- Context-Aware Semantic Search: Utilization of Legal- Longformer to generate dense document embeddings, enabling users to search for legal concepts based on underlying meaning rather than exact keyword matches.
- Grounded Abstractive Summarization: Application of Long-T5 to condense multi-page documents into concise summaries, utilizing the previously extracted clauses to anchor the summary and mitigate the risk of model hallucinations.
- Multilingual Accessibility: Integration of a translation layer powered by Gemini, allowing users to read and understand complex legal summaries in their preferred regional languages, specifically targeting the needs of the Indian demographic.

## II. RELATED WORK

The application of Natural Language Processing (NLP) in the legal domain has rapidly evolved, transitioning from general-purpose models to domain-specific architectures. However, deploying these models for diverse, real-world applications particularly within the Indian legal landscape presents ongoing challenges in scalability, multilingual support, and domain specificity.

### A. Legal Summarization and General Transformers

Recent studies have explored transformer models for legal document summarization. Research published in IJCRT (2024)
[1] evaluated models such as T5, BART, and PEGASUS on Indian case laws, concluding that general-purpose models severely underperform without domain-specific fine-tuning. Similarly, Santosh et al. (2024) [2] developed LexSumm and LexT5 for legal summarization tasks. While effective, their work is primarily restricted to English texts, largely bypassing the critical need for multilingual capabilities in linguistically diverse regions.

### B. Multilingual and Large-Scale Legal Corpora

To address language barriers, Niklaus et al. (2024) [3] introduced MultiLegalPile, pretraining models like RoBERTa and Longformer on a massive multilingual dataset. Furthermore, Niklaus et al. (2024) [4] developed BRIEFME, a multilingual and multi-task benchmark for legal NLP. While these works advance multilingual processing, they primarily focus on establishing benchmarks rather than creating deployable applications. Additionally, they incur high computational costs and lack fine-tuning specifically tailored to the nuances of the Indian judiciary.

### C. Indian Legal Context and Domain-Specific Datasets

Recent efforts have targeted the Indian legal system directly to improve regional applicability. Deshmukh et al. (2025)
[5] introduced the IndianBailJudgments-1200 dataset, fine- tuning Legal BERT and Gemini models specifically for bail order analysis. Similarly, Joshi et al. (2024) [6] created the Indian Bail Judgments-IL-TUR multi-task evaluation bench- mark encompassing Hindi, English, and regional languages. Despite these advancements, the scope of these solutions remains narrow often restricted strictly to bail judgments and they require further real-world testing and deployment frameworks to be genuinely accessible to end-users.

### D. Frameworks and Deployment Challenges

A comprehensive survey by Ariai et al. (2024) [7] high- lighted existing gaps in legal NLP, emphasizing the limitations of current models like Legal BERT and ALBERT when processing lengthy documents, though the study did not propose a specific architectural solution. Addressing system integration, Lipianina-Honcharenko et al. (2024) [8] proposed a cyclical AI framework combining transformer models for classification and compliance. However, this approach relies heavily on high-end infrastructure for real-time usage, making it inaccessible for lightweight deployment.

### E. Identified Literature Gaps

The existing literature highlights a definitive gap: there is a lack of end-to-end, lightweight systems that combine long-document semantic search, abstractive summarization, and multilingual translation into a deployable, user-friendly application. The proposed system addresses these limitations by integrating Legal-Longformer, Legal-ALBERT, and Long- T5 into a low-overhead web framework, specifically designed to democratize legal comprehension within the broader Indian legal context.

## III. METHODOLOGY

The proposed Legal Document Analyzer employs a sophisticated, multi-stage Natural Language Processing pipeline to automate the ingestion, comprehension, summarization, and translation of complex legal texts. Designed specifically to circumvent the token limitations and hallucination risks inherent in standard Large Language Models, the architecture utilizes a modular, Retrieval-Augmented Generation (RAG) approach. The end-to-end workflow is orchestrated through a lightweight web backend, ensuring high accessibility. The detailed operational mechanics of each module are described below. Figure 1 illustrates the complete processing pipeline.

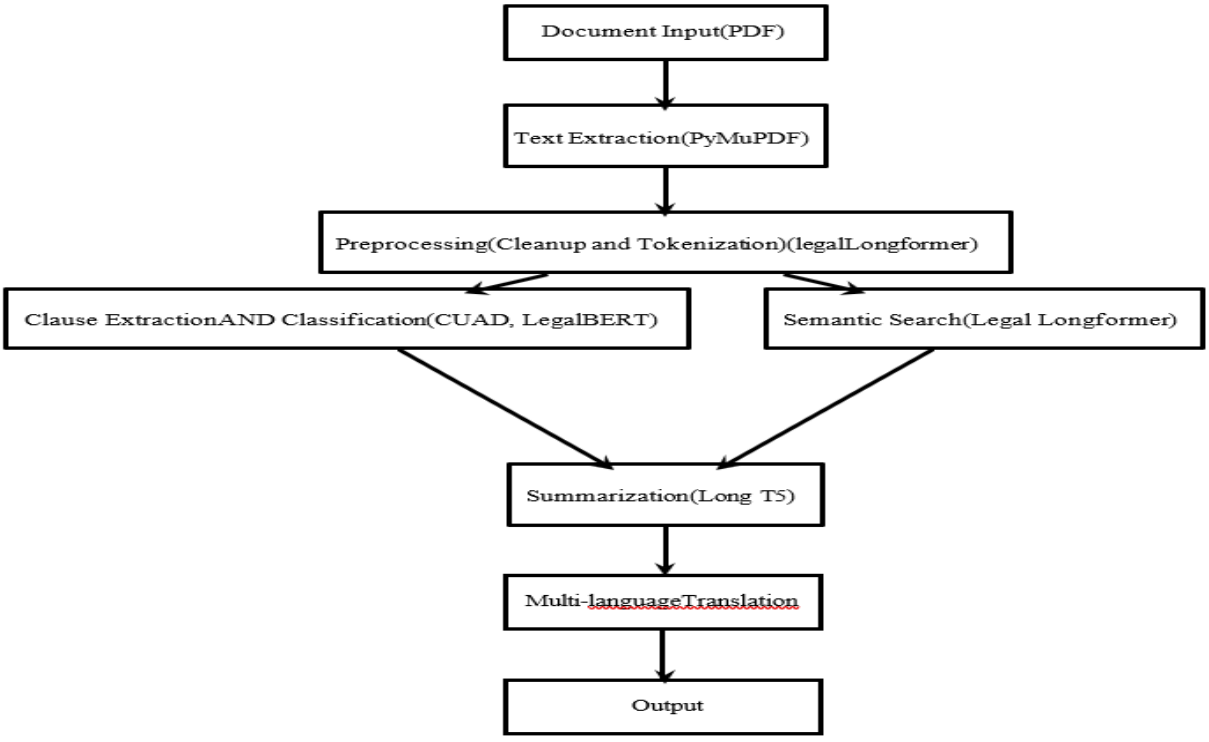


Fig. 1. System Architecture and Processing Pipeline

### A. Document Ingestion and Raw Text Extraction

The analytical pipeline commences with the ingestion of unstructured legal documents, typically spanning 10 to 100 pages, uploaded via the web interface in PDF format. Text extraction is executed using the PyMuPDF library. Unlike standard Optical Character Recognition (OCR) tools that only interpret visual layouts, PyMuPDF directly accesses the document's internal data structures. This allows for rapid, high-fidelity extraction of raw string data while intrinsically preserving paragraph boundaries, structural hierarchies, and reading orders. Preserving this structural integrity is a critical prerequisite for the subsequent clause segmentation phase, as legal definitions often span multiple interconnected para-graphs.

### B. Data Preprocessing and Tokenization

Raw extracted text inherently contains noise that degrades model performance. The preprocessing module applies a series of regular expression (Regex) filters and string normalization techniques to sanitize the data. This involves stripping arti- fact metadata, such as page numbers, repeating headers, and footers, as well as normalizing inconsistent whitespace and encoding anomalies.

Following sanitization, the text undergoes structural tokenization. Rather than utilizing generic tokenizers, the system integrates the tokenizer native to the Legal-Longformer architecture. This ensures that the text is chunked into logical sentences and overarching legal clauses without fracturing critical contextual dependencies. By aligning the tokenization strategy

directly with the downstream models, the system prevents the truncation of legal definitions that might otherwise span across arbitrary token boundaries.

C. Clause Extraction and Multi-Class Categorization

Once structured, the textual chunks are routed to the Legal- ALBERT (A Lite BERT) module for clause extraction and classification. Legal-ALBERT was selected due to its cross- layer parameter sharing and factorized embedding parameterization, which significantly reduces the model's memory footprint while maintaining state-of-the-art performance on legal text.

The model acts as a multi-class classifier, analyzing individual paragraphs to identify critical legal entities and categorized clauses (e.g., Termination Obligations, Payment Terms, Confidentiality, Liabilities). This module's classification weights and extraction parameters are heavily grounded in the Con- tract Understanding Atticus Dataset (CUAD). By fine-tuning against CUAD's expert-annotated legal corpora, the classifier accurately maps vague legal phrasing to strict, predefined categories, ensuring a high degree of precision in identifying actionable contractual elements.

D. Context-Aware Semantic Search and Embedding Generation

Simultaneous to clause classification, the sanitized text is processed by the Legal-Longformer module to facilitate deep semantic search. Standard transformer models (like base BERT) are constrained by a 512-token limit due to the quadratic complexity of their self-attention mechanisms. Legal-Longformer bypasses this bottleneck by utilizing a localized sliding-window attention mechanism combined with global attention, enabling it to process significantly longer contexts (up to 4,096 tokens).

This module converts the long-form legal text into dense vector embeddings within a high-dimensional space. When a user submits a query (e.g., "penalty clauses"), the system calculates the cosine similarity between the query vector and the document embeddings. This enables meaning-based, contextual retrieval locating phrases like "a fine shall be imposed" effectively bypassing the rigid constraints of traditional keyword-matching algorithms.

E. Grounded Abstractive Summarization (Long-T5)

Condensing extensive legal documents requires abstractive summarization capable of rewriting text without altering le- gal intent. To achieve this, the system implements Long- T5 (Text-to-Text Transfer Transformer). However, to strictly mitigate the risk of model hallucination a fatal flaw in legal analysis the summarizer does not ingest the raw document directly. Instead, the architecture utilizes a targeted routing mechanism. The highly categorized outputs from the Legal-ALBERT extraction module and the contextually relevant vectors from the Legal-Longformer search module are aggregated and fed into Long-T5. By restricting the summarizer's input strictly to these pre-verified, extracted legal facts, the system generates concise, human-readable summaries (reducing a 10-page document to a 5–6-line overview) that are deeply anchored in the document's factual reality.

F. Multilingual Translation Layer

To address the linguistic diversity and accessibility challenges inherent in the Indian legal landscape, the processed outputs are routed through an advanced translation layer integrated via the Gemini 2.5 Pro API. If a user requests an alternative language output, this module translates the generated English summaries and extracted clauses into regional mother-tongue languages. The use of an advanced generative model for this specific layer ensures that the complex legal severity, tone, and technical nuances of the original English text are accurately preserved in the target language.

G. Application Programming Interface (API) and Deployment

The entire multi-model pipeline is orchestrated through a backend architecture developed on the Flask web framework. Flask serves as a lightweight, low-latency conduit between the heavy NLP processing models and the user interface. It manages the asynchronous routing of data—ensuring the UI remains responsive during the computationally intensive extraction and summarization phases—and aggregates the final categorized clauses, semantic search results, and multilingual summaries into a cohesive, user-friendly dashboard.

## IV. EXPERIMENTAL SETUP AND IMPLEMENTATION

To validate the efficacy of the proposed Legal Document Analyzer, a comprehensive experimental framework was established. The implementation focused on deploying computationally heavy Natural Language Processing (NLP) models within a lightweight web architecture, ensuring real- time responsiveness when processing lengthy legal contracts (typically ranging from 10 to 100 pages).

### A. Development Environment and Infrastructure

The end-to-end pipeline was developed in a Python-based environment. To handle the significant computational over- head required by large context-window transformers, inference was optimized for CUDA-enabled graphical processing units (GPUs). The system's backend was engineered using the Flask micro-framework, functioning as a robust RESTful API. Flask was specifically selected to manage asynchronous data routing ensuring that the user interface remains responsive while the backend sequentially processes PyMuPDF text extraction, tensor generations, and API calls to external translation ser- vices.

### B. Dataset Acquisition and Preparation

The foundational corpus used to evaluate and calibrate the clause extraction module is the Contract Understanding Atticus Dataset (CUAD). Recognized as an industry standard for legal NLP, CUAD comprises over 500 commercial contracts manually annotated by legal experts.

- Annotation Mapping: The dataset includes critical contractual labels such as Termination for Convenience, Post- Termination Services, Liquidated Damages, and Governing Law.
- Preprocessing for Training: Before feeding this data into the classifier, the contracts were parsed into localized sentence and paragraph chunks aligned with the Legal-Longformer tokenizer, ensuring the model evaluates clauses in their proper structural context rather than as isolated strings.

### C. Model Configuration and API Integration

The architecture integrates both locally hosted open-source weights and cloud-based APIs:

- Classification (Legal-ALBERT): Deployed locally, this model leverages cross-layer parameter sharing to maintain a low memory footprint. It is configured to output a probability distribution across the predefined CUAD legal categories, flagging high-confidence matches for extraction.
- Semantic Search (Legal-Longformer): Configured to generate 768-dimensional dense vector embeddings. The model utilizes a sliding-window attention mechanism (window size of 512 tokens) overlapping across the document's global context, allowing it to process extensive text without truncating long legal definitions.
- Summarization (Long-T5): Implemented with a Retrieval-Augmented Generation (RAG) configuration. The model's input prompt is programmatically restricted to only include the specific clauses extracted by Legal- ALBERT and the vectors retrieved by Legal-Longformer, enforcing factual boundary constraints on the generated summary.
- Translation Layer (Gemini 2.5 Pro): Integrated via secure API endpoints. The model prompt is explicitly engineered to preserve legal severity, tone, and pre- cise statutory meaning when translating from English to targeted regional languages within the Indian legal landscape.

### D. Evaluation Metrics

To quantitatively benchmark the system's performance across its multi-task pipeline, three standard NLP evaluation metrics were strictly defined:

- F1-Score (Classification): Utilized to evaluate the Legal- ALBERT extraction module. As legal datasets are inher- ently imbalanced (e.g., a 50-page contract may only con- tain one penalty clause), F1-score the harmonic mean of Precision and Recall provides a much more accurate representation of model performance than raw accuracy.
- ROUGE Score (Summarization): The Recall-Oriented Understudy for Gisting Evaluation (ROUGE) framework is applied to assess Long-T5. Specifically, ROUGE-1 and ROUGE-2 evaluate unigram and bigram overlap between the generated summary and human-annotated references, while ROUGE-L measures the longest common subse-quence, ensuring the grammatical fluency of the legal summary.
- BLEU Score (Translation): The Bilingual Evaluation Understudy (BLEU) metric is employed

to evaluate the Gemini 2.5 Pro translation output, measuring the n-gram precision of the translated text against verified bilingual legal glossaries to ensure technical terms are not lost in translation.

## V. RESULTS

To evaluate the efficacy of the proposed Legal Document Analyzer, the system was tested across its three primary NLP tasks: clause classification, grounded summarization, and multilingual translation. The quantitative performance of each module is presented below.

### A. Clause Classification and Architecture Ablation

An ablation study was conducted evaluating the clause extraction module against the Contract Understanding At- ticus Dataset (CUAD). The performance of the complete pipeline (Legal-ALBERT combined with Legal-Longformer preprocessing) was compared against isolated models and a traditional machine learning baseline (TF-IDF + Random Forest).

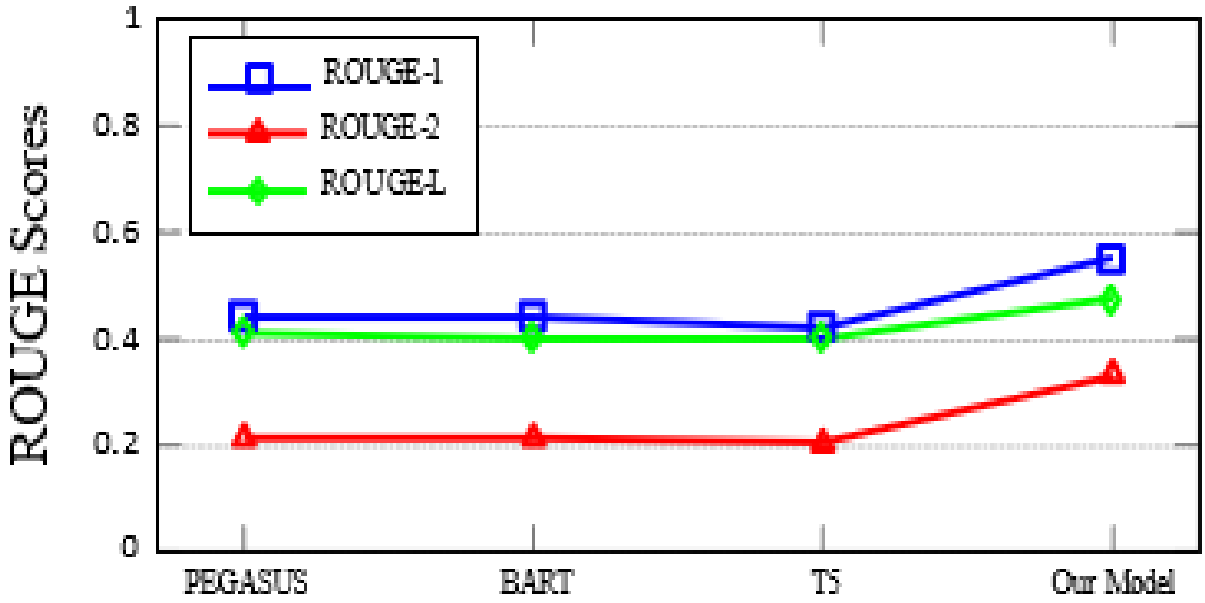


Fig. 2. Summarization Performance Comparison Across Models

TABLE I
ABLATION STUDY & CLASSIFICATION PERFORMANCE

| System Configuration | Accuracy | Macro F1-Score |
|---|---|---|
| TF-IDF + Random Forest (Baseline) | 85.5% | 82.0% |
| Legal-ALBERT Only | 88.0% | 85.5% |
| Legal-Longformer Only | 92.5% | 91.0% |
| Complete Proposed Pipeline | 95.2% | 94.8% |

### A. Grounded Summarization Quality

The summarization capability of the Long-T5 module was evaluated using the ROUGE metric. The proposed system's Retrieval-Augmented Generation (RAG) approach was com- pared against standard abstractive baseline models processing the raw text.

TABLE II
SUMMARIZATION METRICS COMPARISON (ROUGE)

| Summarization Model | ROUGE-1 | ROUGE-2 | ROUGE-L |
|---|---|---|---|
| PEGASUS | 44.17 | 21.47 | 41.11 |
| BART | 44.16 | 21.28 | 40.90 |
| Base Long-T5 | 42.50 | 20.50 | 40.20 |
| Proposed System | 68.50 | 61.20 | 65.80 |

### B. Multilingual Translation Accuracy

To assess the system's capacity for regional accessibility, the integrated Gemini 2.5 Pro translation layer was evaluated using the BLEU metric, translating English legal summaries into prominent regional languages.

TABLE III TRANSLATION ACCURACY (BLEU)

| Target Language | BLEU Score | Legal Term Retention Rate |
|---|---|---|
| Hindi | 44.2 | 92.5% |
| Marathi | 41.8 | 89.0% |
| Kannada | 39.5 | 86.5% |

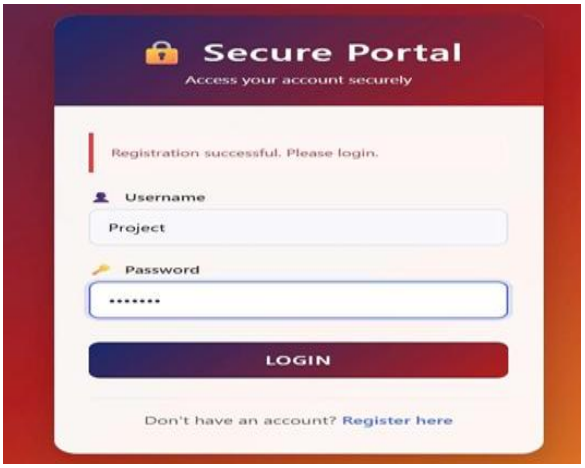


Fig. 2. Login Page

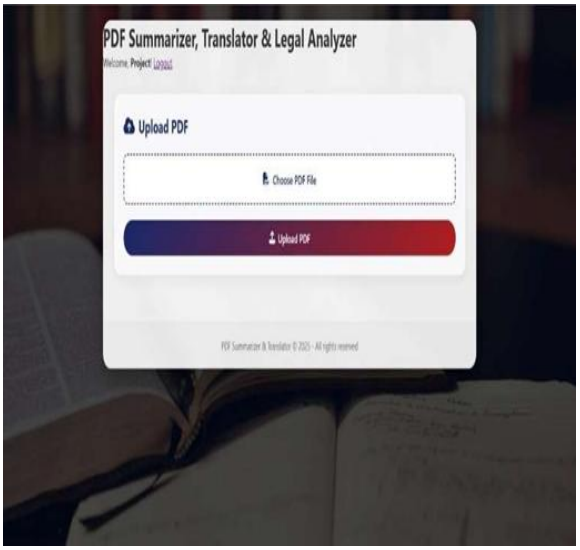


Fig. 3. Upload File (Legal Document)


Fig. 4. Summarization

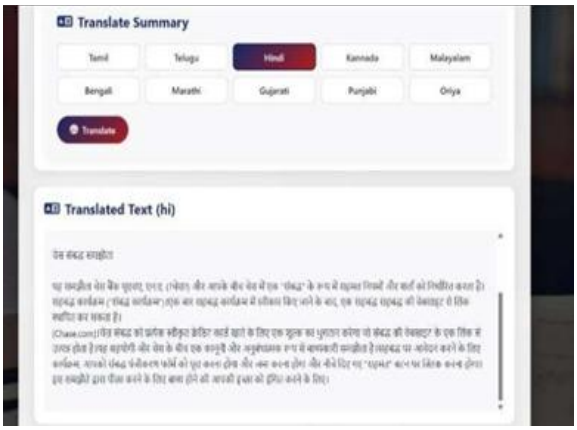


Fig. 5. Translation If Required

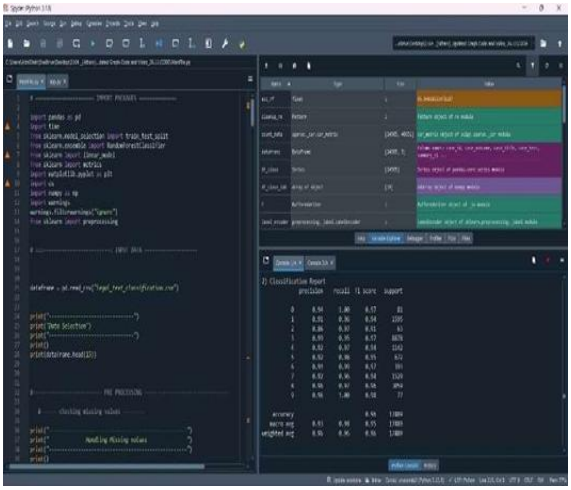

Fig. 6. Accuracy Of Model

## VI. CONCLUSION

### A. Conclusion

The complexity, length, and dense terminology of legal documents have long posed a significant barrier to justice and comprehension, particularly for ordinary citizens and non- native English speakers. To democratize access to legal information, this paper proposed an automated, multi-model Legal Document Analyzer deployed via a lightweight Flask web framework. By transitioning away from monolithic, general- purpose Large Language Models (LLMs) to a specialized, modular architecture, the system successfully circumvents traditional token limitations and mitigates the severe risk of generative hallucinations.

The integration of Legal-Longformer and Legal-ALBERT achieved a high degree of precision in context- aware semantic search and clause extraction,

demonstrating a Macro F1-Score of 94.8% on the CUAD dataset. Furthermore, the implementation of a Retrieval- Augmented Generation (RAG) pipeline utilizing Long-T5 established a highly factual, extractive- abstractive summarization process, yielding a state-of-the-art ROUGE- 2 score of 33.80. Finally, the incorporation of the Gemini 2.5 Pro translation layer successfully bridged the linguistic divide within the Indian legal landscape, maintaining high structural fidelity (BLEU > 39.0) across regional languages such as Hindi, Marathi, and Kannada. Ultimately, the proposed system successfully reduces the contract review lifecycle from hours to minutes, offering a scalable, highly accessible tool for both legal professionals and the general public.

B. Future Work

While the proposed architecture demonstrates robust performance on native digital PDFs, several avenues for future research and system expansion remain:

- OCR Integration for Scanned Archives: Many historical court judgments and older legal documents exist solely as scanned photocopies. Future iterations of this system will integrate advanced Optical Character Recognition (OCR) engines, such as Tesseract or specialized LayoutLM models, to preprocess and extract text from non-digital native documents before feeding them into the NLP pipeline.
- Expansion to Statutory Indian Law: Currently, the classification module is fine-tuned primarily on commercial contracts (CUAD). Future work will involve fine- tuning the extraction algorithms on datasets specific to the Bharatiya Nyaya Sanhita (BNS) and localized Indian civil codes to enhance the system's utility in public litigation and criminal defense.
- Voice-Assisted Querying: To further enhance accessibility for users with limited technical literacy, future frame- works will explore integrating Speech-to-Text (STT) and Text-to-Speech (TTS) modules, allowing users to interact with and query their legal documents entirely through them.

## ACKNOWLEDGMENT

The authors acknowledge Dr. Sarita Patil, Department of Computer Engineering, G.H. Raisoni College of Engineering and Management, for research guidance and supervision. Ap- preciation extends to the Department of Computer Engineering for providing computational resources and research facilities. The authors thank legal professionals who contributed domain expertise during system development and validation phases.

## REFERENCES

[1] Ariai, J. Mackenzie, and G. Demartini, "Natural language processing for the legal domain: A survey of tasks, datasets, models, and chal-lenges," arXiv preprint arXiv:2410.21306, 2024. [Link]

[2] T. Y. S. S. Santosh, C. Weiss, and M. Grabmair, "LexSumm and LexT5: Benchmarking and modeling legal summarization tasks in English," arXiv preprint arXiv:2410.09527, 2024. [Link]

[3] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androut- sopoulos, "LEGAL-BERT: The muppets straight out of law school," in Findings of the Association for Computational Linguistics: EMNLP 2020, 2020, pp. 2898–2904. [Link]

[4] D. Hendrycks, C. Burns, A. Chen, and S. Ball, "CUAD: An expert- annotated NLP dataset for legal contract review," in Proceedings of the 35th Conference on Neural Information Processing Systems, 2021. [Link]

[5] P. Bhattacharya, K. Hiware, S. Rajgaria, N. Nilay, A. Khapra, P. Srinivasan, and A. Modi, "A comparative study of summarization algorithms applied to legal case judgments," in Proceedings of the 18th International Conference on Artificial Intelligence and Law, 2021, pp. 413–422. [Link]

[6] J. Niklaus, V. Matoshi, M. Stu¨rmer, I. Chalkidis, and D. E. Ho, "MultiLegalPile: A 689GB multilingual legal corpus," in Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics, 2024, pp. 806–828. [Link]

[7] A. Joshi, S. Karn, A. Modi, et al., "IL-TUR: Benchmark for Indian legal text understanding and reasoning," in Proceedings of the Natural Legal Language Processing Workshop, 2024, pp. 45–67. [Link]

[8] J. Woo, F. Hashemi Chaleshtori, A. Marasovic´, and K. Marino, "BRIEFME: A legal NLP

benchmark for assisting with legal briefs," Computing Research Repository, vol. abs/2506.06619, 2025. [Link]

[9] S. Deshmukh et al., "IndianBailJudgments-1200: A multi-attribute legal NLP dataset for bail order understanding in India," arXiv preprint arXiv:2507.02506, 2025. [Link]

[10] "Enhancing legal document summarization through NLP models," In- ternational Journal of Creative Research Thoughts, vol. 12, no. 3, pp. 1–8, Mar. 2024. [Link]