

Towards Risk-Aware Legal NLP: Benchmarking Transformer Models and Proposing an Attention-Based Anomaly Detection Framework

Atharva Dhuri

Department of Computer Engineering
Mukesh Patel School of Technology
Management & Engineering, SVKM's
NMIMS, Mumbai, India
atharva.dhuri45@nmims.in

Manisha Tiwari

Department of Computer Engineering
Mukesh Patel School of Technology
Management & Engineering, SVKM's
NMIMS, Mumbai, India
manisha.tiwari@nmims.edu

Shrivardhan Wagh

Department of Computer Engineering
Mukesh Patel School of Technology
Management & Engineering, SVKM's
NMIMS, Mumbai, India
shrivardhan.wagh38@nmims.in

Yash Narayan

Department of Computer Engineering
Mukesh Patel School of Technology
Management & Engineering, SVKM's
NMIMS, Mumbai, India
yash.narayan45@nmims.in

Pragati Khare

Department of Computer Engineering
Mukesh Patel School of Technology
Management & Engineering, SVKM's
NMIMS, Mumbai, India
pragati.shrivastava@nmims.edu

Abstract—The rising volume and complexity of legal documents make manual review and keyword search inefficient for contextual understanding. This paper proposes an Integrated Legal Document Analysis System that unifies summarization, semantic retrieval, anomaly detection, and conversational reasoning using transformer models such as BART, Legal-BERT, and E5. A Retrieval-Augmented Generation (RAG) layer enhances factual accuracy, while an attention-based Clause-Level Deviation Profiling Framework detects irregular clauses via pseudo-perplexity, semantic drift, and contextual cohesion scoring. Results show that the fine-tuned BART model achieved a ROUGE-L of 0.3454 and BERTScore-F1 of 0.8682, outperforming baselines in coherence and factuality. The fused anomaly model effectively highlights complex or inconsistent clauses, achieving over a certain threshold in qualitative evaluation. Overall, the framework provides a scalable and explainable pipeline for automated legal document analysis and clause-level deviation detection.

Keywords—Legal NLP; Retrieval-Augmented Generation (RAG); Legal Document Summarization; Semantic Clause Search; Anomaly Detection; Clause Deviation Profiling; Risk Scoring; Transformer Models; Explainable AI; Legal Information Retrieval.

I. INTRODUCTION

A large number of professionals face the difficulty of managing legal data, which continuously seems to grow in size and complexity. These documents, which vary from contracts, court records, and laws, are stored digitally in a large number of files, which is difficult to navigate through

simply ordinary means. Simple keywords or rule-based searches are no longer enough when it comes to finding documents with a deeper meaning at a contextual level [1], [2], [4]. As documents continue to become more complex and domain specific, lawyers and the people working under them have to manually read, compare, and assess various clauses to understand these documents. They also have to watch out for any potential risks, this approach is not suited for the modern scale of operations as it ends up taking a lot of time and effort on their end.

The latest progress in the field of Artificial Intelligence has resulted in models like BART, T5, and LegalBERT, [3], [6], [7] which are used to assist in reading and summarizing large documents, retrieving clauses, and validating accuracy through retrieval. These models help in summarizing long judgments, while maintaining the accuracy of the generated data [8], [9], [10]. Retrieval Augmented Generation is another factor to consider in the field of Artificial Intelligence. RAG models help ensure the answers are accurate by fact-checking the documents that are stored in their database [9], [19]. However, there is a lack of systems fully utilizing all these advancements in a unified format, to detect the potential risks, or answer questions in real time [15, [17]].

Current legal datasets help classify clauses, provide pretrained LLMs and set the benchmarks, but are unable to determine the risks of each clause [15], [17]. For instance, terms like “one-sided indemnity” or “never-ending non-

compete" end up going unnoticed [17], [18]. As seen, current systems lack the ability to extract semantic clauses when the wordings are applied in a different manner.

To address these gaps, this study proposes a system that can generate retrieval augmented summarization, search clauses based on intent, detect potential deviation of clauses, and perform dialogue-based reasoning. By integrating clause embedding with vector similarity search and using an attention-based deviation model, the system is able to identify clauses that differ from standard patterns and also allows users to explore them through dialogue [9]. This framework provides a solid, scalable foundation for legal document analysis from summary to clause deviation and decision support through context-based retrieval.

II. LITERATURE REVIEW

The application of Natural Language processing in the legal domain continues to grow to the point that it has led to a variety of studies. These studies cover summarization of legal documents, clause tagging, and anomaly detection in the legal context.

A. Foundational Surveys and General Legal NLP

TABLE I. FOUNDATIONAL SURVEYS AND GENERAL LEGAL NLP

Author(s), Year	Study Details		
	Objective/ Contribution	Techniques / Models Used	Gaps / Limitations
Jonnalagadda et al., 2025 [1]	Comprehensive overview of text summarization across domains, including legal.	Rule-based (TF-IDF, TextRank), ML (SVM, NB), Seq2Seq (RNN/LSTM), Transformers (BERTSUM, T5, BART, PEGASUS), GPT models.	Limited legal focus, no RAG, stance detection, or risk awareness.
Chalkidis et al., 2024 [2]	Survey of NLP tasks, datasets, and challenges specific to the legal domain.	Pretrained LLMs (BERT, RoBERTa, GPT, T5, BART), LegalBERT, CaseLaw-BERT, LexLM.	Few tools for explainability, limited adoption of RAG, lack of stance/sentiment aware NLP.
Mondal et al., 2025 [3]	Survey on legal summarization with >120 papers reviewed.	Extractive (TextRank, BM25, Legal-BERT), Abstractive (BART, T5, Pegasus, LED, Longformer).	No stance detection, risk flagging, or interactive systems.

Author(s), Year	Study Details		
	Objective/ Contribution	Techniques / Models Used	Gaps / Limitations
Rani, 2023 [4]	Survey of legal document summarization methods.	Extractive (TF-IDF, LexRank, Naïve Bayes), Abstractive (RNN/LSTM, Transformers).	No mention of RAG, multi-document summarization, or semantic search.
George et al., 2025 [5]	Systematic review of legal document analysis software.	TF-IDF + Cosine Similarity, TextRank, BERT, T5, GPT-2.	No clause-level tagging, risk scoring, or scalability.

B. Legal Summarization Techniques & Systems

TABLE II. LEGAL SUMMARIZATION TECHNIQUES AND SYSTEMS

Author(s), Year	Study Details		
	Objective/ Contribution	Techniques / Models Used	Gaps / Limitations
Shen et al., 2023 [6]	Proposed a dual-stage, argument-aware summarization pipeline for long legal opinions.	Extractive (contrastive BERT ranking), Abstractive (BART).	Limited to judicial opinions, no stance/risk analysis, not applicable to contracts or policies.
Yang et al., 2023 [7]	Introduced STRONG, a structure-controllable summarization system for legal texts.	Flan-T5 with structure planning + conditional generation.	No clause-level analysis or semantic search; lacks risk awareness.
Nguyen et al., 2023 [8]	Developed an end-to-end system for Vietnamese legal case summarization.	TextRank (extractive), Multilingual BERT, GPT-2, BART, T5.	Language-specific, no RAG or stance detection, limited generalizability.
Sinha et al., 2025 [9]	Enhanced legal summarization through Retrieval-Augmented Generation (RAG) with domain-specific adaptation.	Dense retrievers (DPR) + generative LLMs (BART, T5).	High inference cost; no interactive querying or clause-level analysis.
Agarwal et al., 2024 [10]	Built a multi-step summarization pipeline for very long regulatory documents.	LongT5, LED, Fusion-in-Decoder (FiD).	No risk or anomaly detection; lacks interactivity or explainability.

Author(s), Year	Study Details		
	Objective/ Contribution	Techniques / Models Used	Gaps / Limitations
Gupta et al., 2024 [11]	Comparative evaluation of transformer models for legal summarization.	T5-small, PEGASUS, BART-large (fine-tuned).	Model bias risks: smaller T5 versions struggle with long texts.
Rakesh, 2024 [12]	General advancements in legal summarization using NLP + ML.	TextRank, TF-IDF, Naïve Bayes, Logistic Regression.	Focused mainly on extractive methods, with limited deep learning integration.
Wagh et al., 2023 [13]	Proposed NLP-based pipeline for legal case summarization (tender/legal documents).	Extractive (Gist), Abstractive (Legal-Summ)	No use of LLMs, lacks interactive query-based summaries.
Yadav et al., 2025 [14]	Built a hybrid legal document summarizer with a user interface.	BART (extractive) + T5 (abstractive), hybrid weighted output.	Preprocessing risks loss of context; limited to text inputs.

C. Clause Classification, Risk & Semantic Awareness

TABLE III. CLAUSE CLASSIFICATION, RISK IDENTIFICATION, AND SEMANTIC AWARENESS

Author(s), Year	Study Details		
	Objective/ Contribution	Techniques / Models Used	Gaps / Limitations
Bommasani et al., 2023 [15]	Clause classification in contracts (SemEval-2023).	LegalBERT, RoBERTa, DistilBERT	No risk-weighting or anomaly scoring.
Khan et al., 2025 [16]	Legal analysis using LLMs for accuracy and risk mitigation.	TF-IDF, TextRank, BERT, T5	Lack of clause-level risk tagging.
ContractEv al, 2025 [17]	Benchmark for clause-level risk identification.	DeepSeek, LLaMA, Gemma, Qwen	LLMs lag in precision, missing clauses.

Author(s), Year	Study Details		
	Objective/ Contribution	Techniques / Models Used	Gaps / Limitations
ACORD, 2025 [18]	Expert annotated dataset for clause retrieval.	BM25, MiniLM, Cross-Encoders	Focused on relevance, not anomaly.
LexRAG, 2025 [19]	Multi-turn legal consultation RAG benchmark.	BM25 + GPT-4o + LLaMA	No multilingual coverage.
NyayaAnuman & INLegalLlama, 2024 [20]	Introduced an Indian legal judgment dataset and a domain specific LLM for decision prediction.	INLegalLlama (fine-tuned LLaMA)	Focused on judgment prediction, not anomaly scoring.

D. Anomaly Detection and Fraud Prevention

TABLE IV. ANOMALY DETECTION AND FRAUD PREVENTION IN LEGAL AND DOCUMENT ANALYSIS

Author(s), Year	Study Details		
	Objective/ Contribution	Techniques / Models Used	Gaps / Limitations
Goyal et al., 2025 [21]	Anomaly detection in tax filings.	OCR + Bi-LSTM + iForest	Jurisdiction-specific.
Dey et al., 2025 [22]	Multimodal fraud detection for document verification.	GNN + attention fusion	High inference time.
ScienceDirect, 2024 [23]	Anomaly detection in subjective text reviews.	MPNet + Autoencoder	Limited to short text.
TAD-Bench, 2025 [24]	Benchmark for embedding based text anomaly detection.	BERT, MiniLM, LOF, iForest	Limited hyperparameter tuning.
Arxiv (RAG-based), 2025 [25]	Real-time fraud detection with RAG LLMs.	Retrieval-Augmented Generation + LLM	No legal corpus.

Author(s), Year	Study Details		
	Objective/ Contribution	Techniques / Models Used	Gaps / Limitations
RAAD, 2025 [26]	Retrieval Augmented Anomaly Detection framework.	Vector-store feedback fusion	No multi-class anomaly handling.
Software Engineering Meets Legal Texts, 2024 [27]	Applied LLMs for automatic detection of “contract smells” and structural anomalies.	GPT-family, fine-tuned BERT	Focused on “smells,” not risk quantification.
Salazar et al., 2020 [29]	Proposed pseudo perplexity as a metric for linguistic irregularity in masked language models.	Masked Language Model Scoring (MLM)	Requires model-specific calibration.
Paul et al., 2022 [30]	Introduced InLegalBERT for domain adapted legal language understanding	Transformer fine-tuned on Indian legal corpora	Limited cross-jurisdiction transferability.
Wang et al., 2022 [31]	Developed E5, a contrastive embedding model for semantic cohesion and retrieval.	E5 base encoder, contrastive training	General-purpose, non-legal fine-tuning.

III. RESEARCH GAPS

The studies across Table I to IV show various contributions as well the gaps, where the studies show the transition from relying on simple rule based and extractive methods to using transformers like BERT and T5. However, the summaries were only at surface level, lacking the explanations at a deeper level. It is shown that models like LED, Flan T5 and BART have improved summaries but can make mistakes due to hallucinations, and lack of factual checking. It was found that many systems lacked reasoning at a clause level and there weren't any conversation-based interactions where the users can communicate with the system in a question-and-answer format. While there exist tools like CUAD, ContractEval and LexRAG for clause classification it does not account for potential risks. There were contributions that expanded into the Indian Legal field, but they did not offer explainable or risk aware clause

scoring. It is also observed how anomaly detection models like RAAD and TAD-Bench can find anomalies, but end up being domain specific and lacking transparency. This stressed on the need for clear explanations and understanding at a clause level and deviation-based analysis that is scalable. Overall, the existing modules operate individually, summaries, retrieval of clauses, and anomaly detection all operate as separate units. These gaps stress on the need for a unified, explainable architecture that can use semantic retrieval, clause level anomaly detection, and conversational reasoning within a single pipeline.

IV. PROPOSED METHODOLOGY

In the field of legal natural language processing, there are several frameworks that help in processing legal documents. Retrieval Augmented Generation (RAG) plays an important role in this system as it is used by these frameworks to gain accurate information which is fact checked. The framework also utilizes clause level deviation profiling, to detect complex sentences worth checking. The proposed Legal Document Analysis System combines the mentioned frameworks in order to produce accurate summaries, improve clause level reasoning, and include risk awareness.

As illustrated in Figure 1 and logic outlined in Algorithm I, the architecture integrates modules that assist in summarization, semantic retrieval, detecting deviated clauses, and conversation-based queries. This workflow addresses the gaps found in the literature review. The gaps include a lack of risk awareness, contextual retrieval of clauses, and a unified RAG supported reasoning across multiple types of legal texts [1], [2], [3], [9], [15].

Algorithm 1. Legal Document Analysis using Retrieval-Augmented Transformer Architecture

Input: Legal documents D = {d₁, d₂, ..., d_n}

Output: Summaries S, Clause Risk Levels R, Query Answers A

BEGIN

// (1) Document Ingestion and Preprocessing

FOR EACH document d IN D DO

IF d.format ∈ {PDF, Image} THEN

text ← Apply_OCR(d)

ELSE

text ← Extract_Text(d)

```

    END IF

    text ← Clean(text)      // Remove headers,
    footers, spaces

    tokens ← Normalize_And_Tokenize(text)

END FOR

// (2) Clause Segmentation and Structuring

FOR EACH text IN D DO

    clauses ← Segment_Text(text, rules,
    transformer_cues)

    FOR EACH clause c IN clauses DO

        Assign_Metadata(c, Clause_ID, Section,
        Token_Length)

    END FOR

END FOR

// (3) Semantic Clause Search

FOR EACH clause c IN clauses DO

    embedding[c] ← Encode(c, model =
    LegalBERT/SentenceBERT)

    Store(embedding[c], VectorDB = Qdrant/FAISS)

END FOR

// (4) Summarization Engine

FOR EACH document d IN D DO

    context_clauses ← Retrieve_Relevant_Clauses(d,
    VectorDB)

    summary[d] ← Summarize(context_clauses, model
    = BART)

    S ← Generate_Summary(summary[d])

END FOR

// (5) Conversational Retrieval-Augmented Legal
Assistant

WHILE user_query EXISTS DO

    query_embedding ← Encode_Query(user_query)

    top_k_clauses ← Retrieve_TopK(VectorDB,
    query_embedding)

    answer ← Generate_Answer(top_k_clauses, model
    = fine-tuned T5)

    Maintain_Context(user_query, answer)

END WHILE

// (6) Clause-Level Risk and Anomaly Detection

FOR EACH clause c IN clauses DO

    p_score ← Compute_PseudoPerplexity(c,
    InLegalBERT)

    s_score ← Compute_SemanticDrift(c, LegalBERT)

    c_score ← Compute_ContextualCohesion(c, E5-
    base-v2)

    fused_score ← Reciprocal_Rank_Fusion(p_score,
    s_score, c_score)

    IF fused_score ≤ τ1 THEN

        Risk_Level[c] ← "Green" // Low deviation

    ELSE IF τ1 < fused_score ≤ τ2 THEN

        Risk_Level[c] ← "Yellow" // Moderate
        deviation

    ELSE

        Risk_Level[c] ← "Orange" // High deviation

    END IF

END FOR

// (7) System Integration & Output

Aggregate(S, R, A)

Display_Dashboard(Summaries = S, Risks = R,
Answers = A)

END

```

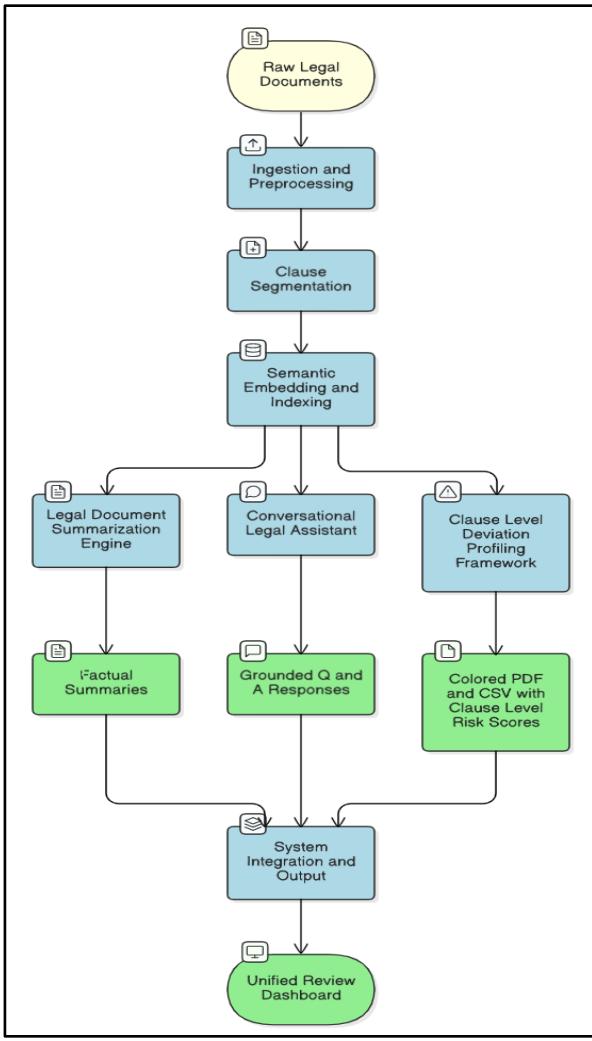


Fig. 1. End-to-end architecture of the proposed System.

A. Document Ingestion and Preprocessing

As shown in Figure 1 the pipeline starts with ingestion and preprocessing, in which raw text is taken in and converted into machine readable text. The raw text from legal documents used in the initial stage include, contracts, court judgements, or policy statements. To provide more accurate data while maintaining the legal content's integrity within the document, the system uses OCR based PDF extraction to remove unnecessary or repetitive content which is also reflected in Algorithm I Step 1 [13]. The process includes whitespace normalization and filtering of tokens. This helps remove repeated headers, footer, citations and page markers. This way there is a logical flow of clauses while ensuring the data's authenticity [4].

After cleaning, the input is tokenized, where the text is split into smaller units of tokens which are then stored in an organized structure for later segmentation and search related tasks for data retrieval. The system combines rule-based methods with transformers models for segmentation. The

system splits the documents into sections belonging to logical clauses varying from "Termination", "Liability", to "Confidentiality" clauses. For semantic analysis, this approach is crucial for better reasoning. This pipeline applies semantic embedding which gives the sections during which extra details are assigned in the form of metadata. The metadata contains information like clause id, respective section it belongs to and length of token which is stored in a structured manner as per logic outlined in Algorithm I Step 2. The combination of segmentation with semantic embedding helps the system understand and compare the relevant clause efficiently later during analysis.

B. Clause Segmentation and Semantic Indexing

The system relies on pretrained language models (LegalBERT, MiniLM and SentenceBERT) trained on legal datasets like CUAD [17] and other existing legal texts [18], [20], [33] for embedding the clauses to store in the system. These embeddings help the systems comprehend the actual meaning along with intent, in place of following rule-based approach and looking for keywords [17], [18]. These embeddings are stored in the form of vectors in databases like Qdrant and FAISS, this helps increase the speed of the search related tasks for retrieval which corresponds to logic outlined in Algorithm I Step 3. This way the system is able to find clauses based on intent despite the change in wording. For example, if the user gives "early termination", the system based on intent will find clauses like "unilateral withdrawal" or "premature dissolution". This module of the system serves as the foundation for RAG assistants, making searches based on intent and accuracy.

C. Legal Document Summarization Engine

Under this module, long context models like BART and LED [6] [10] go through fine tuning on legal datasets like Indian Legal Court Judgements (Abs) dataset (Sairam N., 2024) [33]. These models help in generating short, easily interpretable summaries from legal documents or contracts having many pages as illustrated in Figure I. The system applies a hybrid approach, where the most important and relevant clauses are extracted by using a salience-based approach which is then the summaries generated by the model. This helps improve understanding of the information while keeping the accuracy. The Retrieval Augmented Generation (RAG) layer helps in retrieving relevant clauses or references, this way it is able to reduce hallucination and prevent loss of information as described in Algorithm I Step 4 [9]. This module in the end generates factual summaries that are clear and easy to understand while keeping the legal meaning intact.

D. Conversational Retrieval-Augmented Legal Assistant

In order to make the document interactive, the Conversational Legal RAG assistant lets users engage in

conversation with the system where they ask questions in plain english as depicted in Figure I. On receiving the question the system uses the existing clause embedding and retrieval based system to find the relevant sections stored in its database. The stored text is then fed to the pre-trained T5-Large model [9], [19].

The model then provides clear and factual answers which refer to the exact clause IDs and sentence fragments used. Unlike standard summarization tools, this assistant supports continuous dialogue where it is able to keep the context of the overall context from the first question to the existing one aligning with Algorithm I Step 5. For example, if a user asks “Who can terminate the contract early?” followed by “Is compensation required for that termination?” the system will be able to answer based on overall conversation. This helps bridge the gap between standard search tools and advisory systems, similar to frameworks like ContractEval and LexRAG [17], [19].

E. Clause Level Deviation Profiling Framework

The proposed system uses a Clause Level Deviation Profiling Framework which through unsupervised means checks legal documents as summarized in Algorithm I Step 6. Instead of relying on labelled data having fixed clause types or supervised training, it detects rarely used text, text not within range of topic, or inconsistent sentences. It uses a group of channels that generate different scores, which on combining generates the quantitative output. This numerical score helps the framework understand how much the clause differs from the usual pattern and highlight the text accordingly. This can help fill the gap in current legal NLP systems that lack risk and anomaly flagging [24], [26], [27].

a) Framework Architecture: The framework integrates three scoring channels which assist each other, each checking a different type of deviation within the clause text as detailed in Table V.

TABLE V. INPUT CHANNELS FOR RECIPROCAL RANK FUSION (RRF) SCORING

Channel	Channel Characteristics		
	Model Used	Signal Captured	Output
Pseudo-Perplexity (PPL) [28]	InLegalBERT [29]	Linguistic irregularity and token-level uncertainty	Mean pseudo-perplexity per clause
Semantic Drift	Legal-BERT	Topical deviation from the document centroid	Euclidean distance in embedding space

Channel	Channel Characteristics		
	Model Used	Signal Captured	Output
Contextual Cohesion	E5-base-v2 [30]	Discourse discontinuity between neighboring clauses	Inverted cosine similarity

Each clause after segmentation from the processed document in the initial stage is fed through all the three channels. The system uses the z-score normalization to standardize the results for comparison. These scores are then combined using a method called Reciprocal Rank Fusion to make one final deviation score, as shown in Equation 1:

$$S_i = m \in \{PPL, SEM, CTX\} \sum 1/(k + R_m(i)), k=60 \quad (1)$$

where $R_m(i)$ represents the rank of clause i under detector m . The S_i is to measure if the clause is more different in wording, meaning or context compared to the rest of the document. If it crosses a threshold it will be marked, this helps the system spot potentially irregular or complex clauses that stand out, while reducing false alarms from normal language differences.

b) Interpretability and Visualization: To improve interpretation for the user, the system shows each clause’s deviation score visually using three colours ranging from green to orange, which reflects the intensity of the score with orange being the highest. The text highlighted in green shows that the text is matching with the overall document’s style with meaning as well. Orange shows the possibility of the text being inconsistent in phrasing, or flow. The system produces a coloured PDF and a CSV file with all deviation scores as shown in Figure I. During inspection, this helps the reviewers recognize the sections that follow the usual pattern and prioritise on sections that are potentially irregular.

V. RESULTS AND DISCUSSION

This section presents the testing with qualitative and quantitative analysis of Integrated Legal Document Analysis system, which uses summarization, reasoning based on retrieved clauses and clause level deviation detection. Each module in the system was tested to evaluate its performance in terms of speed, accuracy and interpretability. The evaluation focuses on BART summarizer to compress long legal documents into simple summaries, RAG model which uses Flan T5 and Sentence BERT embeddings to retrieve the clauses and provide factual replies, and clause level deviation framework which identifies unusual or complex clauses [4], [6], [10]. The discussion covers both qualitative and quantitative results to show how together, the components under this approach can provide accurate answers in a

transparent and an efficient manner for understanding legal documents.

A. Summarization Results

The performance of three summarization models from basic extractive methods, fine-tuned BART and Longformer Encoder Decoder [6], [10] were compared using ROUGUE and BERTscore to measure how accurate the summaries were in text and meaning. ROUGE - Recall-Oriented Understudy for Gisting Evaluation, uses n-grams and longest common subsequence matching to measure lexical overlap between generated and reference summaries. BERTScore uses contextual embeddings from a pretrained-BERT model to assess semantic similarity. The BERTScore-F1 aggregates precision and recall across all matched reference and generated tokens and gives a single semantic alignment score.

TABLE VI. IMPLEMENTATION COMPARISON OF SUMMARIZATION MODELS

Metric	Summarization Model Variants		
	Baseline (Extractive)	BART-base (Fine-tuned)	LED Longformer (GPU constrained)
Model	Centroid TF-IDF + MMR	Seq2Seq abstractive (BART)	Long document abstractive (LED)
Method	Extractive	Abstractive	Abstractive
Input Limit	Full document (no limit)	1024 tokens	4096 tokens (8GB GPU)
Output Limit	No token cap	256 tokens	256 tokens
ROUGE-1	0.5281	0.5635	0.4844
ROUGE-2	0.2923	0.2952	0.2399
ROUGE-L	0.2846	0.3454	0.2732
BERT_F1	0.6238	0.8682	0.8523
Observation	Simple and fast; lexical overlap only	Fluent summaries; limited context window	Handles long documents; slower but more coherent

The BART based model ended up having the best overall performance with a ROUGUE-L score of 0.3454 and BERTScore-F1 of 0.8682. It shows that this model is able to produce accurate summaries within the intended context. The summaries generated were coherent and easy to comprehend,

but the model was only able to process up to 1024 tokens at once, this limited the context window for the model. The extractive baseline which used TF-IDF and MMR required less time and was a simple approach, but it only captured information at a surface level and lacked the ability to comprehend the meaning of the content [4]. The LED Longformer, while being capable of handling long documents with a larger context window of 16384 tokens, suffered GPU hardware constraints which restricted the input limit to only 4096 tokens. Despite the model showing contextual understanding, due to these constraints it took more time to process and had lower performance scores in comparison to BART.

In the end, BART gave the best mix of accuracy, fluency, and speed for documents with lengthy texts. For text requiring a larger context window, LED Longformer can still be useful, but when combined with RAG framework, a fine-tuned BART model will be capable of handling long documents effectively. Thus, BART was selected as the primary model for summarization.

B. Retrieval-Augmented Generation

RAG is applied in this pipeline to help fix the limits of transformer-based summarization, this setup helps the system keep track of the overall context of the conversation that takes place between the user and the system. The system is able to find the required clauses based on meaning, and stay factually correct across multiple queries on the user's end. The RAG module works with BART, to help generate summaries that are factual and reduce the necessity of keeping the entire document in memory. Manual qualitative testing on several legal documents have shown that the outputs received were factually correct and easy to understand compared to normal summaries that were generated. The answers generated being linked to their source clauses make them more trustworthy and reliable.

When it comes to speed, the Qdrant database is used with a fast vector search (HNSW) method that handles larger documents smoothly, allowing faster delivery in real time. While detailed metrics like Precision@k or factual scores are yet to be used for evaluation, early qualitative results show gains in accuracy, relevance and factual grounding. Overall, the RAG based Flan-T5 model improves how well the system is able to comprehend and explain the answers relating to long legal documents.

C. Clause-Level Deviation Profiling

Experimental tests on several legal PDFs show that the majority of text has low deviation scores, showing that these documents follow the regular structure. Clauses with lists, cross references or sudden change in context have appeared to show higher scores. These high scores usually are on complex or detailed writing, not necessarily legal errors. The

combined score brings all detectors together to detect irregular text while reducing random noise, this paves the path for clause consistency before relying on RAG. The colour coded results help in quick identification of documents that need a closer review. This approach can also serve as a base for future research on legal risk or compliance analysis.

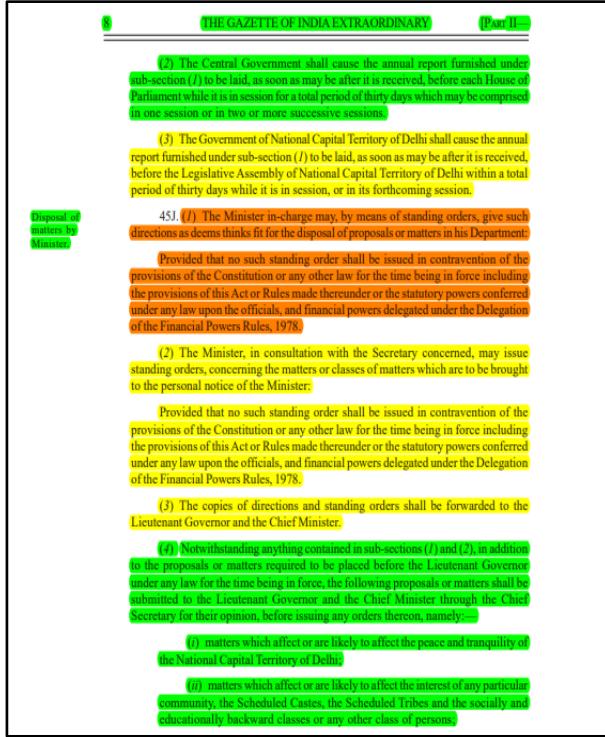


Fig. 2. Example of Clause Level Deviation Visualized in The Gazette of India [32]

VI. CONCLUSION

The entire integrated system which uses modules for summaries, retrieval of clauses and clause deviation, works better when integrated, as it leads to more accurate results which are easy to understand. The framework uses models like BART, Legal BERT [3], [6], InLegalBERT [30], and E5 [31] to summarize long legal texts, [7], answer context-based queries on taking the documents and conversation into account, and highlight clauses that differ from normal patterns. BART being fine-tuned produced the most balanced results with regards to comprehension and factual accuracy for summarization. While the BART module was able to summarize the legal text well it did not have enough context of the entire document, but on adding the RAG layer the results were better than before. The RAG layer helped in providing output based on the larger context window of the stored documents while reducing hallucinations [6], [9], [10]. The embeddings from Flan T5 and Sentence BERT were used in RAG to answer legal questions based on the context of the conversation correctly as well as the document [16]. The RAG layer achieved this by relying on the relevant clauses in

order to provide factual answers which helped increase the reliability of the overall system.

The deviation profiler acts as a quality checker, it looks for sentences that appear irregular, not coherent or misplaced and highlights them accordingly. While it may not detect anomalies, it helps reviewers recognize the sections worth looking into and what sections to prioritize. Qualitative tests on various documents have shown that these modules when combined work well together within the system as it is able to produce factual and interpretable answers through a modular pipeline. This workflow also helps reviewers also find irregular text through unsupervised means without any labelled data. The only drawback this system suffers is lack of available datasets to obtain numerical data for evaluation. Additionally, the lengthy documents lead to increase in the time required for processing, which affects overall speed. This is due to the combination of long context models [15], [17], [21-27]. Despite these challenges, the system's ability to provide easy to understand and factual answers make it a strong base for future research in explainable and risk aware legal NLP.

VII. FUTURE SCOPE

Future work should aim on improving the system's speed along with the ability to provide results that are more accurate and easier to comprehend, the improvements revolve around summarization, retrieval and clause level deviation. For large documents with long texts, hierarchical and discourse aware models can help create coherent and less repetitive content in generated summaries [6], [7], [10]. Adding Reinforcement Learning with Human Feedback (RLHF) can make summaries clearer and put it in parallel with legal reasoning [9], [16]. Additionally, building interactive dashboards that show users visual clause level insights will make the system more transparent and easier for legal experts to use and understand its content [23], [24], [27]. The current system suffers from few limitations as it depends on legal data from certain domains, which results in partiality and causes the results to be less general [24], [27]. Models like Legal-BERT and E5 also require a lot of computational power, which results in longer periods of time to compute the scores [15]. Since the framework has no labeled data, it relies on internal coherence metrics and validation from an expert in the legal field [24]. In the future the system can improve by adjusting its scores on the different types of clauses, adding supervised models trained with labeled data which can help transform deviation vectors into features for risk detection, or integrating it with the RAG pipeline in order to receive more accurate and context-based answers in relation to the deviation scores [17], [19], [26]. By training the system on multilingual and international datasets, we can expand the system to other languages and countries, which will make it applicable on a global scale [20], [24], [27]. These updates would make the system faster, more accurate and easier to use for legal document analysis.

REFERENCES

- [1] V. Jonnalagadda, et al., “A Survey of Text Summarization: Techniques, Evaluation, and Challenges,” ScienceDirect, 2025.
- [2] I. Chalkidis, et al., “Natural Language Processing for the Legal Domain: A Survey of Tasks, Datasets, Models, and Challenges,” arXiv preprint arXiv:2410.21306, 2024.
- [3] P. Mondal, et al., “A Comprehensive Survey on Legal Summarization: Challenges and Future Directions,” arXiv preprint arXiv:2501.17830, 2025.
- [4] A. Rani, “A Survey of Legal Document Summarization Methods,” International Journal of Advanced Research in Computer and Communication Engineering, 2023.
- [5] A. George, et al., “Natural Language Processing in Legal Document Analysis Software: A Systematic Review,” International Journal of Intelligent Research and Scientific Studies (IJRSS), 2025.
- [6] Y. Shen, et al., “Towards Argument-Aware Abstractive Summarization of Long Legal Opinions with Summary Reranking,” arXiv preprint arXiv:2306.00672, 2023.
- [7] J. Yang, et al., “STRONG – Structure Controllable Legal Opinion Summary Generation,” arXiv preprint arXiv:2309.17280, 2023.
- [8] H. Nguyen, et al., “A Deep Learning-Based System for Automatic Case Summarization,” arXiv preprint arXiv:2312.07824, 2023.
- [9] R. Sinha and K. S. Easwarakumar, “Optimizing Legal Text Summarization Through Dynamic Retrieval-Augmented Generation and Domain-Specific Adaptation,” Symmetry, vol. 17, no. 5, p. 633, 2025, doi: 10.3390/sym17050633.
- [10] P. Agarwal, et al., “Summarizing Long Regulatory Documents with a Multi-Step Pipeline,” in Proceedings of the NLLP Workshop, ACL Anthology, 2024.
- [11] S. Gupta, et al., “Enhancing Legal Document Summarization Through NLP Models: A Comparative Analysis of T5, Pegasus, and BART Approaches,” International Journal of Creative Research Thoughts (IJCRT), vol. 12, no. 4, 2024.
- [12] M. Rakesh, et al., “Advancements in Legal Document Processing and Summarization,” International Journal of Novel Trends and Innovation (IJNTI), vol. 2, no. 4, Apr. 2024.
- [13] S. Wagh, et al., “Legal Case Document Summarization Using NLP,” International Research Journal of Modernization in Engineering, Technology and Science (IRJMETS), vol. 5, no. 12, Dec. 2023.
- [14] A. Yadav, U. Chauhan, S. Zahra, and A. I. Abidi, “Legal Document Summarizer,” Preprints, Apr. 2025, doi: 10.20944/preprints202504.1960.v1.
- [15] R. Bommasani, et al., “SemEval-2023 Task 6: Clause Classification in Legal Contracts Using Pretrained Transformers,” in Proceedings of SemEval-2023, ACL Anthology, 2023.
- [16] M. Khan, et al., “Enhancing Legal Document Analysis with LLMs: Accuracy, Context Preservation, and Risk Mitigation,” Scientific Research Publishing (SCIRP), 2025.
- [17] ContractEval, “Benchmarking LLMs for Clause-Level Legal Risk Identification in Commercial Contracts,” arXiv preprint arXiv:2508.03080, 2025.
- [18] ACORD, “An Expert-Annotated Dataset for Legal Contract Clause Retrieval,” arXiv preprint arXiv:2501.06582, 2025.
- [19] LexRAG, “Benchmarking Retrieval-Augmented Generation in Multi-Turn Legal Consultation Conversation,” in Proceedings of the ACM Conference, 2025.
- [20] NyayaAnumana and INLegalLlama, “The Largest Indian Legal Judgment Prediction Dataset and Specialized Language Model for Enhanced Decision Analysis,” arXiv preprint arXiv:2412.08385, 2024.
- [21] A. Goyal, et al., “Anomaly Detection in Tax Filing Documents Using NLP Techniques,” EWA Direct, 2025.
- [22] A. Dey, et al., “AI-Driven Multimodal Anomaly Detection for Document Fraud Prevention,” ResearchGate Preprint, 2025.
- [23] S. Zhao, et al., “Explained Anomaly Detection in Text Reviews: Can Subjective Scenarios Be Correctly Evaluated?,” Expert Systems with Applications, ScienceDirect, vol. 235, 2024.
- [24] TAD-Bench, “A Comprehensive Benchmark for Embedding-Based Text Anomaly Detection,” arXiv preprint arXiv:2501.11960, 2025.
- [25] Advanced Real-Time Fraud Detection Using RAG-Based LLMs, arXiv preprint arXiv:2501.15290, 2025.
- [26] Retrieval Augmented Anomaly Detection (RAAD): “Nimble Model Adjustment Without Retraining,” arXiv preprint arXiv:2502.19534, 2025.
- [27] R. Zhang, et al., “Software Engineering Meets Legal Texts: LLMs for Auto Detection of Contract Smells,” SoftwareX, ScienceDirect, vol. 29, 2024.
- [28] Salazar, J., Liang, D., Nguyen, T. Q., & Kirchhoff, K. (2020). Masked Language Model Scoring. Proceedings of ACL 2020.
- [29] Paul, S., et al. (2022). Pre-trained Language Models for the Legal Domain: A Case Study on Indian Law (InLegalBERT). arXiv:2209.06049.
- [30] Wang, L., Yang, N., Huang, X., Jiao, B., Yang, L., Jiang, D., Majumder, R., & Wei, F. (2022). Text Embeddings by Weakly-Supervised Contrastive Pre-training (E5). arXiv.
- [31] Cormack, G. V., Clarke, C. L. A., & Buettcher, S. (2009). Reciprocal Rank Fusion Outperforms Condorcet and Individual Rank Learning Methods. Proceedings of SIGIR '09.
- [32] The Gazette of India (Extraordinary), Ministry of Home Affairs, May 22 2023. Available: <https://cdnbbsr.s3waas.gov.in/s380537a945c7aaa788ccfcdf1b99b5d8f/uploads/2023/05/2023052214.pdf>
- [33] Sairam N., Indian Legal Court Judgments (Abs) Dataset, Hugging Face, 2024. [Online]. Available: <https://huggingface.co/datasets/sairamn/in-abs-judgement-summary>