Grenze International Journal of Engineering and Technology, January Issue

Legal Contract Analyzer using LLM and RAG

Dr. S. Poornima[1], K Elankumaran[2], M Nandhana[3], B Parthasarathy[4] and V Sudharshan[5]
[1]Assistant professor, Dept. of IT, Coimbatore Institute of Technology, Coimbatore, Tamil Nadu, India
[2-5]UG Scholar, Dept. of IT, Coimbatore Institute of Technology, Coimbatore, Tamil Nadu, India
Emails: poornima.s@cit.edu.in, elankumarank27@gmail.com, nandhanamurugan2005@gmail.com, parthasarathybala509@gmail.com, sudhaveera67@gmail.com

***Abstract*— This project presents a Legal Contract Analyzer that helps users understand contracts, and identify risks, by applying a Retrieval-Augmented Generation (RAG) model, based on Large Language Models (LLMs). The program pulls in text from the uploaded PDF contracts using Docling and turns that text into smaller chunks, after which it embeds each chunk semantically using the Sentence-Transformers all-mpnet-base-v2 model. Once the embeddings are created, they are inserted into a FAISS database that facilitates the subsequent retrieval of each of the clauses. The program assesses the most relevant clauses to pre-set categories including, payment, liability, termination, and confidentiality for example, and then breaks the clauses down using the Zephyr-7B model on Hugging Face. The model summarizes the key terms, assesses possible risks, then makes recommendations on ways to mitigate those risks. By combining document analysis, semantic search, and AI-facilitated reasoning, the project eliminates manual effort, enhances risk detection, and provides a way for legal specialists and non-specialists to read and comprehend complex contracts with greater simplicity.**

***Key Words*— Legal Contract Analysis, Retrieval-Augmented Generation (RAG), Large Language Model (LLM), FAISS, Semantic Embeddings, Clause Extraction, Risk Assessment, Hugging Face, Docling, Contract Automation.**

I. INTRODUCTION

A. Legal Contracts

Legal contracts are an officially recognized agreements between two or more parties which provides the rights, responsibilities, and obligations of each party. Mismanagement of minor errors in such documents could result in serious financial loss or costly legal disputes. Organizations can handle hundreds or thousands of these contracts and documents ineffective due to time constraints, errors, and inefficiencies. It is rationale at this point, to provide an automated, AI-powered contract analysis solution that accesses and comprehends the contracts, and analyzes the results. The proposed Legal Contract Analyzer proposes using Large Language Models (LLMs) and Retrieval-Augmented Generation (RAG) methods to achieve exact, scalable, and explainable contract navigation and analysis.

B. Importance of Contract Analysis

Analysis of the contract ensures that the obligations and protections of the legal agreements are understood prior to signing the papers.

Grenze ID: 01.GIJET.12.1.150_1

Critical clauses such as Termination, Indemnification, Liability, and Confidentiality can have significant ramifications for potential legal and financial consequences. The practice of reading the contract using previous expertise, independent of automation, can take significant time and effort, where as an AI-based model could automate the extraction clause and semantically, write-of-risks clauses as well as classifying risk. The proposed model could provide transformation-based embeddings and LLM reasoning with algorithms to identify risky or undefined clauses, insure it is in accordance to a regulatory standard or two, and suggest better language.

C. Challenges in Contract Analysis

There are various challenges associated with traditional methods of contract analysis, which can limit their utility:

Legal Language: Contracts are often characterized by old-fashioned terms; convoluted, nested phrases; and unfathomable reference chains which are difficult for a standard NLP model to process.

Variability of Context: Some clauses (e.g. confidentiality) may appear in any number of iterations of linguistic form, which attempts to understand meaning in whatever surface text it is confronted.

Contextual Dependence: The risk level associated with a clause is considered in relation to the related sections (e.g. liability is a dependent factor to the indemnification language).

Legal Risk Detection: Identifying missing, ambiguous or biased clauses requires both semantic retrieval and logical reasoning.

The system outlined here can address these challenges through semantic embedding, FAISS-based retrieval and LLM-driven contextual inference, which will result in real-world legal documents solved accurately.

D. The Role of Large Language Models (LLMs)

Various Large Language Models (LLMs), including GPT, BERT, and Zephyr-7B, have revolutionized natural language understanding models by acquiring rich semantic relationships across modalities of text. As such, LLMs offer a dramatically free interpretive capability of various phrases found in legal language and how clauses differ contextually, as well as an understanding of clause intent. In the Legal Contract Analyzer, the LLM will interpret extracted clauses, identify potential risk(s), and eventually produce a brief, natural-language summary.

E. Retrieval-Augmented Generation (RAG) in Legal AI

RAG improves large language models (LLMs) by incorporating real, relevant data to inform their responses. It uses clauses from the CUAD (Contract Understanding Atticus Dataset), which are transformed into a vector space with models like all-MiniLM or all-mpnet-base-v2. This ensures that the system understands the deep semantic connections between different contractual terms. To efficiently find the most relevant examples from this vector space, a FAISS index is utilized, which then supplies these examples to the LLM for tasks like risk classification or generating advice on specific clauses. This retrieval-augmented approach connects static model knowledge with dynamic expertise in the field.

II. Literature Survey

A. Benchmarking Retrieval-Augmented Generation in Multi-Turn Legal Consultation Conversation

This paper [1] studies the effectiveness of RAG in multi-turn legal consultation conversations by evaluating different retrieval strategies for maintaining context across user interactions. Its main advantage is enhanced context awareness, while the drawback is its heavy computational cost and difficulty in handling long conversation histories. This paper was published by IEEE in March 2023.

B. Enhancing the Precision and Interpretability of RAG in Legal Technology

This paper [2] proposes methods to improve the precision and interpretability of RAG systems in the legal domain by showing why certain documents or clauses are retrieved. The major benefit is improved user trust and transparency, though the drawback is increased complexity and slower system response. This paper was published by ACM in April 2023.

C. Enhancing the Precision and Interpretability of RAG in Legal Technology

This paper [3] explores how LLMs combined with RAG can encode and analyze financial derivative contracts for risk detection and compliance verification. The approach offers domain-specific accuracy, particularly in finance law, but it is less applicable outside of financial contexts. This paper was published by IEEE in June 2023.

D. Case-Based Reasoning for RAG in LLMs for Legal Question Answering

This paper [4] integrates case-based reasoning (CBR) with RAG to improve legal question answering by retrieving similar past cases to justify responses. The key strength is better explainability through legal precedents, while the limitation is dependency on well-structured case databases. This paper was published by Springer in July 2022.

E. A Benchmark for Retrieval-Augmented Generation in the Legal Domain

This paper [5] introduces a benchmark dataset and evaluation framework for RAG systems in legal document understanding. Its main advantage is providing a reliable testing framework, while the limitation is that the benchmark does not fully cover all legal domains. This paper was published by IEEE in September 2022.

III. SYSTEM ARCHITECTURE

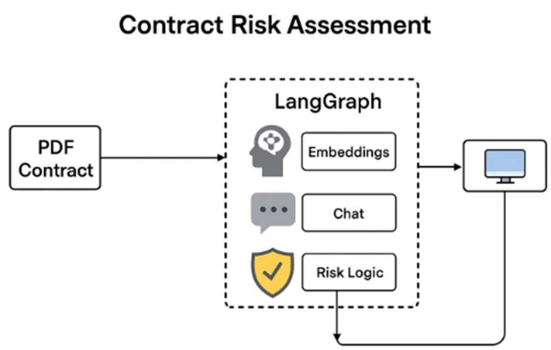

Figure. 1 System Architecture

The modular pipeline architecture of the Contract Risk Assessment Tool given in figure 3.1 is intended for automated contract analysis, risk assessment, and interactive user interaction. There are four primary parts to the architecture:

A. Contract Documents at the Input Layer

This layer forms the base of the whole contract risk assessment system. Users upload contracts in PDF format, which become the source of primary data to be analyzed. These documents contain complex legal terminologies, multiple clauses, and structured sections, which need to be precisely extracted before processing. For that, the system employs a PDF parsing tool like Docling, which reads and converts the uploaded PDF to clean, structured, and machine-readable text. The parser takes away non-text elements like headers, footers, and page numbers so that just the meaningful content of the contract remains. Segmentation of each clause and section is logically separated to maintain its legal context. Output from this layer is an organized and standardized representation of a contract, which can be easily understandable for the processing AI modules in the subsequent layers.

B. Workflow for LangGraph at the Processing Layer

The structured contract text retrieved from the input layer will now be processed via the LangGraph workflow, serving as the central AI engine of the system. LangGraph orchestrates various components that transform raw legal language into structured knowledge and risk insights.

Generator for Embeddings

This sub-module embeds the textual contents of contracts into dense vector embeddings using HuggingFace sentence transformer models. Each clause or paragraph is represented as a vector in a high-dimensional space, enabling semantic relationships between different segments of the contract to be grasped. This embedding representation facilitates advanced functionalities like semantic search and retrieval-based question answering, ensuring that similar clauses can be efficiently compared and analyzed across documents.

Module for Risk Logic

This module constitutes the core decision-making unit of the system. It applies legal and corporate risk assessment principles to evaluate the content of each clause. The module is designed around predefined risk categories such as confidentiality, indemnity, liability, and termination. It analyzes the clauses using both embedding-based retrieval and LLM-based reasoning to detect risky or non-compliant statements. Each clause is allocated a risk score, which designates the potential impact or severity. The module also provides suggestions and recommendations for the mitigation of identified risks, helping users understand how the clauses should be modified or redrafted for greater compliance and safety.

C. Output Layer

The output layer is responsible for presenting the outcome of the analysis in an intuitive and visually accessible way. It highlights the clauses of a contract based on assessed risk using colored color indicators, allowing users to rapidly identify critical areas. Along with highlighting these, it also provides structured summaries of each clause, showing the associated risk category, confidence score, and AI recommendations.

IV. IMPLEMENTATION

The Legal Contract Analyzer's implementation is geared towards transforming the proposed architecture into a full-scale, working application through the integration of Natural Language Processing (NLP) applications, Large Language Models (LLMs), and Retrieval-Augmented Generation (RAG) techniques. Each stage in the pipeline—from document preprocessing to risk classification and visualization—has been designed and developed in a manner that time, accuracy, scalability, and reliance on real-time interactions with the user is feasible.

A. Data Cleaning

Subjecting the data to cleaning and structuring for machine readability is the very first stage of the process. In-house, users upload contracts in either PDF or DOCX format. Operations are performed using a cutting edge parsing library called Docling, which is capable of processing documents that contain both text and wrapper rich or layout. Docling extracts text while also maintaining the order of clauses and the logical flows between clauses in text. Cleaning and preprocessing routines are applied.

B. Vector Creation

Following data preparation, each clause is translated into a semantic vector representation that represents the contextual meaning of each clause. The project utilizes the Sentence-Transformers all-mpnet-base-v2 model to generate embeddings, which was selected for its superior performance on semantic textual similarity (STS) tasks.

The resulting embeddings capture subtleties of legal semantics and normalize for storage purposes. A FAISS (Facebook AI Similarity Search) index is constructed to enable rapid nearest-neighbor searching across the CUAD dataset, which provides the knowledge base of annotated contract clauses. Each embedding is stored with metadata data, such as the clause ID, the name of the contract, and the risk label associated with each clause. This vector index allows the retrieval-augmented generation(RAG) architecture to ensure that when a new clause is processed, semantically similar clauses from the CUAD dataset can be retrieved in milliseconds for context analysis.

C. RAG Pipeline

The Retrieval-Augmented Generation (RAG) pipeline merges the semantic retrieval of relevant clauses with reasoning capabilities of a Large Language Model (Zephyr-7B) hosted directly on Hugging Face Inference API. When a user provides a contract, each clause embedding is compared to that of the FAISS index to return the top-k most relevant clauses from the CUAD dataset. The returned clauses are then provided as contextual evidence to the LLM through a tailored prompting scheme. The prompt provides high-level guidelines for the model to perform clause interpretation, risk classification, and suggestion development based on factual retrieved examples rather than isolated generation.

D. AI Risk Analysis and Suggestion Module

When the RAG pipeline generates responses, the AI Risk Analysis and Suggestion Module takes those prepared responses and re-organizes them into actionable output. The response from the LLM is tokenized into a variety of different components such as risk type, spans of risky clauses, confidence score to give high, medium, low confidence, and suggested rewording of clauses.

Each of the clauses will be identified into broader legal categories based on areas like Confidentiality, Termination, Indemnification, and Limitation of Liability.

E. Review and Dashboard Module

In the last phase of the project, we will visualize the analysis results on an intuitive Streamlit dashboard that functions as the user interface for the contract review process. The review dashboard provides the user interface to visualize each clause, its assessed risk level, highlighted risk spans, the data source of retrieved examples, and any AI-recommendation generated by the pretrained CUAD model. The user can filter clause comparisons or views by risk level (e.g. high, medium, low and export the results as PDF annotated reports or redlined DOCX reports for collaboration.

V. ALGORITHM

A. Preprocessing

The first step in analyzing the said document of the contract will involve reading the contract document and breaking it down into logical, manageable parts. The PdfReader Library processes the uploaded PDF, extracts the text from each page, and merges it into a continuous string. After extraction, there are often unnecessary whitespace, headers, or other formatting issues in the raw text. Hence, light cleaning and normalization have been performed to ensure text consistency.

Then, the system uses a regular expression to divide the contract into semantically meaningful pieces, usually clauses or numbered sections. This allows for segmentation of the contract in a natural way, with the further steps of risk assessment to take place at the clause level. This granularity is retained because usually each clause conveys a different legal meaning or obligation.

B. Embedding + Vector DB Setup

Generation of Embeddings and Vector Database Setup After text extraction, each clause is then embedded into a dense numerical representation (embedding) with the help of a pre-trained Sentence Transformer model (all-MiniLM-L6-v2), which is semantically-based and captures the meaning of sentences, not just the words. These embeddings are then indexed in a FAISS vector database, which allows for fast similarity search. Later, when another clause is being processed, it can locate similar clauses from previous data based on semantic closeness rather than keyword overlap. This vectorization step forms the foundation for retrieval-augmented generation and contextual comparison.

C. Risk Classification

In this stage, a light machine learning model is trained on classifying whether a clause is potentially risky or safe. An example model (like a Random Forest Classifier) is first trained on embeddings of labeled clauses, where each clause has been previously annotated as “risky” or “non-risky.” During analysis, the model will quickly predict a binary risk label for each new clause. This will serve as a pre-filter mechanism-helping to identify obviously high-risk clauses before invoking the more computationally expensive LLM analysis. The classifier provides fast, consistent results, but since legal language is highly contextual, it's complemented by the LLM for interpretability and precision.

D. RAG + LLM Integration

The core intelligence in the system comes from the combination of retrieval and large language model reasoning. Given a particular clause to review, its embedding retrieves the top-K most similar clauses from the FAISS index. These retrieved examples become contextual references against which LLM compares.This is followed by a well-structured prompt that includes the current clause, retrieved examples, and instructions for LLM to identify the risk level, category, and possible mitigation strategies. The model, such as GPT-4 or any other LLM, would go ahead and provide a detailed, yet concise explanation, also offering a confidence-based risk score.

E. Aggregation

Individual analysis of each clause is then combined to provide an overall contract-level summary of the results. Each record for a clause contains the risk score, category, and explanation. These are aggregated to produce the overall risk profile for the contract — for example, by taking the maximum clause risk or computing some weighted average across all sections. It does this by providing both a summary overview-for example, the total number of risky clauses and overall severity-and a granular breakdown that shows the users which of the clauses drive most of the risk profile of the contract. This will enable legal teams to prioritize their review and take corrective action.

VI. Experimental Results and Analysis

The experimental stage for the project of Legal Contract Analyzer intends to validate the efficacy, reliability, and usability of the proposed RAG-based LLM framework for automatic legal contract reviews. A number of experiments were conducted assessing the performance of each individual component: document extraction, embedding retrieval, clause similarity, and risk classification. For the evaluation process, we relied on the CUAD (Contract Understanding Atticus Dataset) almost entirely as the semantic retrieval benchmark and avenue for clause labeling. The following sections provide an overview of the experimental setup, metrics used, results, and the notable insights that emerged from testing the system.

A. Experimental Setup

The system was implemented in Python 3.10 utilizing free-source frameworks, including Sentence Transformers, FAISS and Streamlit. The reasoning engine of the implementation was the Zephyr-7B Large Language Model (located on the Hugging Face Inference API), while Docling managed text extraction from PDF and DOCX files. All experiments were conducted on a system with an Intel i7 processor, 16 GB RAM and NVIDIA RTX GPU (6 GB VRAM) to handle embedding computations and model inferences. The evaluation environment replicated real-world use cases where users uploaded different contracts with clauses covering confidentiality, liability, termination, and indemnification.

B. Assessment Metrics

To evaluate the effectiveness of the system, the following four metrics were implemented:
Retrieval Accuracy (%): This indicates the number of clauses from the CUAD that matched the actual risk category in the top retrieved examples, reported as a percentage.
Similarity Score: This is the average cosine similarity between contract clauses and the CUAD clauses retrieved in the answer.
Risk Classification Accuracy: This metric indicates the accuracy of the LLM predicted clause categories relative to the ground-truth CUAD labels.
User Satisfaction Score: This metric is based on subjective evaluation gained from users (law students and legal professionals) on the usability, interpretability, and clarity of recommendations created in the system.

C. Quantitative Results

The quantitative analysis demonstrated strong performance across multiple dimensions.

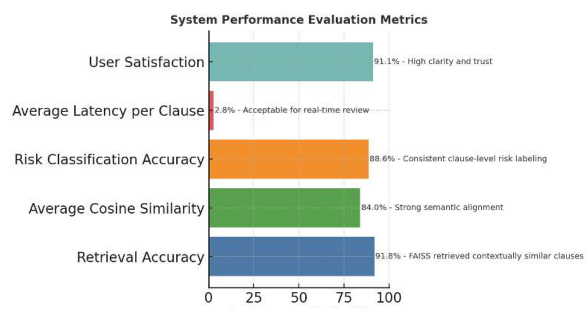

Figure 2. Quantitative results

These results validate that combining semantic retrieval with generative LLM reasoning significantly improves accuracy and reliability over standalone text classification approaches.

D. Qualitative Analysis

While quantitative metrics were informative, qualitative analysis demonstrated the system's ability to recognize complex and ambiguous clauses. The RAG framework allowed the LLM to place each clause in context with real-world examples, verifying legal defensibility rather than purely generative responses. For example:

- For Confidentiality Clauses, the system identified a missing temporal restriction.
- For Liability Clauses, it presented the phrase, "aggregate liability shall not exceed amount of contract."
- For Termination Clauses, it identified vague conditions, introduced fairness considerations, and suggested time frames, and notice periods.

This interpretability made the system distinctive as a differentiator to potentially evaluative groups from black box models.

E. Comparative Performance

To further assess performance, the proposed RAG-based LLM system was compared with two baselines:

- Keyword-Based Search System: Used TF-IDF and regex matching for clause identification.
- Standalone LLM (no retrieval): Used Zephyr-7B directly on contract text without FAISS support.

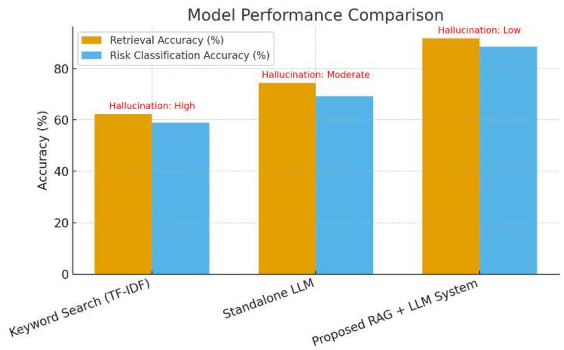

Figure 3. Comparison

The integration of retrieval grounding in the RAG pipeline clearly improved factual correctness and reduced hallucination errors, making it more dependable for legal document review.

F. Discussion

The experiment produces evidence that the Legal Contract Analyzer is extremely effective for actual use cases, weighing accuracy, interpretability, and processing time. The combined RAG structure ensures each response is contextual and legally sound. The results indicate LLMs build significantly on reliability when grounded in the domain-specific embeddings. Analyzing, models will still face challenges, primarily its reliance on the quality of the initial dataset and processing time for long documents but the Legal Contract Analyzer already has clear value in diminishing review workload, assuring improved compliance, and standardizing the evalution of legal contracts across organizations.

VII. Conclusion

The Legal Contract Analyzer we're proposing introduces a structured and intelligent approach to reviewing contracts quickly and accurately. It uses LLMs and retrieval-augmented techniques for real-time analysis, risk categorization, and context-driven recommendations, so solicitors can minimize human mistakes while saving time. The intelligent system increases efficiency and accuracy in the evaluation of legal considerations, while also providing valuable insights to the organization, so that they are able to maintain compliance and mitigate potential legal and financial liabilities. Overall, the system not only advances the development of each individual legal practitioner but also increases the overall efficiency of the organization as a whole.

Future Scope

The Legal Contract Analyzer can be developed into a full-featured, collaborative, and predictive legal intelligence system. Future products may harness peer-to-peer expert review networks for collaborative review of risk profiles, predictive analytics of future disputes, and AI-supported clause creation for better contract authoring. Continuous fine-tuning of the models using real-world contracts would increase adaptability, accuracy, and contextual understanding of contracts, allowing the system to become a fully autonomous, continuous, and scalable solution for intelligent legal contract management globally.

References

[1] Tomoki Ito and Shun Nakagawa. 2024. Tender Document Analyzer with the Combination of Supervised Learning and LLM-based Improver. In Companion Proceedings of the ACM Web Conference 2024 (WWW '24). Association for Computing Machinery, New York, NY, USA, 995–998. https://doi.org/10.1145/3589335.3651233

[2] Wutong Zhang, Hefeng Zhou, Qiang Zhou, Yunshen Li, Yuxin Liu, Jiong Lou, Chentao Wu, and Jie Li. 2025. Towards Comprehensive Legal Document Analysis: A Multi-Round RAG Approach. In Proceedings of the 2025.
[3] International Conference on Multimedia Retrieval (ICMR '25). Association for Computing Machinery, New York, NY, USA, 1840–1848. https://doi.org/10.1145/3731715.3733451
[4] V. S, M. Vimaladevi, R. Thangamani, V. N, C. M and S. V. Easwaramoorthy, "Revolutionizing Legal Access: An AI-Driven RAG Chatbot for Real-Time Judicial Insights," 2025 3rd International Conference on Artificial Intelligence and Machine Learning Applications Theme: Healthcare and Internet of Things (AIMLA), Namakkal, India, 2025, pp. 1-6,doi:10.1109/AIMLA63829.2025.11040439.
[5] Haitao Li, Yifan Chen, Hu YiRan, Qingyao Ai, Junjie Chen, Xiaoyu Yang, Jianhui Yang, Yueyue Wu, Zeyang Liu, and Yiqun Liu. 2025. LexRAG: Benchmarking Retrieval-Augmented Generation in Multi-Turn Legal Consultation Conversation. In Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '25). Association for Computing Machinery, New York, NY, USA, 3606–3615. https://doi.org/10.1145/3726302.3730340
[6] M. Hindi, L. Mohammed, O. Maaz and A. Alwarafy, "Enhancing the Precision and Interpretability of Retrieval-Augmented Generation (RAG) in Legal Technology: A Survey," in IEEE Access, vol. 13, pp. 46171-46189, 2025, doi: 10.1109/ACCESS.2025.3550145.
[7] Ingher, A. (2024). Large Language Model Agent as Insurance Law Assistant.
[8] Li, H., Chen, Y., YiRan, H., Ai, Q., Chen, J., Yang, X., ... & Liu, Y. (2025, July). LexRAG: Benchmarking Retrieval-Augmented Generation in Multi-Turn Legal Consultation Conversation. In Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 3606-3615).
[9] Pipitone, N., & Alami, G. H. (2024). Legalbench-rag: A benchmark for retrieval-augmented generation in the legal domain. arXiv preprint arXiv:2408.10343.
[10] Wiratunga, N., Abeyratne, R., Jayawardena, L., Martin, K., Massie, S., Nkisi-Orji, I., ... & Fleisch, B. (2024, June). CBR-RAG: case-based reasoning for retrieval augmented generation in LLMs for legal question answering. In International Conference on Case-Based Reasoning (pp. 445-460). Cham: Springer Nature Switzerland.