International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 06 | June - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

AI-Powered Contract Compliance Checker

Rahul R N, Monish V, Akshaya R S

Rahul R N – B.tech AI/ML – Sri Shakthi Institute of Engineering & Technology
Monish V – B.tech AI/ML – Sri Shakthi Institute of Engineering & Technology
Akshaya R S – B.tech AI/ML – Sri Shakthi Institute of Engineering & Technology

---

Abstract - This project presents an AI-powered system designed to automate contract compliance checking using advanced natural language processing and machine learning techniques. The primary objective is to ensure that legal agreements adhere to specified compliance requirements, reducing manual review time and minimizing the risk of oversight. The system employs a retrieval-augmented generation (RAG) framework, where contract data is converted into vector embeddings using ChromaDB and then queried for relevant context. A Streamlit-based user interface allows users to upload contracts and view compliance insights. Text extraction and clause identification are performed through a FastAPI backend integrated with a large language model via the Groq API. The system effectively identifies missing or non-compliant clauses by comparing input contracts against a preprocessed dataset. Results demonstrate the tool's ability to detect inconsistencies with high accuracy, offering significant potential for legal and corporate environments. This solution streamlines the compliance review process, increases efficiency, and provides real-time AI-driven suggestions to improve contractual documents.

***Key Words*:** Contract compliance, Natural Language Processing, ChromaDB, Retrieval-Augmented Generation, Legal AI, Streamlit.

1.INTRODUCTION

In today's fast-paced digital environment, the need for efficient and accurate contract management has become increasingly critical. Organizations regularly handle a vast number of legal documents that must comply with internal policies, regulatory standards, and industry-specific requirements. Manual review of contracts is not only time-consuming but also prone to human error, which can lead to significant legal and financial risks.

To address this challenge, the proposed project—**AI-Powered Contract Compliance Checker**—leverages advancements in Artificial Intelligence (AI), particularly in Natural Language Processing (NLP), to automate the process of verifying contract compliance. By integrating a Retrieval-Augmented Generation (RAG) approach with a user-friendly interface, this system assists legal professionals in identifying non-compliant or missing clauses in contractual documents.

The project uses ChromaDB for storing and retrieving document embeddings, while a FastAPI backend and Groq large language model API enable efficient clause detection and semantic understanding of contracts. A Streamlit-based frontend ensures an accessible and interactive user experience.

The system is trained on a curated dataset of compliant clauses, allowing it to highlight discrepancies between input contracts and established standards. This innovation significantly reduces the time and effort required for contract review while enhancing accuracy and compliance assurance.

2. SYSTEM DESIGN AND METHODOLOGY

2.1 System Architecture

The AI-Powered Contract Compliance Checker is built using a modular architecture consisting of data storage, backend processing, and a frontend interface. The system uses Retrieval-Augmented Generation (RAG), which combines document retrieval and generative language modeling to enhance understanding and compliance checking. Fig. 1 presents a high-level overview of the system architecture, which includes the ChromaDB storage layer, FastAPI backend, and Streamlit user interface.

2.2 Dataset Preparation

The dataset used in this project includes a collection of standard contract clauses. These have been preprocessed and stored in update_dataset.csv. The preprocessing was performed using a large language model (LLM) to ensure clause consistency, relevancy, and formatting. The processed data is then embedded into vector form using ChromaDB for similarity search and retrieval.

2.3 Embedding Generation and Storage

As shown in Sec. 2.2, the cleaned dataset is transformed into embeddings using contract_rag.py. These embeddings represent the semantic meaning of each clause and are stored in the chromdb_storage folder. This vector database enables efficient retrieval of similar clauses when a user uploads a new contract for analysis.

2.4 Similarity Search and Context Retrieval

The context_rag.py module performs the core functionality of similarity search. When a contract is uploaded, the system extracts each clause and compares it with existing embeddings using cosine similarity. This contextual matching helps identify missing or non-compliant clauses, which are then flagged for user review.

2.5 Backend Clause Analysis

The FastAPI backend (main.py) is responsible for extracting text from uploaded contracts, parsing key clauses, and interfacing with the Groq API. The Groq-powered LLM is used to generate human-like analysis and suggestions based on clause content and similarity results. **This API-driven** structure ensures that responses are both accurate and scalable.

2.6 Frontend and User Interaction

The frontend of the system is implemented using Streamlit (app.py), providing a responsive and user-friendly interface. Users can upload contract files, view real-time compliance reports, and receive AI-driven suggestions. The results include a similarity score, visual indicators, and detailed recommendations for each clause, helping users understand the rationale behind each compliance decision.

2.7 Security and Configuration

To ensure security and configuration management, sensitive information such as the Groq API key is stored in a .env file. This approach keeps credentials secure while enabling easy updates and deployment.

Table -1: Overview of System Components, Functions, and Corresponding Modules

Component	File/Module Name	Function
Dataset Storage	update_dataset.csv	Stores preprocessed compliant clauses
Vector Database	chromdb_storage	Stores semantic embeddings
Embedding Generator	contract_rag.py	Converts text to embeddings
Similarity Search	context_rag.py	Finds similar clauses
Backend API	main.py	Extracts and analyzes clauses
Frontend Interface	app.py	User interface and display
Environment Config	.env	Stores API keys securely

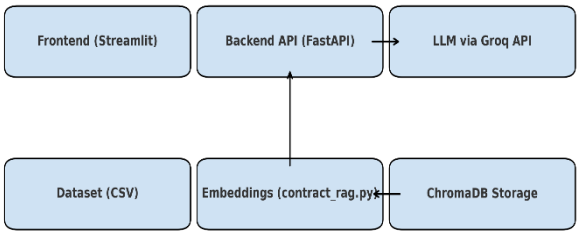

Fig -1: System Architecture

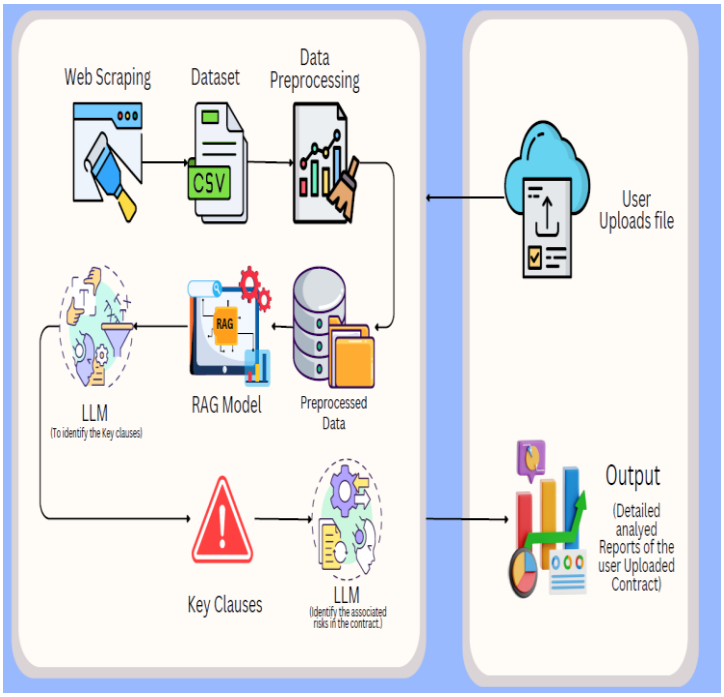

Fig – 2: Architecture Diagram

3. CONCLUSIONS

The AI-Powered Contract Compliance Checker automates legal contract compliance analysis using advanced AI techniques like Retrieval-Augmented Generation, ChromaDB vector storage, and the Groq API. Its combination of a Streamlit frontend and FastAPI backend offers an efficient, user-friendly experience with fast, accurate clause comparisons and actionable insights. This system reduces manual review time, improves compliance reliability, and is scalable for integration with enterprise tools. By enabling semantic search of contract language, it helps identify subtle risks and allows legal professionals to focus on strategic tasks. Future enhancements could expand multilingual support and collaboration features, highlighting AI's growing role in transforming legal technology and contract management. Overall, this project demonstrates how AI can streamline complex legal workflows, enhancing accuracy and operational efficiency across industries.

ACKNOWLEDGEMENT

The author would like to thank the developers and communities behind Streamlit, FastAPI, ChromaDB, and Groq for their open-source contributions that made this project possible. Special thanks to academic mentor and peers for their valuable feedback and encouragement throughout the development of this project.

REFERENCES

1. Johnson, J., Douze, M., Jégou, H.: Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data* (2019).
2. Ramírez, S.: FastAPI: Modern, fast (high-performance) web framework for building APIs with Python 3.6+. *FastAPI Documentation*, https://fastapi.tiangolo.com/ (2020)
3. Streamlit Inc.: Streamlit: Turn data scripts into shareable web apps in minutes. *Streamlit Documentation*, https://docs.streamlit.io/ (2020)
4. OpenAI: Embeddings API Guide. *OpenAI Platform Documentation*, https://platform.openai.com/docs/guides/embeddings (2023)
5. GroqCloud**:** Groq LLM API for contract analysis. *GroqCloud API Documentation*, (Access restricted; refer to your .env configuration)