International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 22 No. 2(2) (2026)

ijerst.editor@gmail.com

editor@ijerst.com

International Journal of Engineering Research and Science & Technology
ISSN 2319-5991 www.ijerst.com

Peer Reviewed, Referred & Indexed Journal

Research Paper

AI Legal Assistant Using RAG and LLMs

Mrs T. Aruna Jyothi
Assistant Professor
Department of Information technology
Matrusri engineering college

Mahesh Karva
Department of computer engineering
Matrusri engineering college
maheshkarva.finix@gmail.com

K. Madhumitha
Department of computer engineering
Katamadhu13@gmail.com

ABSTRACT:

Access to legal assistance remains a significant challenge for individuals who cannot afford professional lawyers or navigate the complexity of legal documentation independently. This paper presents an AI-powered legal assistant designed to help users draft, understand, and retrieve information from legal documents through a conversational interface. The system is developed as a full-stack web application built with Flask for the backend, LangChain for orchestration, ChromaDB as a vector database, and the Ollama large language model for inference.

The platform enables users to upload legal documents in PDF or DOCX format, after which the system processes and indexes their contents using a Retrieval-Augmented Generation pipeline. Document chunks are embedded and stored in ChromaDB, allowing the system to retrieve contextually relevant passages in response to natural language queries. The large language model then uses these retrieved passages as context to generate precise, grounded answers rather than relying on parametric knowledge alone.

Beyond question answering, the system provides several intelligent features including clause extraction, AI-driven contract and agreement drafting, advocate search by location and specialty, automated formal letter generation, and downloadable PDF document output. The platform also includes user authentication, a personal dashboard, and an admin management interface for retraining and dataset management.

By integrating large language model capabilities with a retrieval-augmented architecture and an interactive web interface, the system assists users in understanding legal documents, reducing dependence on costly legal consultations, and accessing accurate legal information in real time. The proposed platform aims to democratize legal assistance by making intelligent, document-grounded legal support accessible to a broad audience.

KEYWORDS:

Artificial Intelligence, Large Language Models, Retrieval-Augmented Generation, Legal Document Analysis, Conversational AI, Natural Language Processing, LangChain, ChromaDB, Con-tract Drafting, Legal Chatbot

1. INTRODUCTION

Legal services have historically been inaccessible to a large segment of the population due to high consultation costs, complex terminology, and the sheer volume of documentation involved in legal processes. Individuals seeking guidance on contracts, agreements, non-disclosure documents, or dispute letters often lack the resources to engage professional legal counsel. As a result, many are forced to rely on outdated templates or incomplete information, increasing the risk of procedural and substantive errors [1][4].

The emergence of large language models and generative AI has introduced new possibilities for automating knowledge-intensive tasks that previously required specialized human expertise. Recent research has demonstrated the ability of transformer-based models to comprehend and reason over legal text, perform document summarization, extract critical clauses, and even achieve competitive scores on professional legal examinations [3]. However, standalone generative models are prone to hallucination, particularly when producing specific legal facts or document-specific content that falls outside their training data [2].

Retrieval-Augmented Generation addresses this limitation by combining the generative capability of language models with the precision of information retrieval. In a RAG pipeline, relevant document segments are retrieved from a vector database and provided as context to the language model, ensuring that responses are grounded in the specific documents the user has uploaded rather than in potentially outdated parametric knowledge [1]. This architecture is particularly well suited to legal applications where accuracy, specificity, and document fidelity are paramount.

The proposed system implements this RAG-based approach within a complete web application for legal assistance. Users can upload legal documents, interact with an AI chatbot to query document contents, request clause extraction, and commission the drafting of standard legal agreements. The system also provides an advocate search feature and a formal letter drafting and email dispatch module, making it a comprehensive tool for legal self-service [5].

By integrating large language models with retrieval-augmented generation, LangChain orchestration, and ChromaDB vector storage within an accessible web interface, the system aims to bridge the gap between professional legal expertise and the broader public. The platform is designed to be context-aware, fast, and accurate, enabling users to obtain substantive legal assistance without incurring the costs of traditional consultation.

1.1. OBJECTIVES

- To develop an AI-powered legal assistant that can answer natural language queries about uploaded legal documents using a Retrieval-Augmented Generation pipeline.
- To design a document ingestion module that processes PDF and DOCX files, segments them into meaningful chunks, and stores them as vector embeddings in ChromaDB.
- To implement a clause extraction feature that enables users to identify and retrieve specific provisions from legal agreements and contracts.
- To provide an AI-driven document drafting capability that can generate standard legal documents such as non-disclosure agreements, service contracts, and memoranda of understanding.
- To integrate an advocate search module that connects users with legal professionals based on geographic location and area of specialty.
- To build a legal communication module that drafts and dispatches formal legal correspondence via email on the user's behalf.
- To develop a secure, user-friendly web platform using Flask, LangChain, and modern frontend technologies that provides personalized dashboards and session management.

1.2. PROPOSED SYSTEM

The proposed system is an AI-powered legal assistant that addresses the inefficiencies and accessibility barriers of traditional legal consultation. The system is designed to process user-uploaded legal documents and provide intelligent, document-grounded responses through a conversational interface, eliminating the need for users to manually search through lengthy legal texts.

When a user uploads a legal document, the system employs LangChain to load and split the document into smaller text segments. These segments are converted into high-dimensional vector embeddings using the Ollama language model and stored in a ChromaDB vector database. This indexing process enables fast semantic search over the document contents at query time.

Upon receiving a user query, the system retrieves the most semantically relevant document chunks from ChromaDB and passes them to the language model as context. The model then generates a response that is grounded exclusively in the retrieved content, minimizing the risk of hallucinated or factually incorrect output. This RAG-based interaction model ensures that responses remain anchored to the specific documents provided by the user.

In addition to document querying, the platform supports clause extraction, contract generation, advocate discovery, and formal legal letter drafting. These features are exposed through a modular Flask backend with dedicated API endpoints for each functional capability. All user sessions are protected through secure authentication, and each user interacts only with their own uploaded documents.

The system is built on a stack comprising Flask, LangChain, ChromaDB, Ollama LLM, SQLite, ReportLab for PDF generation, and SMTP for email dispatch. The frontend provides an interactive chat interface and personal dashboard, making the platform accessible to users regardless of their legal expertise.

2. LITERATURE SURVEY

The application of artificial intelligence to legal document analysis and legal question answering has gained substantial research attention in recent years [3][4]. These efforts span multiple paradigms, from rule-based expert systems and statistical methods to modern deep learning architectures and large language models, each contributing a distinct set of capabilities and limitations.

Early computational approaches to legal text processing relied on keyword-based retrieval and hand-crafted rule systems. These systems could identify relevant statutes or case summaries by matching terms in user queries against indexed legal databases, but they lacked the semantic understanding necessary to interpret ambiguous legal language or infer meaning from context [6]. The rigid structure of rule-based methods made them brittle when applied to the diverse and nuanced language characteristic of legal documents.

With the advancement of machine learning, researchers began applying probabilistic and statistical methods to legal text classification and information retrieval [7]. Support vector machines and logistic regression models were used to categorize legal documents by topic and jurisdiction, while TF-IDF-based retrieval approaches improved the relevance of search results over pure keyword matching [8]. However, these methods still treated text as a bag of features without capturing the relational and contextual structure of legal language.

The introduction of transformer-based language models, beginning with BERT and extending to GPT-class models, significantly advanced the state of legal natural language processing [2]. Domain-specific models such as LegalBERT and ContractBERT were pre-trained on large legal corpora to capture the specialized vocabulary and syntactic patterns of legal text, enabling more accurate classification, clause detection, and semantic similarity tasks [4]. These models demonstrated that pre-trained representations substantially improve performance on downstream legal tasks compared to general-domain models.

More recently, the emergence of large instruction-tuned language models has enabled a shift toward generative legal AI. Studies have shown that GPT-4 class models can achieve competitive performance on standardized legal examinations such as the Uniform Bar Exam, suggesting

that these models internalize sufficient legal reasoning capability for practical application [3]. Parallel work on retrieval-augmented generation demonstrated that grounding LLM responses in retrieved documents significantly reduces hallucination in knowledge-intensive domains, making RAG an attractive architecture for legal applications where factual accuracy is critical [1].
The following table summarizes key contributions from the literature related to AI-driven legal document processing, question answering, and contract analysis.

Table 1: Summary of Related Work in AI-Powered Legal Assistance

Author & Year	Method Used	Dataset / Domain	Key Contribution
Lewis et al. (2020) [1]	Retrieval-Augmented Generation	Open-domain QA benchmarks	Introduced RAG framework combining dense retrieval with generative models for knowledge-intensive tasks.
Vaswani et al. (2017) [2]	Transformer Architecture	Machine translation corpora	Proposed the attention-based transformer model that under-lies all modern large language models.
Katz et al. (2023) [3]	GPT-4 Evaluation	Uniform Bar Exam	Showed that large language models can pass professional le-gal examinations, establishing their legal reasoning ability.
Chalkidis et al. (2021) [4]	ContractBERT (Domain Pre-training)	Commercial contracts	Pre-trained BERT on contract corpora to improve clause clas-sification and contract under-standing.
Zhong et al. (2020) [6]	BERT Fine-tuning	Chinese legal judgments	Applied transformer models to legal judgment prediction, demonstrating strong perfor-mance on charge classification.
Lippi et al. (2019) [7]	Machine Learning Classifier s	Terms of service documents	Developed classifiers to de-tect potentially unfair clauses in online consumer contracts (CLAUDETTE).
Manor & Li (2019) [8]	Text Matching / NLP	Contract clause datasets	Proposed a contract clause re-trieval approach using semantic similarity and TF-IDF for legal search.
Hendrycks et al. (2021) [9]	BERT + GPT Evaluation	CUAD legal benchmark	Released the Contract Under-standing Atticus Dataset and benchmarked multiple models on clause extraction.

Lewis et al. (2020) [1] introduced the Retrieval-Augmented Generation framework, which com-bines a dense passage retrieval module with a generative language model. Their work demon-strated that grounding generation in retrieved documents substantially improves factual accu-racy on open-domain question-answering benchmarks. This architecture forms the core of the

proposed system's response generation pipeline.

Vaswani et al. (2017) [2] proposed the transformer architecture based on self-attention mechanisms, replacing recurrent models and establishing the foundation for all subsequent large language models including BERT, GPT, and Llama. The attention mechanism enables models to capture long-range dependencies in text, a property essential for understanding complex legal clauses and cross-referencing provisions within contracts.

Katz et al. (2023) [3] evaluated GPT-4 on the Uniform Bar Exam, a standardized examination for legal professionals. Their results showed that the model achieves passing scores, indicating that large language models encode sufficient legal reasoning capability to be useful in practical legal assistance scenarios.

Chalkidis et al. (2021) [4] presented ContractBERT, a BERT model pre-trained on a large corpus of commercial contracts. The model outperformed general-domain BERT on tasks including clause classification and contractual obligation detection, demonstrating the value of domain-specific pre-training for legal NLP.

Zhong et al. (2020) [6] applied transformer models to legal judgment prediction within the Chinese judicial system. Their system predicted charges, applicable articles, and prison terms from fact descriptions with high accuracy, illustrating the breadth of legal tasks amenable to deep learning approaches.

Lippi et al. (2019) [7] developed CLAUDETTE, a system for detecting potentially unfair clauses in online terms-of-service documents using machine learning classifiers. Their work highlighted the practical value of AI tools for consumer legal protection and clause-level document analysis.

Manor and Li (2019) [8] studied contract clause retrieval using semantic text matching techniques. Their work showed that similarity-based retrieval substantially outperforms keyword search for identifying relevant contract provisions, underpinning the use of vector embeddings in retrieval-augmented systems.

Hendrycks et al. (2021) [9] released the Contract Understanding Atticus Dataset (CUAD), containing over 500 contracts annotated for 41 legal clause categories. The benchmark enabled systematic evaluation of models on clause extraction, demonstrating that while general models perform reasonably, specialized fine-tuned models achieve superior results on targeted extraction tasks.

3. SYSTEM ARCHITECTURE

The proposed system follows a modular, layered architecture that integrates large language model inference, vector-based retrieval, and web application services to deliver intelligent legal assistance. The system comprises three principal layers: the frontend presentation layer, the Flask-based backend service layer, and the AI and data layer incorporating the RAG pipeline and persistent storage.

The frontend layer provides a browser-based user interface built with HTML, CSS, and JavaScript. Users can register, log in, and access a personal dashboard through which they upload legal docu-ments, interact with the AI chatbot, request document drafts, search for advocates, and manage their session. The interface exposes dedicated views for each functional module, presenting results in a readable format and enabling document downloads.

The backend layer is implemented using the Flask web framework, which exposes a set of REST-ful API endpoints corresponding to each platform feature. The authentication module handles user registration and login using hashed credentials and session tokens. File upload endpoints accept PDF and DOCX files, store them on the server, and trigger the document ingestion

pipeline. Separate endpoints handle chat queries, clause extraction, document generation, advocate search, and legal mail drafting and dispatch via SMTP.

The AI and data layer forms the core of the system's intelligence. When a document is uploaded, LangChain loads and splits it into overlapping text chunks to preserve local context. The Ollama large language model generates vector embeddings for each chunk, which are stored in a ChromaDB vector database indexed by user session. At query time, the user's natural language input is embedded and a semantic similarity search retrieves the most relevant chunks from ChromaDB. These retrieved chunks are assembled into a structured prompt and passed to the Ollama LLM, which generates a response grounded in the retrieved context. For document drafting requests, the LLM generates complete agreement templates based on user-specified parameters. SQLite stores user account data, session information, and interaction history, while ReportLab converts AI-generated documents into downloadable PDF files.

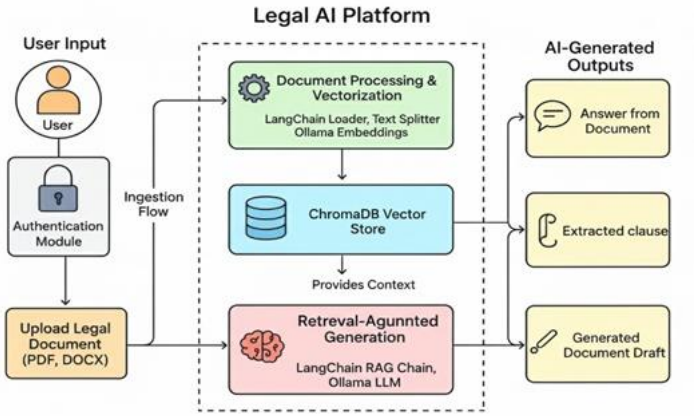

Figure 1: Technical Architecture of the AI Legal Assistant System — Legal AI Platform Data Flow

3.1. METHODOLOGY

The methodology of the proposed system is organized into distinct processing stages that collectively transform a user's raw legal document into an interactive, queryable knowledge source, and enable downstream legal assistance tasks.

The first stage is document ingestion. Upon upload, LangChain's document loaders parse the PDF or DOCX file and extract its full text content. The text is then split into overlapping chunks using a recursive character splitter, with chunk size and overlap parameters tuned to balance retrieval granularity and contextual coherence. This chunking strategy ensures that clause boundaries and cross-references are preserved within individual chunks wherever possible.

The second stage is embedding and indexing. Each text chunk is passed through the Ollama embedding model to produce a dense vector representation that captures the semantic content of that passage. These embeddings are stored in ChromaDB together with metadata including the source document identifier and chunk position, enabling per-user retrieval isolation and efficient nearest-neighbour search.

The third stage is query processing. When a user submits a question, the query text is embedded using the same model to produce a query vector. ChromaDB performs a cosine similarity search to identify the top-k most semantically relevant chunks from the user's indexed document. These chunks are concatenated and inserted into a structured prompt template alongside the original query.

The fourth stage is response generation. The assembled prompt is submitted to the Ollama large language model, which generates a natural language response based exclusively on the provided context. The model is instructed via the system prompt to refrain from introducing information beyond what is contained in the retrieved passages, ensuring response fidelity to the source document.

The fifth stage covers specialized tasks beyond question answering. For clause extraction, the user specifies a clause type and the retrieval pipeline fetches the most relevant passage, which the model then reformats for clarity. For document drafting, the user provides key parameters such as party names, jurisdiction, and contract type, and the model generates a complete document template. For legal letter drafting and dispatch, the user specifies the recipient and intent, and the model produces formal correspondence that is sent via the SMTP email module.

This methodology delivers a complete, end-to-end legal assistance workflow that requires no specialist knowledge from the user and produces outputs grounded in the specific documents and parameters provided.

3.2. RESULTS

- The system provides users with a secure, interactive platform for obtaining accurate, document-specific answers to legal queries without requiring prior legal knowledge. The RAG pipeline ensures that responses are grounded in the content of uploaded documents, substantially reducing the risk of misleading or fabricated information.
- The clause extraction module enables users to identify and retrieve specific provisions from contracts and agreements in seconds, a task that would otherwise require manual review of potentially lengthy documents. The document drafting capability generates standard legal templates including NDAs, service agreements, and memoranda of understanding based on user-specified parameters.
- The advocate search feature connects users with legal professionals who match their geographic location and area of legal need, bridging the gap between AI-assisted self-service and situations that require qualified human counsel. The legal communication module automates the drafting and dispatch of formal legal correspondence, reducing administrative burden for both individuals and small organizations.
- The overall outcome is an integrated AI-powered legal assistance platform that makes substantive legal support accessible, fast, and affordable. By combining large language model intelligence with retrieval-augmented grounding and a modular feature set, the system addresses the principal pain points of the existing legal consultation landscape: high cost, slow turnaround, and limited accessibility outside business hours.

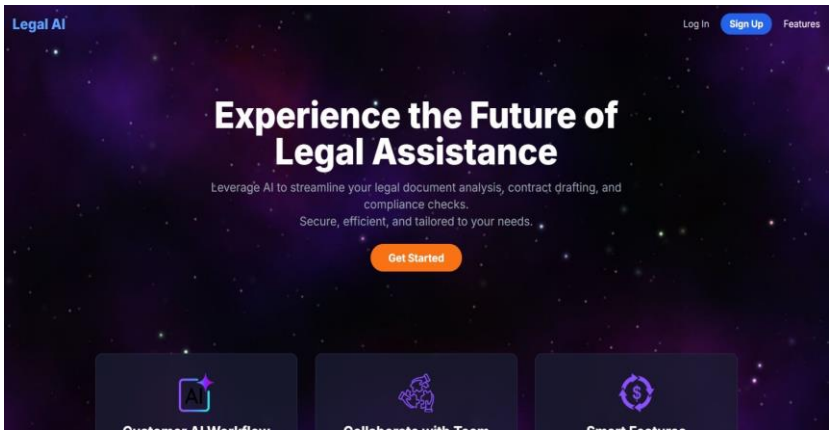

Figure 2: User Interface

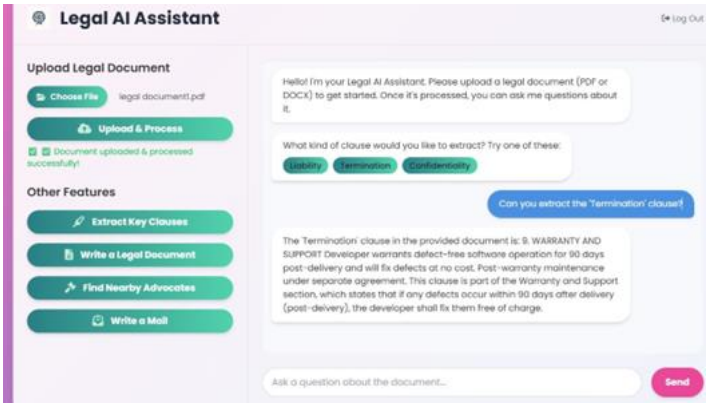

Figure 3: System Modules: Document Upload, AI Interaction and Document Generation

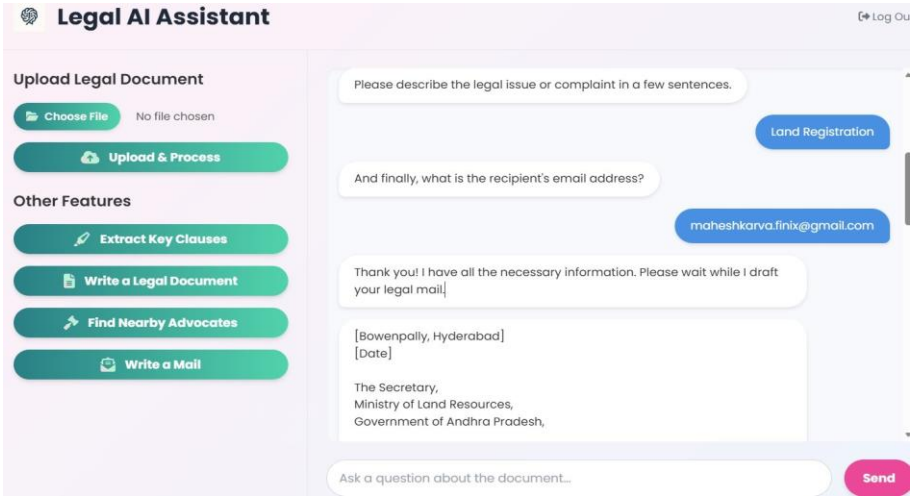

Figure 4: Advocate Search, Legal Communication

4. FUTURE SCOPE

- The platform can be extended by integrating access to structured legal databases such as case law repositories, statute collections, and regulatory archives. Connecting the system to live legal information sources would allow it to supplement document-specific responses with authoritative citations to relevant precedent and legislation, substantially increasing the depth and reliability of legal guidance provided.
- Future work could incorporate multi-document reasoning, enabling users to upload multiple related contracts or documents and ask cross-document queries such as identifying conflicting clauses or tracing the evolution of contractual terms across successive agreement versions. This capability would be valuable for legal professionals managing complex transaction doc-

umentation.

- The system can be enhanced with jurisdiction-aware reasoning by fine-tuning or prompting the language model with jurisdiction-specific legal corpora. Users would specify their country or state, and the system would tailor its responses, document templates, and clause recommendations to the applicable legal framework.
- Additional modalities such as speech input and output could be integrated to make the platform accessible to users with low digital literacy or disabilities. A mobile application version with voice-driven legal querying would extend the system's reach to underserved populations in regions with limited access to legal professionals.

5. CONCLUSION

The AI Legal Assistant demonstrates the practical potential of combining large language model inference with retrieval-augmented generation to deliver accurate, document-grounded legal assistance at scale. By processing user-uploaded legal documents through a LangChain and ChromaDB pipeline and generating responses via the Ollama LLM, the system provides contextually precise answers to legal queries while avoiding the hallucination risks associated with ungrounded generative models.

The platform consolidates multiple legal assistance functions including conversational document querying, clause extraction, contract drafting, advocate discovery, and formal correspondence generation within a single secure web application. These capabilities collectively reduce the time, cost, and expertise barrier associated with accessing legal support, democratizing assistance for individuals and small organizations that cannot afford traditional legal consultation.

By implementing the system using a production-grade technology stack of Flask, LangChain, ChromaDB, Ollama, SQLite, ReportLab, and SMTP, the platform achieves both functional completeness and operational reliability. The proposed system serves as a foundation for next-generation AI-powered legal tools that prioritize accuracy, privacy, and accessibility, and offers a clear pathway for extension toward jurisdiction-aware reasoning, multi-document analysis, and integration with live legal databases.

REFERENCES

[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-T. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 2020.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," in *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, 2017.

[3] D. M. Katz, M. J. Bommarito II, S. Gao, and P. Arredondo, "GPT-4 Passes the Bar Exam," *SSRN Electronic Journal*, 2023. https://doi.org/10.2139/ssrn.4389233

[4] I. Chalkidis, M. Fergadiotis, S. Kotitsas, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, "An Empirical Study on Large Pre-Trained Language Models for Contract Understanding," in *Proceedings of EMNLP 2021*, 2021.

[5] LangChain AI, "LangChain Documentation," Available: https://python.langchain.com/docs/get_started/introduction. [Accessed: 2025].

[6] H. Zhong, Z. Guo, C. Tu, C. Xiao, Z. Liu, and M. Sun, "Legal Judgment Prediction via Topological Learning," in *Proceedings of EMNLP 2018*, 2018.

[7] M. Lippi, P. Pal-ka, G. Contissa, F. Lagioia, H.-W. Micklitz, G. Sartor, and P. Torroni, "CLAUDETTE: An Automated Detector of Potentially Unfair Clauses in

Online Terms of Service," *Artificial Intelligence and Law*, vol. 27, no. 2, pp. 117–139, 2019.

[8] L. Manor and J. Li, "Plain English Summarization of Contracts," in *Proceedings of the Natural Legal Language Processing Workshop at ACL 2019*, 2019.

[9] D. Hendrycks, C. Burns, A. Chen, and S. Ball, "CUAD: An Expert-Annotated NLP Dataset for Legal Contract Review," in *NeurIPS Datasets and Benchmarks Track*, 2021. https://arxiv.org/abs/2103.06268

[10] N. Aletras, D. Tsarapatsanis, D. Preo¸tiuc-Pietro, and V. Lampos, "Predicting Judicial Deci-sions of the European Court of Human Rights: A Natural Language Processing Perspective," *PeerJ Computer Science*, vol. 2, e93, 2016.

[11] I. Chalkidis, A. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, "LEGAL-BERT: The Muppets Straight out of Law School," in *Findings of EMNLP 2020*, 2020.

[12] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, "Large Language Models are Zero-Shot Reasoners," in *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*, 2022.