IN PARTNERSHIP WITH PLYMOUTH UNIVERSITY

Name: Weerasinghe Dissanayakalge Methsara Nisaga Dissanayaka
Student Reference Number: 10899302

Module Code: PUSL 3190	Module Name: Computing Individual Project
Coursework Title: AI for Legal Document Analysis	
Deadline Date: 05/05/2024	Member of staff responsible for coursework: Dr. Mohomed Shafraz
Program: BSc. (Hons) in Data Science	

Please note that University Academic Regulations are available under Rules and Regulations on theUniversity website www.plymouth.ac.uk/studenthandbook.

Group work: please list all names of all participants formally associated with this work and state whether the work was undertaken alone or as part of a team. Please note you may be required to identify individual responsibility for component parts.

We confirm that we have read and understood the Plymouth University regulations relating toAssessment Offences and that we are aware of the possible penalties for any breach of theseregulations. We confirm that this is the independent work of the group.

Signed on behalf of the group:

Individual assignment: ***I confirm that I have read and understood the Plymouth University regulations relating to Assessment Offences and that I am aware of the possible penalties for any breach of these regulations. I confirm that this is my own independent work.***

Signed: *Methsara Nisaga Dissanayaka*

Use of translation software: failure to declare that translation software or a similar writing aid has been used will be treated as an assessment offence.

I *have used/not used translation software.

If used, please state name of software……

Acknowledgement

Acknowledgement

First and foremost, I am deeply grateful to my supervisor,
Dr. Mohamed Shafraz,
whose invaluable insights, constructive feedback, and unwavering encouragement have been instrumental in the successful completion of this project. His dedication and expertise have played a pivotal role in shaping the final outcome.

I would also like to express my sincere appreciation to the lecturers whose teachings and mentorship provided the foundational knowledge and skills necessary for this work. Their guidance has been essential in helping me navigate the complexities encountered throughout this project.

To my family and friends, I extend my heartfelt thanks for your unwavering support and encouragement. Your belief in me and your patience have been a constant source of strength and motivation, enabling me to overcome challenges along the way. I am truly grateful for your kindness and love.

Additionally, I wish to thank the members of Zaara Labs for their insightful feedback and caring throughout the project. Their understanding and encouragement have been invaluable, especially during challenging times, and have helped keep the team motivated.

In conclusion, I am profoundly grateful to everyone who has contributed to the success of the **Legal.AI** project. Thank you all for being an integral part of this achievement.

~~~~~~~
~~~~~~~

Abstract

The legal sector is often overlooked in the broader AI revolution due to its critical nature and the sheer complexity of its work. This complexity largely stems from the structure of legal documents themselves, particularly case law, which often have a complicated structure, densely written, and filled with domain-specific jargon. These features make legal texts challenging for traditional AI systems to interpret and process effectively, hindering the broader adoption of automation within legal workflows. This project seeks to address these challenges by developing an AI/NLP-powered pipeline that processes, extracts, and enhances the content of UK legal case files, focusing on improving the accessibility and usability of case information through semantic enrichment and structured metadata. The system integrates a hybrid rule-based and transformer-based natural language processing (NLP) approach, using LegalBERT and spaCy to extract metadata and legal references, and to segment legal text into semantically meaningful sentences. Legal references such as statutes and Acts are identified and linked contextually to support enhanced legal research and document navigation. The cleaned and structured output is stored using PostgreSQL and Pinecone for vector embeddings, enabling semantic chunk-level retrieval through LLM-powered interactions. To bridge the gap between traditional case reading and intelligent digital analysis, the system features a frontend interface that visualizes case content with interactive highlights, legal reference overlays, and AI-supported follow-up queries (GPT powered). In addition, sentence-level embeddings were employed using the LLaMA embedding model, facilitating in-document semantic search. Moreover, this system is also capable of summarizing an entire document, simplifying complex jargon and legal text classification. This work contributes to the growing field of legal AI by demonstrating a fully operational end-to-end pipeline capable of converting raw legal case files into structured, searchable, and semantically annotated resources. The results suggest significant improvements in usability, legal information access, and integration with advanced retrieval methods. Future work could expand upon this framework by integrating real-time legal updates, court-specific heuristics, and multilingual legal corpora.

Keywords: Natural Language Processing (NLP), Vector Embeddings, Semantic Search, Pinecone, Legal Document Analysis, REGEX, Metadata Extraction, Legal Reference Extraction, PostgreSQL, UK Law (Cybercrime), Transformer Models, GPT chatbot, Next.js Frontend

UNIVERSITY OF PLYMOUTH

PUSL3190 - Computing Individual Project

Final Report

AI for Legal Document Analysis

Supervisor: Dr. Mohamed Shafraz

Name: Weerasinghe Dissanayakalage Methsara Nisaga Dissanayaka
Plymouth Index Number: 10899302
Degree Program: BSc. (Hons) in Data Science

Table of Contents

1. Introduction

"The Lawyer of the future will be a hybrid: part legal expert, part technologist, harnessing artificial intelligence to deliver better and more efficient legal services" – Richard Susskind, a leading authority on the intersection of law and technology. The legal system is a fundamental pillar of society, establishing frameworks for rights, obligations, and justice. At its core are legal documents: such as acts, statutes, case law, contracts, and regulations, meticulously crafted by responsible professionals. These documents consist of extremely precise language and unique terminologies adhering to any situation that could occur. These texts require careful interpretation by legal professionals to ensure accurate interpretation and application. However, traditional methods of analyzing these documents, which rely on manual review, are increasingly inadequate in today's digital age. The growing volume and complexity of legal materials, combined with continuously amending regulations, has created a clear need for advanced technologies that can efficiently process large-scale text. This highlights the urgent demand for innovative solutions to improve accuracy, reduce inefficiencies, and support legal practitioners in addressing modern challenges.

While various systems have been developed to support the legal sector and its practitioners, many existing solutions remain incomplete, as they fail to integrate the full range of advanced techniques in one place that could offer significant improvements. Legal practitioners often struggle to locate the most relevant sections, track amendments, or fully understand how a provision has been interpreted across different cases with present systems. Keyword-based search methods frequently miss key information due to linguistic variations, and context-blind results often overwhelm users with irrelevant retrievals. As a result, legal professionals report spending more time searching for information than analyzing it. This inefficiency not only slows down legal workflows but can have tangible consequences in terms of missed arguments, overlooked clauses, or costly delays.

Artificial Intelligence (AI) and Natural Language Processing (NLP) offer strong solutions to these issues. Transformer-based language models have shown excellent performance in comprehension, summarization, and information retrieval from dense text data in recent years. In legal texts, these models can read through long documents, extract pertinent data, identify legal citations at sentence level, and perform semantic search by meaning and not exact words using vector embedding. They can help bring out big legal provisions, make definitions clearer, and even generate plain-English versions of difficult terms. They don't replace legal experts but instead they augment them, reducing time spent on pointless reading and allowing practitioners to focus on strategy and choice-making.

This project builds a legal document analysis system that integrates these state-of-the-art AI features into an operational pipeline. Using this system, a user is able to upload

a case text file in various formats such as (PDF, RTF, DOCX etc). and the system will perform multi-level processing to extract metadata, identify act and statute references, annotate legally relevant sentences, and store cleaned, structured versions of each document in a vector database for semantic search / retrieval. Sentence labeling and named-entity recognition are accomplished with transformer-based models like LegalBERT, and vector embeddings facilitate dense semantic search and matching. Results are stored in a backend system with a PostgreSQL database and Pinecone vector index, allowing for both precise document-level annotation and large-scale querying efficiently. The user interface allows natural language interaction with uploaded documents, emphasizing pertinent highlights, summaries, and legal citations in context.

Even though the system is generally applicable throughout the legal sphere, it has specific impact where documentation contains unfamiliar terminologies and technical jargon. For example, areas such as cybercrime or data protection law involve computer science, networking, and cryptography jargon – vocabulary used may not be familiar to most general lawyers. In such cases, AI-assisted analysis is well positioned to replace them, to clarify confusion between complex terms, and to identify relevant precedents with seamless transitions etc. That is why the system was built around cases that were related to the Cybercrime case mostly prosecuted under the Computer Misuse Act 1990 of UK, even though some things varied towards the end of.

Ultimately, this project is able to demonstrate how a thoughtfully engineered AI pipeline can reduce the friction in legal research and analysis. By making legal texts easier to search, interpret, and explain, it supports faster decision-making and broader accessibility. As legal systems become increasingly intertwined with digital technology, tools like this can help ensure that legal knowledge remains usable, navigable, and available to those who need it most.

2. Background

Historically, the legal profession has depended on large, complicated structured case files that are kept in a variety of formats. These documents, which can be sometimes hundreds of pages long, are mostly intended to be referred to / read by humans rather than computers. In order to find critical information, precedents, or statutory references, legal research procedures have therefore always involved manual navigation to surf through substantial text volumes. Digital case collections have made legal material more accessible to the public (although that itself often falls short in some aspects), but they haven't significantly altered how time-consuming it is to parse legal text and analysis of it.

The fact that these digitalized media files still aren't organized enough to be structured data in the existing system, is one of its main drawbacks. For the most part important details are buried in free-form writing with no consistent structure or labelling. This absence prevents efficient search, filtering, or automated summarization, creating bottlenecks in both legal practice and academic research. Initially this project aimed to create a pipeline that could reliably extract and organize metadata from case files. As development progressed, it became clear that surface level extraction should only be a part of the solution even though that alone is a big feat, the advancements in technology should allow this sector (legal sector) to accelerate just as much as it has with the others. So, it was observed that real value lies in the ability to interact with case content at a deeper level. his led to the development of a robust sentence-level NLP pipeline capable of parsing and extracting legal text with contextual references to legislation, text summarization, complex legal jargon simplification, enabling semantic search via embeddings and intelligent interaction via large language models.

The motivation for this work was grounded in practical needs: legal researchers, scholars, and even normal people often require tools that go beyond keyword search. Some of them have just a short amount of time to refer to a document so they need a quick summary of the text. They need to ask specific, nuanced questions - like how a certain statute was interpreted in a particular case, or how often a judge has ruled a certain way. By enriching raw legal documents with structured annotations, semantic chunks, vector embeddings, recursive summarization, legal text classification and integration of GPT models, this project sets the groundwork for many capabilities including the ones above.

3. Objectives and Deliverables

3.1 Project Objectives

1. Establish a Pipeline for Legal Document Processing. (**Creating**)
   - Develop a flexible and robust pipeline designed to accommodate various legal document formats (Ex- PDF, DOCX, RTF etc.). This framework will facilitate text extraction, metadata processing, and document normalization, effectively preparing legal texts for further analysis.

2. Integrate enhanced NLP methods for Advanced Summarization and Simplification of Legal Text. (**Integrating**)
   - Leverage state-of-the-art transformer-based models, such as Legal-BERT, fine-tuned on legal corpora, to distill key information, generate concise summaries, and simplify complex legal language. The aim is to reduce the length of legal texts by up to 70% while preserving the essential legal context.
   - Implement a dynamic glossary with contextual linking to the processed document for quick reference of complex legal terms, domain specific terms.

3. Enable Contextual Information Retrieval via Semantic Search. (**Applying**)
   - Implement semantic search capabilities using vector embeddings to enhance document navigation, allowing users to retrieve relevant content based on the underlying meaning of queries, not just exact keyword matches, improving search accuracy and efficiency.
   - Achieve a precision and recall rate of at least 85% for user queries.

4. Develop Knowledge Graphs to Visualize Legal Document Relationships. (if feasible)
   - Construct a knowledge graph model that maps out relationships and dependencies within the text. This is crucial for identifying primary entities, connections and providing an interactive layer of insight into legal networks.
   - Aim to label connections and relationships for at least 80% of entities in the dataset.

5. Customize the AI Tool for Computer Crime case law using domain-specific Model Training and Evaluate Tool Performance. (**Analyzing & Evaluating**)
   - Fine-tune the AI system to address the specific requirements and nuances of a particular domain (cybercrime cases) and carry out performance evaluations on

a dataset specific to the selected domain. Thus, creating a precise and functional AI solution for a targeted legal area.

6. Design an Intuitive & Interactive User interface. (**Developing**)
   - Create an easy-to-navigate, user-friendly interface that allows legal professionals to interact with complex legal documents effortlessly. The interface will support smooth interaction with legal data and facilitate tasks such as document review and query formulation.

3.2 Project Deliverables

1. **Legal Document Preprocessing Pipeline:** A robust preprocessing module was developed to clean raw legal documents before analysis. It handles complex formatting issues common in legal PDFs, RTFs, etc. such as broken words, inconsistent punctuation, irregular spacing, and malformed sections. This ensures a clean, standardized text structure that downstream NLP models can reliably process, forming the foundation for all other stages of the pipeline.

2. **Metadata Extraction System:** The system automatically extracts structured metadata from each case document, including fields such as case name, date, judge, court, and cited legislation. This information is stored in JSON format and indexed in a PostgreSQL database for structured retrieval and linking. The metadata extraction logic is rule-based and fine-tuned to handle UK case formatting conventions.

3. **Sentence-Level Legal Reference Extraction:** Each cleaned document is split into individual sentences (tokenized) utilizing model that was trained on large legal corpus and then processed using a hybrid approach that combines rule-based logic with transformer-backed NLP. Sentences are then scanned using regex and spaCy to identify references to Acts, Sections, and legal terms.

4. **Text Summarization (Recursive Summarization):** An automatic summarization module condenses long case documents and statutes into concise, coherent summaries. These summaries are generated using transformer models designed to retain essential legal meaning. This feature allows users to quickly understand the core themes and decisions in a case without reading the entire document.

5. **Complex Term Simplification:** To make legal documents more accessible, the system includes a simplification feature that identifies legal jargon and explains it in plain English. Using a glossary-based approach combined with GPT-backed interpretation, complex legal terms and phrases are dynamically simplified or explained. This is especially useful for users without a legal background.

6. **Named Entity Recognition Using SpaCy + LegalBERT:** A LegalBERT-based model, integrated via spaCy, is used to identify named entities relevant to law - such as legal statutes, organizations, persons, and temporal expressions. These entities are used to enrich the document context however it isn't fully employed.

7. **GPT powered Chatbot for Legal Document Processing:** A retrieval-augmented chatbot was developed to answer user queries about specific uploaded legal documents. The extracted information can be seamlessly passed to a GPT model that generates a coherent, legally grounded response. This transforms static documents into interactive, query-able interfaces.

8. **Custom Sentence Classification for Legal Clauses:** The system incorporates a fine-tuned transformer model (e.g., RoBERTa or LegalBERT) to classify legal sentences into functional categories such as definitions, obligations, prohibitions, or conditions. This helps structure legal content in a way that improves readability, queryability, and downstream tasks like summarization and compliance checking.

9. **Vector Embedding and Semantic Search Integration:** To support fast and accurate information retrieval, documents are chunked into meaningful units and converted into 1024-dimensional vector embeddings using the `llama-text-embed-v2` model. These embeddings are indexed in a Pinecone serverless vector database, enabling cosine similarity-based semantic search across documents or within a single case.

10. **PostgreSQL and Pinecone Backend Architecture:** A robust backend was designed to manage both structured and unstructured data. The PostgreSQL database stores all metadata, file paths, and act references, while Pinecone handles vector embeddings and semantic queries. This architecture allows for efficient indexing, retrieval, and linking between legal documents and their AI-extracted components

11. **Secure User Login and Session Management:** The system includes secure user authentication to protect access to sensitive legal documents and interactions. Using standard session-based or token-based authentication (e.g., JWT), users can log in and interact with documents in a personalized environment.

12. **Frontend Interface for Interactive Legal Analysis:** A modern frontend interface was built using Next.js to display case documents along with AI-driven insights. Users can view the full case text with highlights for sentences referencing laws, hover over entities to see definitions or simplifications, and interact with chatbot responses in context. The UI is designed to be responsive and user-friendly for both legal professionals and non-experts.

4. Literature Review

Modern legal systems are exposed to advanced societies, providing the formal rules and rights that structure social and economic activity. As one study puts it, "modern societies rely on law as the primary mechanism to direct their development and govern their conflicts," enabling human beings to conduct more complex activities. Meanwhile, the size and complexity of legal corpora have grown exponentially in the digital era. Quantitative research finds that there is an "expansive growth in legal complexity as a function of volume, interconnectivity, and hierarchical structure" of the law. Likewise, legal professionals explain that the "amount of information produced within the legal sphere by its different players ... is overwhelming". In practice, lawyers and judges now deal with huge collections of law, regulation, cases, and contracts, often in multiple jurisdictions and languages. This growth has made legal labor increasingly data-intensive: scholars argue that today's law must be treated as dynamic, networked collections of documents rather than as fixed texts. In short, law remains the foundation of society, but the digital age has made its artifacts – codes, briefs, decisions – much larger and more intricate than before.

Traditional information-retrieval (IR) methods struggle to cope with this deluge of legal information. Early computerized legal research employed boolean and keyword-based search over indexed sets of documents. In these systems, users construct exact-word queries or use simple filters. Word-based search can match only literal text, however, and therefore it lacks semantically equivalent content. For example, keyword systems are prone to return thousands of hits (e.g. "felony murder" yielding 10,000+ cases) without helping users distinguish relevant from non-relevant information. In addition, standard lexical methods "often fail to capture the fine-grained conceptual intersections and rich contextual hints that typify legal questions". The main shortcomings are:

•Exact-match dependency: Users and documents must utilize the same terms when executing keyword searches. Paraphrases, synonyms, or other legal language aren't brought up (Ex - "breach of contract" versus "failure to perform"). Homonyms and abbreviations also stump search (Ex- the word "arm" can either be body part or gun).

• Context insensitivity: Simple search is not sensitive to the context or structure of legal writing. Legal argument often depends on logical relationships (e.g., reference to a definition in statute or precedent), but keyword searches are not sensitive to them. A search for "negligence" will catch any occurrence of the term but can't infer links to other relevant standards (duty of care, causation) unless asked for them.

• Long documents and clauses: Legal documents are unusually lengthy and structured into sections, clauses, and cross-references. Keyword systems treat them as flat text and do not leverage clause hierarchies or follow in-text references automatically. A search

may, bring back a document containing the term but fail to find the specific clause where it appears.
(Zhong *et al.*, 2020)
Consequently, lawyers typically rely on manual iterative improvement, broad Boolean searching (ORs of large sets of terms), and proprietary advanced features (Ex - search maps or concept tools) to get useful results. These approaches are time-consuming and still omit significant material. In short, with growing legal collections and vocabulary, conventional keyword-based approaches show glaring shortcomings – inability to generalize substantially, poor handling of legal structure, and user overload from irrelevant results.

Natural language processing (NLP) and artificial intelligence have been the path researchers have pursued to address these limitations and automate legal argumentation. Large transformer-based language models have transformed text processing in all fields, including law, in recent years. Sophisticated legal NLP systems now "rely on deep learning-based approaches (most notably those building on transformer methods)" as a testament to the victory of models like BERT. For example, LegalBERT brings pre-trained BERT to the legal domain with additional pre-training in legal text. These models attained state-of-the-art or near-state-of-the-art performance for case retrieval, text classification, summarization, and entailment tasks. Even general transformers like RoBERTa (a robustly optimized BERT) became strong baselines for legal tasks. (Another study indicates that RoBERTa attains "very strong performance in legal tasks" comparable to LegalBERT.) (Chalkidis *et al.*, 2020)

These models have been applied in many legal application examples by researchers. NLP can assist with document classification (Ex -. tagging up legal issues within contracts), question answering (extracting answers to legal questions from documents), and summarization (condensing long statutes or opinions). In COLIEE competitions, teams used BERT-based methods to extract relevant case law and even perform logical entailment between statutes and legal questions. Other studies fine-tune GPT-like models to generate law-compliant text or answer legal exam questions. In general, the emergence of large pretrained LMs has brought dramatic progress. A recent study indicates that advances in NLP have "been observed to be appropriate to augment and mechanize operations in the legal profession". However, models need to be appropriately adapted: legal texts are long, domain-specific, and typically multi-lingual, so blind transfer of pre-trained off-the-shelf models is not adequate. Experiments therefore look for solutions such as further pre-training on legal corpuses or tokenization adjustments to suit legal jargon. Great potential despite, these machine-based approaches still pose issues. The next sections analyze the prevalent methods (token adaptation, NER, knowledge graphs) and their limitations when used in actual legal scenarios. (Sachidananda, Kessler and Lai, 2021)

To tackle legal terminology and form, custom preprocessing and info extraction strategies have been developed. Token adaptation refers to the application of the input representation of the model to the legal domain. For instance, tokenizers may be configured to keep intact multi-word legal phrases, or new tokens inserted for frequent legal phrases. While some authors train language models anew from legal text in order to learn domain-specific embeddings, others only continue to pre-train general models over legal corpora (e.g., LegalBERT). Such practices may promote the understanding of low-frequency legal terms and references. Legal documents, however, also require Named Entity Recognition (NER) for labelling domain-specific entities such as statutes, case names, legal concepts, institutions, and people. Legal NER is more advanced than standard NER: models must disambiguate titles and nested names (e.g., "Federal Reserve Board" versus "Board of Governors"), handle references to sections of law (Roman numerals, subsection numbering), and handle co-references over long passages. Because legal documents are long, NER is usually performed with sliding window or document-level inference, which is expensively computationally. Good job, though, NER is crucial for downstream tasks: e.g., building a knowledge graph of legal data is reliant on being able to accurately identify entities and relations between entities.

Knowledge graphs (KGs) attempt to represent legal knowledge in structured form. A legal KG might contain nodes for cases, statutes, courts, and legal concepts, with edges representing citations, enactment relations, or logical entailment. These graphs "formalize [legal] relationships, enabling structured navigation of the domain". By linking related precedents and statutes, a KG can make inferences about context and help with advanced queries (Ex - "show all cases that applied this doctrine"). In reality, the construction of KGs needs NER as well as relation classification to determine how entities are connected. New attempts build KGs by pipelining these operations or even by simply asking large language models to generate triples. Knowledge graphs possess strengths: they represent explicit legal structure and support symbolic reasoning over legal concepts. However, current systems have severe limitations. One such shared issue is that NER and relation extraction are separate modules in most pipelines; it doesn't only require a few models but also breaks their common context, which is most likely throwing away useful cross-task signals. Moreover, KGs usually rely on manually curated ontologies or domain experts' knowledge, so they become brittle and hard to scale across jurisdictions. The legal knowledge graph structure is complex, and as Li et al. note of Chinese law, there is "no unified knowledge graph structure standard in the legal field". In brief, token adaptation and NER/KG techniques enrich legal NLP but remain immature in practice: models struggle with nested references and require high-level legal expertise to develop, and KGs remain labor-intensive to develop and update. (Sabrina Univ-Prof Axel P, 2021)

To surpass keyword matching, recent work has embraced semantic search with dense vector representations. In this setting, documents and queries are embedded into continuous vector spaces (e.g., models like BERT or GPT), so that semantic similarity is measured by distance. Such approaches have revolutionized information retrieval: as one study puts it, embedding "covers semantic senses beyond keyword similarity," helping to

identify relevant documents even when "exact search terms vary." That is, a legal query asked one way can retrieve semantically equivalent texts even if they use different words. More recent RAG (Retrieval-Augmented Generation) methods do it on purpose: they index case law in a vector store so that queries bring back contextually relevant passages rather than raw lexical matches.

Law experiments confirm the promise of semantic methods. Task-oriented case law retrieval research, for example, finds that dense retrieval models (like bi-encoder BERTs or DPR) significantly improve recall compared to vanilla TF-IDF. Embedding-based search can handle synonyms and paraphrasing more naturally (e.g., treat "financial fraud" and "securities violations" as semantically equivalent terms). It also solves the scaling issue: vector indexing enables efficient querying of even millions of documents by approximate nearest-neighbors, whereas boolean querying becomes increasingly slower with growing data. Several legal AI use cases now embed statutes and case text to facilitate semantic lookup and question answering. But there are obstacles: good legal embeddings are trained on large domain corpora, and pure neural search still doesn't capture logical constraints (e.g. citation of statutes). So researchers end up combining vectors with symbolic techniques (e.g. knowledge graphs) to represent meaning and apply legal structure. Generally speaking, semantic search and embeddings have clearly surpassed many of the shortcomings of keyword IR in law but must be well integrated with legal reasoning in order to be accurate and explainable. (Zhong *et al.*, 2020)

More recently, efforts have focused on creating domain-specific AI tools for particular areas of law. For example, privacy law, tax law, and healthcare regulation each have unique language and concepts. In the domain of cybersecurity, researchers report a pressing requirement: although online security is increasingly critical and policy and regulation papers (privacy policies, cyber defense orders, etc.) propagate, there is a shortage of automated tools. Rivas et al. (2023) point out that NLP legal research has hitherto concentrated on privacy policies, to say nothing of neglecting general cybersecurity policies that include guidelines and obligations. They report a "lack of transferability between domain-specific language models" that have been trained, e.g., on privacy text and other security policy. They respond by creating a corpus of U.S. Department of Defense cybersecurity policies (CSIAC-DoDIN) with not only rules but procedures and roles. Baseline experiments on this corpus show that transformer-based models (BERT, RoBERTa, PrivBERT, etc.) can classify policy sections and predict co-occurring topics but only fine-tuned on such domain data. Their observation notes a common void: "the need for more structured data to train Deep Learning models... is a general problem in Legal NLP, and the field of policies does not escape this problem.". Generally, there are still significant gaps in legal AI (Dragoni *et al.*, 2016). General LLMs can deal with text, but few systems are constructed to address legal nuance in areas like cybercrime law or financial regulation. Regulation needs shift rapidly (e.g. GDPR, CCPA, upcoming cybercrime law), but AI solutions lag behind. Also, most AI designs for law have difficulty with real-world data: clean statutes or court opinions are one thing, but messy, mixed-genre documents lawyers encounter every day are quite another. Security and

privacy concerns in areas like cybersecurity also involve robust protection of sensitive information, a challenge not addressed by current models (Yang *et al.*, 2019). Finally, most commercial legal AIs are black boxes; explainability is especially important to lawyers who must be able to trust in a system's reasoning. In short, promising as research has been, existing legal NLP tools still fall short of covering specialized subdomains (e.g., law of cybersecurity) and do not optimally incorporate legal reasoning constraints, thus failing to meet key practical requirements.

The foregoing overview shows that, as vast as the potential of AI/NLP methods for law is, there remain critical gaps – specifically where semantics, domain adaptation, and knowledge integration intersect. Existing approaches are more likely to address only one part of the challenge: e.g., most systems improve search using embeddings, but not structural legal knowledge; some build knowledge graphs but without good language comprehension. In addition, the exponentially growing volume of legal texts and dynamic regulatory landscapes (particularly in the fields of cybersecurity and data protection) create an urgent demand for better tools. Practitioners increasingly need AI that is able to understand legal terminology and reason over legal abstractions. Currently available tools do not always meet those needs: they may recover relevant texts, but not disambiguate legal relations; or they may categorize documents, but not generalize to novel subdomains.

This project aims to bridge exactly that gap by bringing together semantic retrieval and domain-specialized adaptation. Particularly, we target legal document processing in cybersecurity contexts – a lesser-researched area revealed by recent surveys. Leveraging pre-trained legal language models (e.g., LegalBERT) and vector-based semantic search and legal knowledge frameworks, the project will research how to improve document retrieval and processing over current keyword-based systems. This is a timely and relevant strategy: as court workloads grow and specialist compliance demands rise, automated analysis that secures deep semantic sense and legal nuance is needed. In short, evolving needs in legal informatics – from court digitalization to company compliance – demand an AI-based solution. By bridging the gaps outlined, this research hopes to advance legal NLP and deliver tools that address the pressing needs of modern legal practice. (Imogen, Sreenidhi and Nivedha, 2024)(Vayadande *et al.*, 2024)

5. Method of approach

The development and design of this legal document analysis system went through an engineering research-driven iterative process that went through several stages: requirement analysis, data sourcing and preprocessing, NLP model integration, database schema design, UI/UX design, and deployment. Each stage involved evaluating trade-offs between other possible methodologies, testing and perfecting logic, and integrating feedback loops to iteratively improve the solution.

5.1 Data Collection

The case study research was conducted in depth with the help of cybercrime cases acquired from BAILII upon request (British and Irish Legal Information Institute) centered on the Computer Misuse Act 1990. This shift from the Sri Lankan Computer Crime Act No. 24 of 2007 (was the initial focus) to British legislation happened due to case data availability and reliability for model development and training. Responsible professionals provided awareness into how Sri Lanka doesn't have many rich documented cybercrime specific cases. The primary aim of this study was to understand the legal nature of cybercrime litigation and to analyze the complex language and legal reasoning in these documents. But in addition to that, case studies provided valuable insights regarding how cyber offences such as unauthorized access, cyber fraud, and data breaches are interpreted under the UK legal system. By close examination of judgments and opinions, the analysis revealed how lawyers deal with technical evidence, interpret statutory provisions, and deal with the nuances of cybercrime. Such observations were crucial in the development of the AI-based legal document analysis system by guiding the design of natural language models to be employed for simplifying thick legal jargon without compromising the essence of judicial thought.

In parallel with case analysis, a comprehensive review of other relevant legislative and regulatory documents was undertaken. The analysis included key legal frameworks that influence how cybercrime is approached, including:

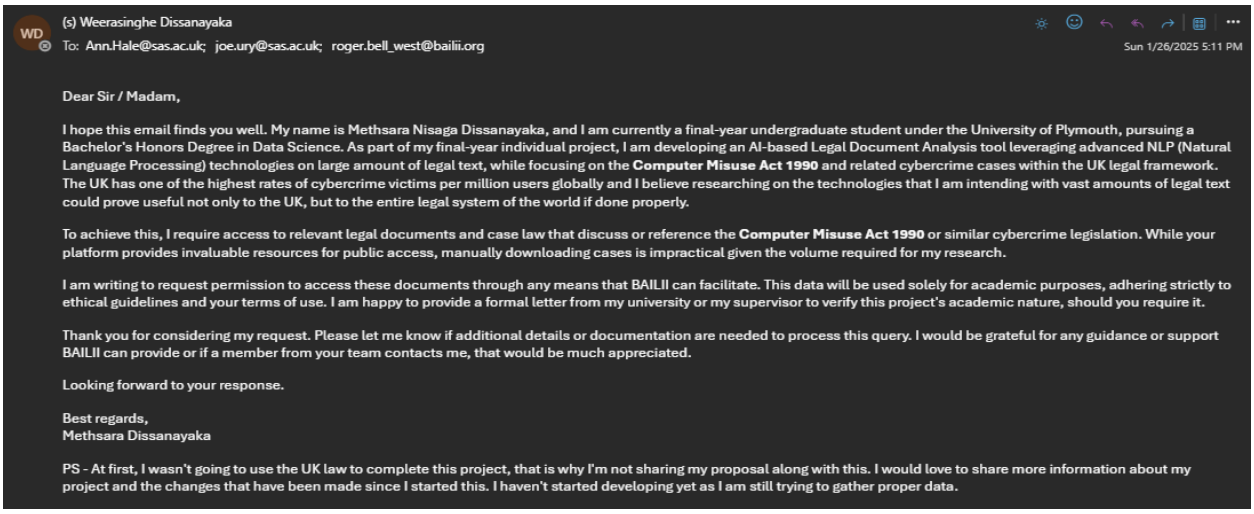
- **Data Protection Act 2018 (DPA) and UK General Data Protection Regulation (UK-GDPR):** These documents underscore the importance of data privacy and set stringent guidelines for handling personal data, aspects that are integral to the interpretation and processing of legal documents.
- **Network and Information Security Directive (NIS2) and Digital Operational Resilience Act (DORA):** These regulations highlight the critical role of operational

resilience and cybersecurity, providing a contextual background for understanding how legal obligations are enforced in the digital realm.

- **UK Operational Resilience Framework:** This framework offers insight into the continuity and robustness expected of organizations in the face of cyber threats, further informing the legal narratives found in cybercrime cases.
- **EU Cybersecurity Act and EU Cyber Resilience Act:** These Acts provide a broader European perspective on cybersecurity standards and resilience, influencing legal arguments and regulatory expectations within the UK.
- **Computer Misuse Act 1990:** As one of the foundational legal documents dealing with unauthorized computer access and cyber offences, it remains a cornerstone in the analysis of cybercrime cases.
- **EU Artificial Intelligence Act:** This emerging legislative framework is beginning to shape discussions around the use of AI in legal settings, particularly in how automated tools process and analyze legal texts.
- **Telecommunications (Security) Act 2021 and Privacy and Electronic Communications Regulations (PCER):** These documents address the security and privacy aspects of digital communications, further adding layers to the legal context of cybercrime investigations.

(Chin, 2025)

This was the initial email that was sent in order to request data from BAILII as most APIs weren't accessible,



WD (s) Weerasinghe Dissanayaka

To: Ann.Hale@sas.ac.uk; joe.ury@sas.ac.uk; roger.bell_west@bailii.org

Sun 1/26/2025 5:11 PM

Dear Sir / Madam,

I hope this email finds you well. My name is Methsara Nisaga Dissanayaka, and I am currently a final-year undergraduate student under the University of Plymouth, pursuing a Bachelor's Honors Degree in Data Science. As part of my final-year individual project, I am developing an AI-based Legal Document Analysis tool leveraging advanced NLP (Natural Language Processing) technologies on large amount of legal text, while focusing on the **Computer Misuse Act 1990** and related cybercrime cases within the UK legal framework. The UK has one of the highest rates of cybercrime victims per million users globally and I believe researching on the technologies that I am intending with vast amounts of legal text could prove useful not only to the UK, but to the entire legal system of the world if done properly.

To achieve this, I require access to relevant legal documents and case law that discuss or reference the **Computer Misuse Act 1990** or similar cybercrime legislation. While your platform provides invaluable resources for public access, manually downloading cases is impractical given the volume required for my research.

I am writing to request permission to access these documents through any means that BAILII can facilitate. This data will be used solely for academic purposes, adhering strictly to ethical guidelines and your terms of use. I am happy to provide a formal letter from my university or my supervisor to verify this project's academic nature, should you require it.

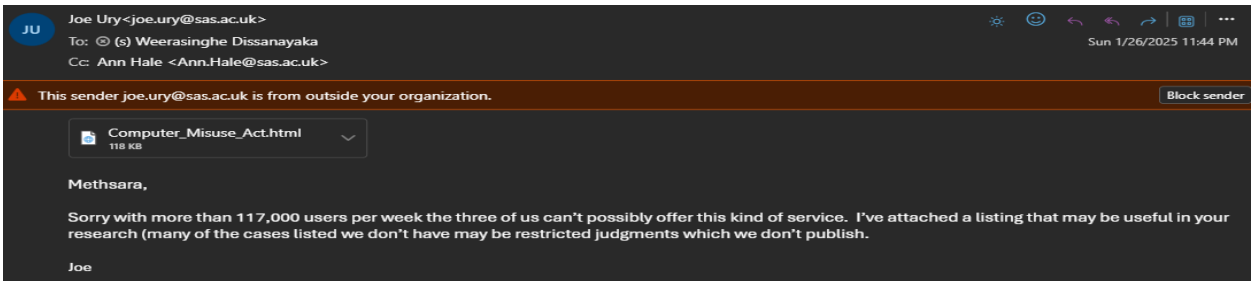
Thank you for considering my request. Please let me know if additional details or documentation are needed to process this query. I would be grateful for any guidance or support BAILII can provide or if a member from your team contacts me, that would be much appreciated.

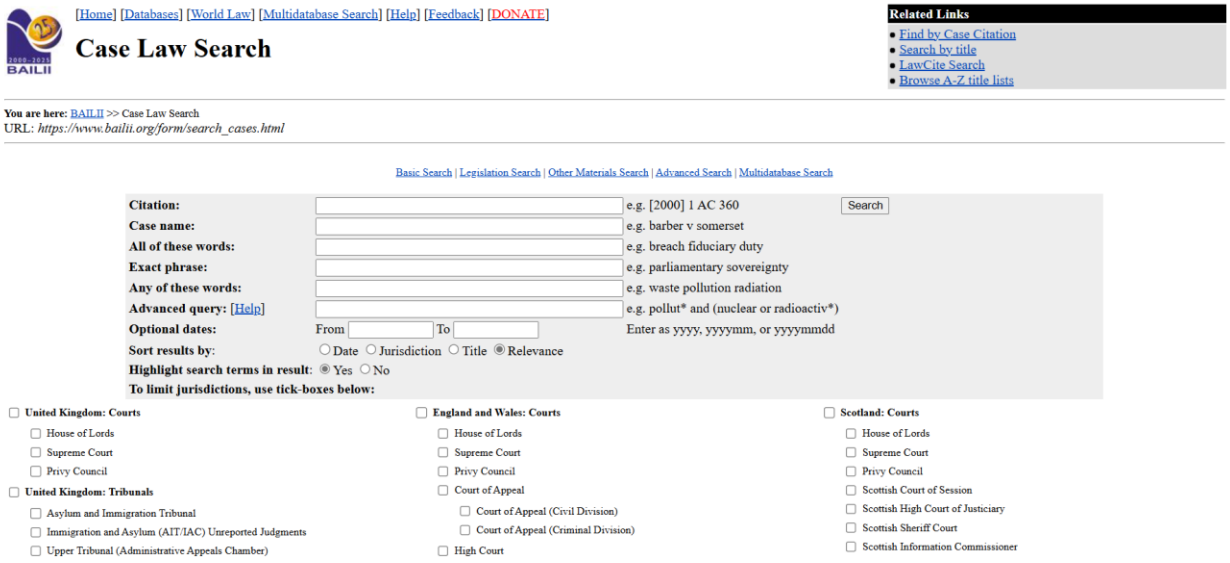
Looking forward to your response.

Best regards,
Methsara Dissanayaka

PS - At first, I wasn't going to use the UK law to complete this project, that is why I'm not sharing my proposal along with this. I would love to share more information about my project and the changes that have been made since I started this. I haven't started developing yet as I am still trying to gather proper data.

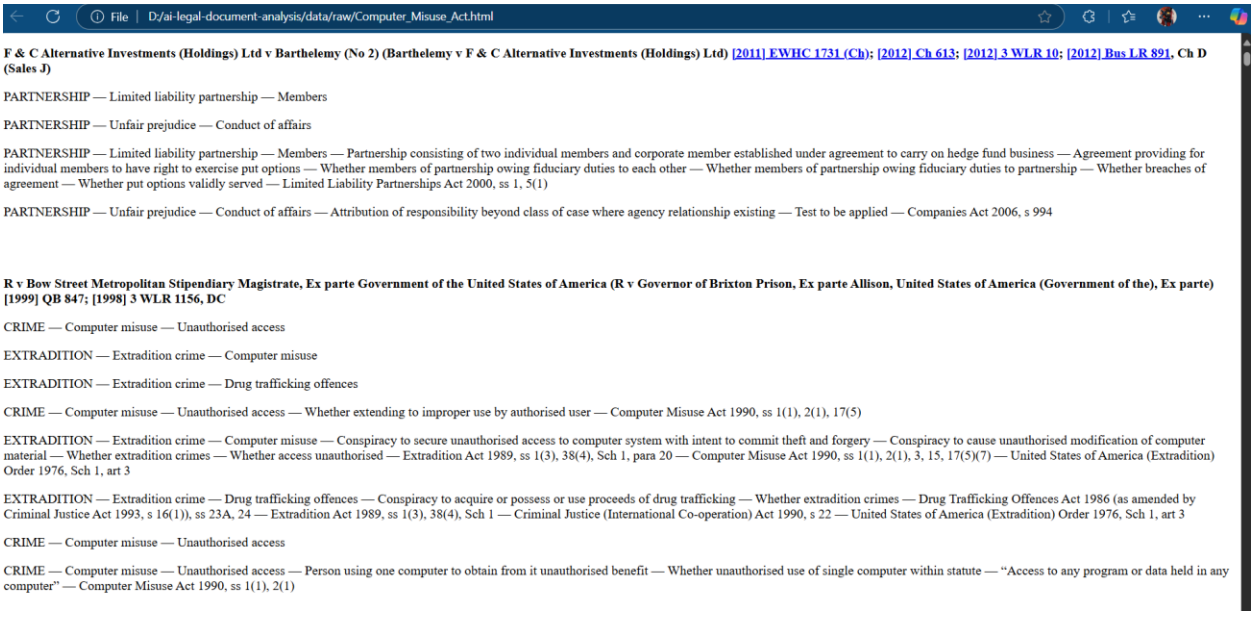
Response,





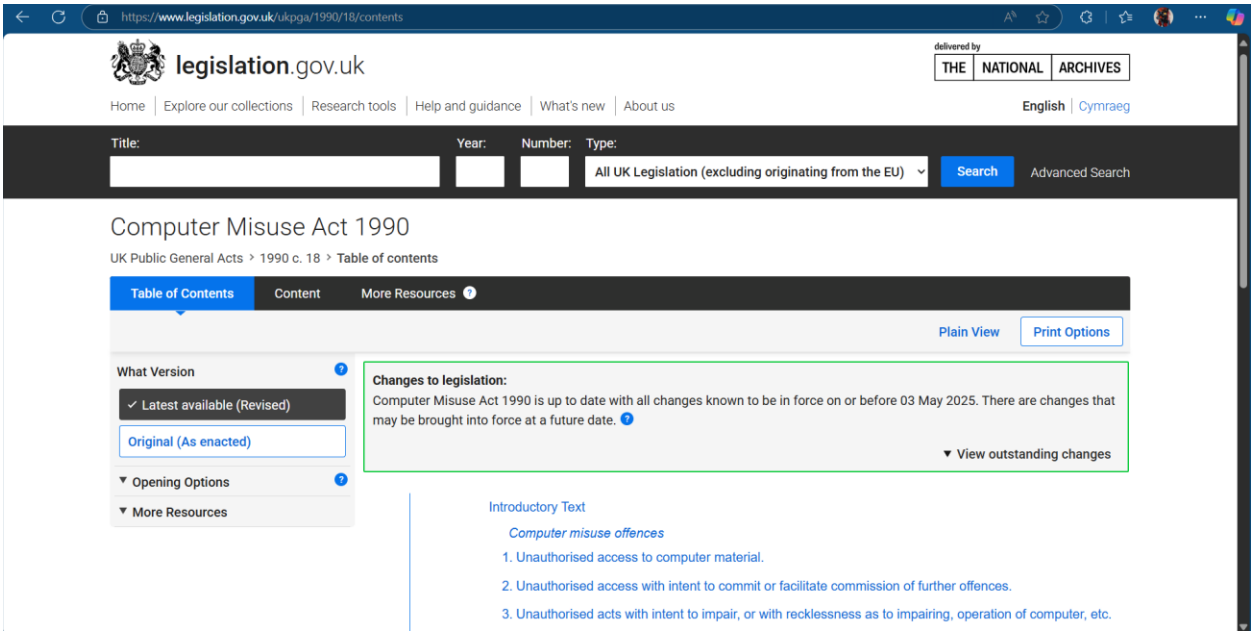
British and Irish Legal Information Institute

The html file that was received regarding the cybercrime related case in the BAILII website,



Even though most of the cases didn't have links to be downloaded, it has to be mentioned that without this, this project could have been in a very dark place. Receiving a complete file consisting of cases related to cybercrimes with downloadable links and structured content proved to be very useful. And the fact that Mr. Joe Ury (Executive Officer, BAILII) tried to provide extended help by offering more clarification has to be admired.

Access to legislation documents was made easy thanks to Legislation.gov.uk.



Knowledge about legal landscape was enhanced by this dual approach of examining legislative papers and case text documents. The detailed case studies and legislative reviews offered useful perspectives on the difficulties faced by attorneys in cybercrime matters as well as court reasoning.

Apart from this case study semi-structured interviews were conducted with diverse stakeholders as well, such as legal practitioners, academics, and industry practitioners by various means as well. Such interviews provided in-depth insights into the practical nuances and challenges of legal document analysis. The interviewees described their first-hand experience of legal research and identified specific issues - such as complexity in legal language, inefficiency in current document review processes, and the need for improved clarity in case summaries. These discussions not only illuminated the issues of functioning in legal practice but also helped to identify the most critical areas where an AI-based solution could significantly impact. The iterative nature of these discussions helped us to constantly refine our approach so that our system remains aligned with real requirements.

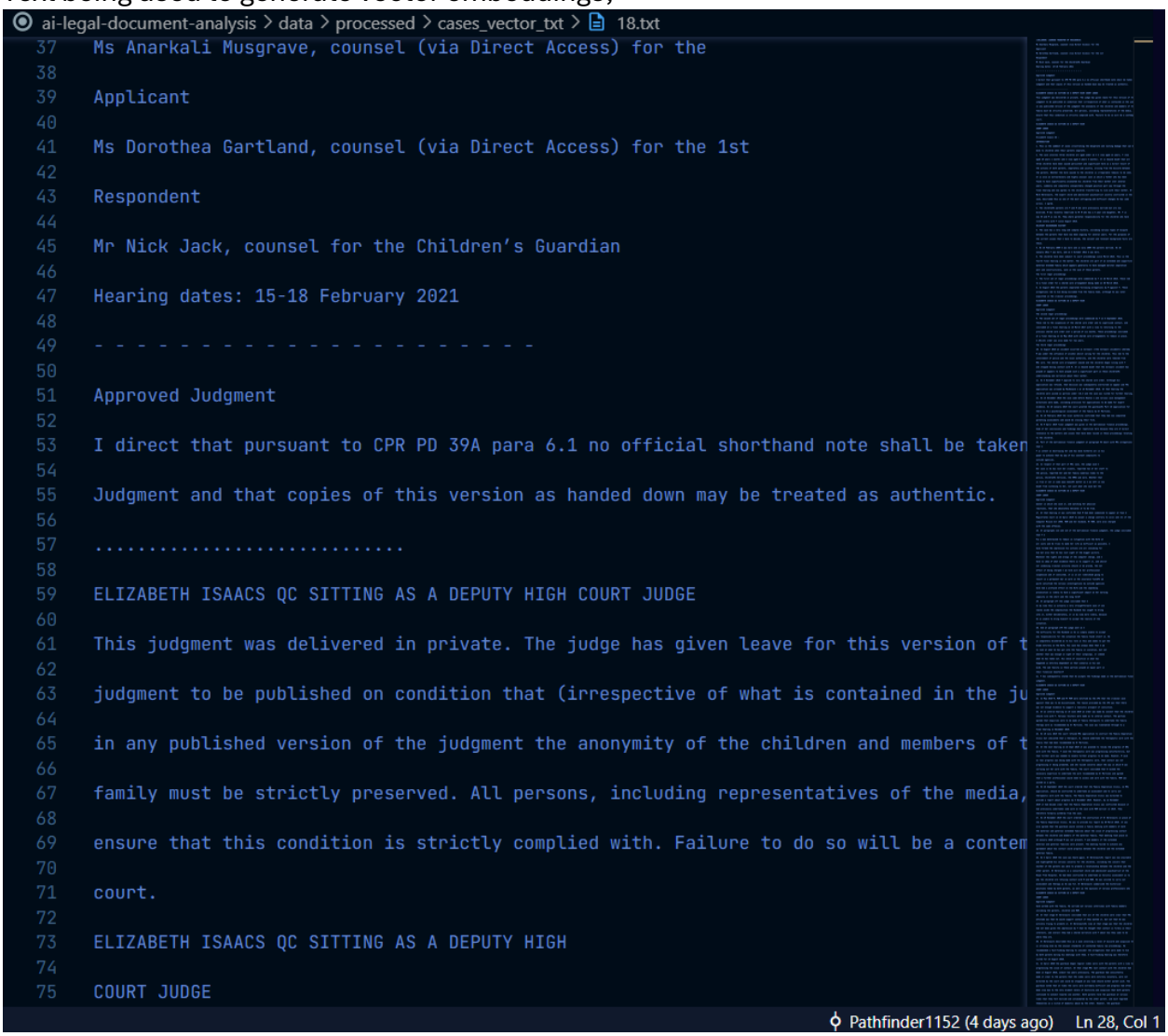
5.2 File Handling and Text Extraction

The system accepts court judgments and legislation documents (user uploaded documents) in multiple formats including PDF, RTF, DOCX, and HTML. These documents are uploaded through a Django-based backend, which handles basic validations (file size, format), sanitizes the filenames, and stores the raw files in a structured directory hierarchy. A Django file upload view stores the raw file and triggers preprocessing via a management command.

The uploaded document is then passed onto a text extraction script where it stores the extracted text as a .txt file in a separate directory. To extract text from PDF, DOCX, RTF document the libraries pdfplumber, py-docx, striprtf libraries were used respectively. Two ways of text extraction methods were employed: one being used for metadata, sentence extraction and the other being used for vector embeddings. These files were stored in two distinct locations as the text files that are being employed to perform metadata, sentence extraction is not separated by paragraphs, as that could lead to inconsistencies in identifying necessary instances, and for the vector embeddings to maintain context between paragraphs, text has to be extracted per natural paragraphing.

Hence Ex- text being used to extract metadata,

Text being used to generate vector embeddings,



5.3 Text Cleaning and Metadata Extraction

After the text files are stored, before metadata is extracted from them, the text gets cleaned using a custom cleaning function clean_paragraph_text(). This function performs multiple targeted preprocessing steps to normalize and sanitize the text before any Natural Language Processing (NLP) tasks occur,

- **Fixes hyphenated or broken words split across line breaks**, common in OCR-processed or justified PDFs, using regex rules to merge fragments correctly. During the extraction process, some words can get separated unintentionally (Ex- autho risation) due to OCR mistakes etc. So, this function employs a robust method to identify if either of those words is a real word or not according to the nltk library, and

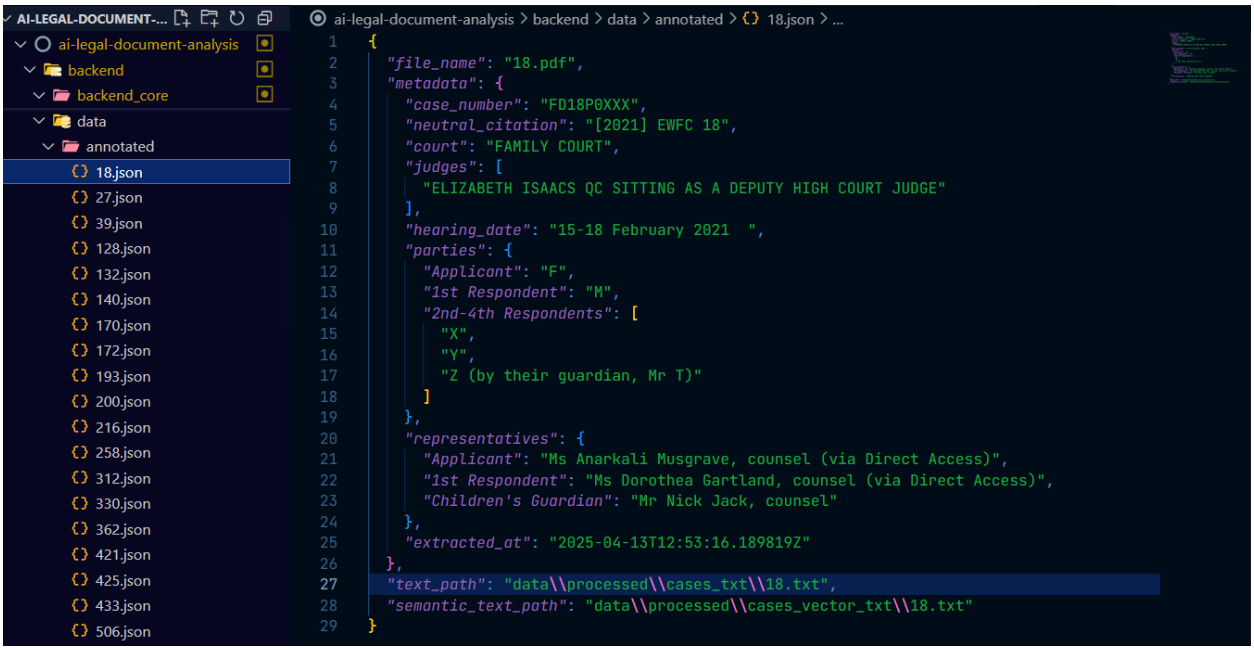
if they aren't and the "words" are short, they get added together (logic was refined upon common observations)

- **Normalizes whitespace**, including removal of excessive spaces, tabs, and inconsistent indentation. This ensures uniformity in paragraph structure across documents.

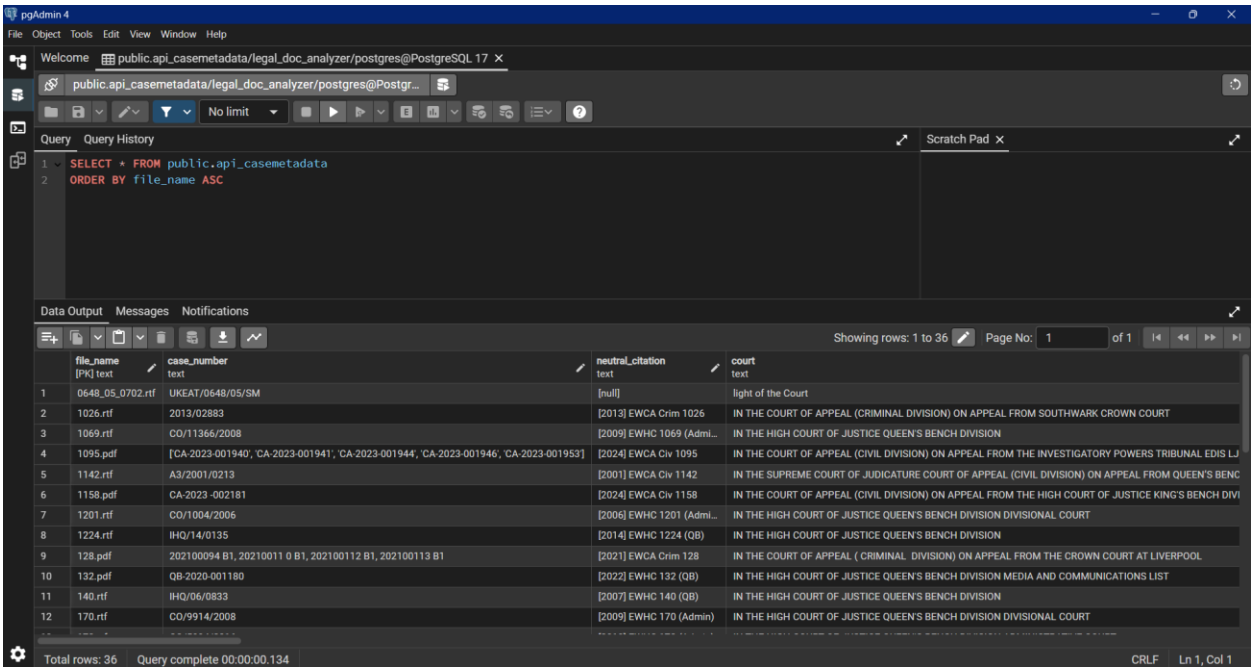
- **Handles Unicode cleanup**, replacing non-breaking spaces, smart quotes, and typographic anomalies with their plain-text equivalents to improve tokenizer accuracy.

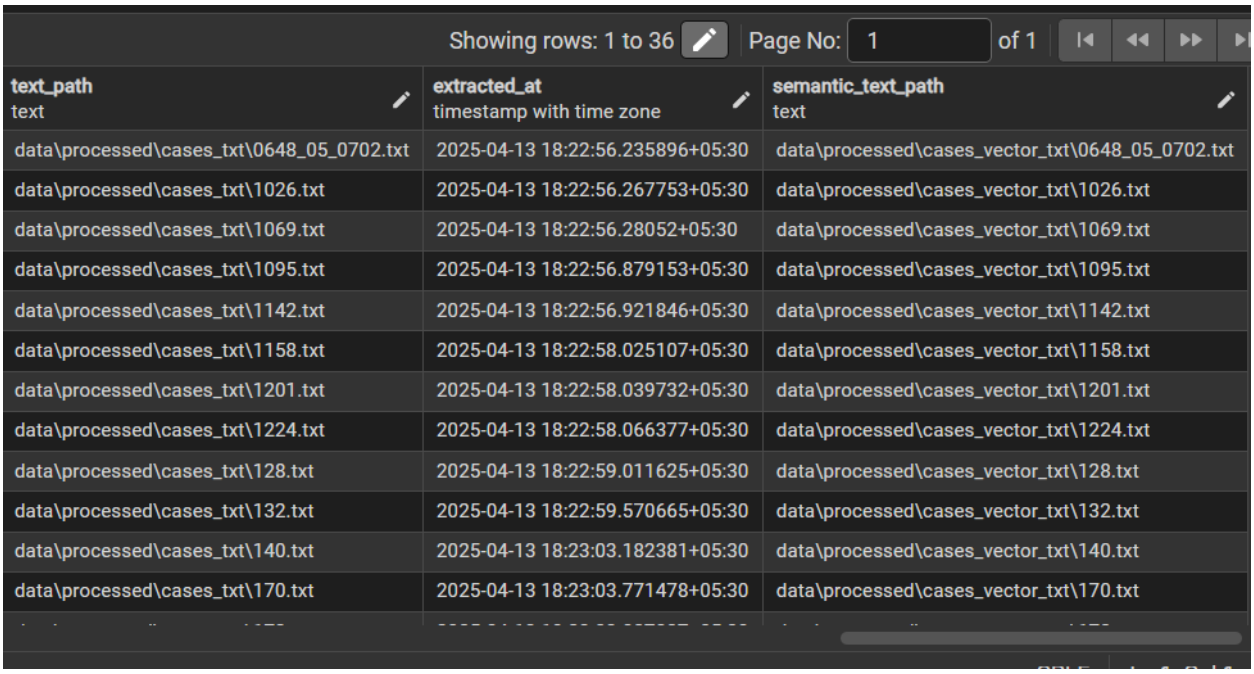
After cleaning, a REGEX based script is called upon to extract key metadata from the uploaded document which stores the metadata in JSON files. When these JSON files are stored, they contain the following information regarding the documents extracted using the metadata_extractor.py script along with the two text paths (raw and semantic) for the relevant document:

- Case/judgment name
- Source file name
- Parties involved
- Judge(s)
- Jurisdiction
- Date of judgment



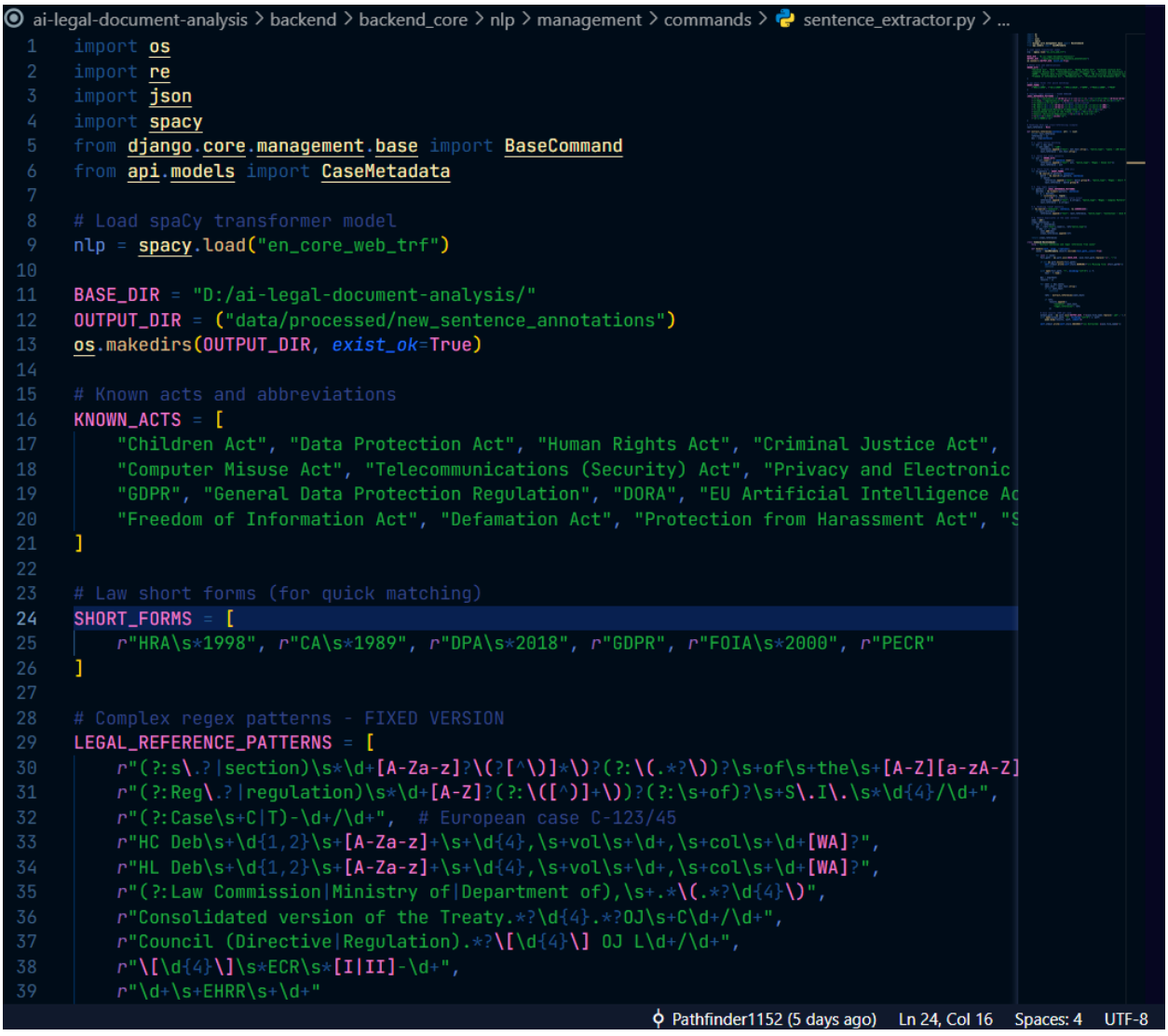
These JSON files are then passed into an ingestion script that connects the backend to a PostgreSQL database where it stores all the data regarding the file which can be used for more convenient calls.


pgAdmin 4
File Object Tools Edit View Window Help
Welcome
public.api_casemetadata/legal_doc_analyzer/postgres@PostgreSQL 17
No limit
Query Query History
Scratch Pad
SELECT * FROM public.api_casemetadata
ORDER BY file_name ASC
Data Output Messages Notifications
Showing rows: 1 to 36 Page No: 1 of 1
file_name [PK] text
case_number text
neutral_citation text
court text
1 0648_05_0702.rtf UKEAT/0648/05/SM [null] light of the Court
2 1026.rtf 2013/02883 [2013] EWCA Crim 1026 IN THE COURT OF APPEAL (CRIMINAL DIVISION) ON APPEAL FROM SOUTHWARK CROWN COURT
3 1069.rtf CO/11366/2008 [2009] EWHC 1069 (Admi... IN THE HIGH COURT OF JUSTICE QUEEN'S BENCH DIVISION
4 1095.pdf ['CA-2023-001940', 'CA-2023-001941', 'CA-2023-001944', 'CA-2023-001946', 'CA-2023-001953'] [2024] EWCA Civ 1095 IN THE COURT OF APPEAL (CIVIL DIVISION) ON APPEAL FROM THE INVESTIGATORY POWERS TRIBUNAL EDIS LJ
5 1142.rtf A3/2001/0213 [2001] EWCA Civ 1142 IN THE SUPREME COURT OF JUDICATURE COURT OF APPEAL (CIVIL DIVISION) ON APPEAL FROM QUEEN'S BENC
6 1158.pdf CA-2023 -002181 [2024] EWCA Civ 1158 IN THE COURT OF APPEAL (CIVIL DIVISION) ON APPEAL FROM THE HIGH COURT OF JUSTICE KING'S BENCH DIVI
7 1201.rtf CO/1004/2006 [2006] EWHC 1201 (Admi... IN THE HIGH COURT OF JUSTICE QUEEN'S BENCH DIVISION DIVISIONAL COURT
8 1224.rtf IHQ/14/0135 [2014] EWHC 1224 (QB) IN THE HIGH COURT OF JUSTICE QUEEN'S BENCH DIVISION
9 128.pdf 202100094 B1, 20210011 0 B1, 202100112 B1, 202100113 B1 [2021] EWCA Crim 128 IN THE COURT OF APPEAL (CRIMINAL DIVISION) ON APPEAL FROM THE CROWN COURT AT LIVERPOOL
10 132.pdf QB-2020-001180 [2022] EWHC 132 (QB) IN THE HIGH COURT OF JUSTICE QUEEN'S BENCH DIVISION MEDIA AND COMMUNICATIONS LIST
11 140.rtf IHQ/06/0833 [2007] EWHC 140 (QB) IN THE HIGH COURT OF JUSTICE QUEEN'S BENCH DIVISION
12 170.rtf CO/9914/2008 [2009] EWHC 170 (Admin) IN THE HIGH COURT OF JUSTICE QUEEN'S BENCH DIVISION DIVISIONAL COURT
Total rows: 36 Query complete 00:00:00.134
CRLF Ln 1, Col 1


Showing rows: 1 to 36 Page No: 1 of 1
text_path text
extracted_at timestamp with time zone
semantic_text_path text
data\processed\cases_txt\0648_05_0702.txt 2025-04-13 18:22:56.235896+05:30 data\processed\cases_vector_txt\0648_05_0702.txt
data\processed\cases_txt\1026.txt 2025-04-13 18:22:56.267753+05:30 data\processed\cases_vector_txt\1026.txt
data\processed\cases_txt\1069.txt 2025-04-13 18:22:56.28052+05:30 data\processed\cases_vector_txt\1069.txt
data\processed\cases_txt\1095.txt 2025-04-13 18:22:56.879153+05:30 data\processed\cases_vector_txt\1095.txt
data\processed\cases_txt\1142.txt 2025-04-13 18:22:56.921846+05:30 data\processed\cases_vector_txt\1142.txt
data\processed\cases_txt\1158.txt 2025-04-13 18:22:58.025107+05:30 data\processed\cases_vector_txt\1158.txt
data\processed\cases_txt\1201.txt 2025-04-13 18:22:58.039732+05:30 data\processed\cases_vector_txt\1201.txt
data\processed\cases_txt\1224.txt 2025-04-13 18:22:58.066377+05:30 data\processed\cases_vector_txt\1224.txt
data\processed\cases_txt\128.txt 2025-04-13 18:22:59.011625+05:30 data\processed\cases_vector_txt\128.txt
data\processed\cases_txt\132.txt 2025-04-13 18:22:59.570665+05:30 data\processed\cases_vector_txt\132.txt
data\processed\cases_txt\140.txt 2025-04-13 18:23:03.182381+05:30 data\processed\cases_vector_txt\140.txt
data\processed\cases_txt\170.txt 2025-04-13 18:23:03.771478+05:30 data\processed\cases_vector_txt\170.txt

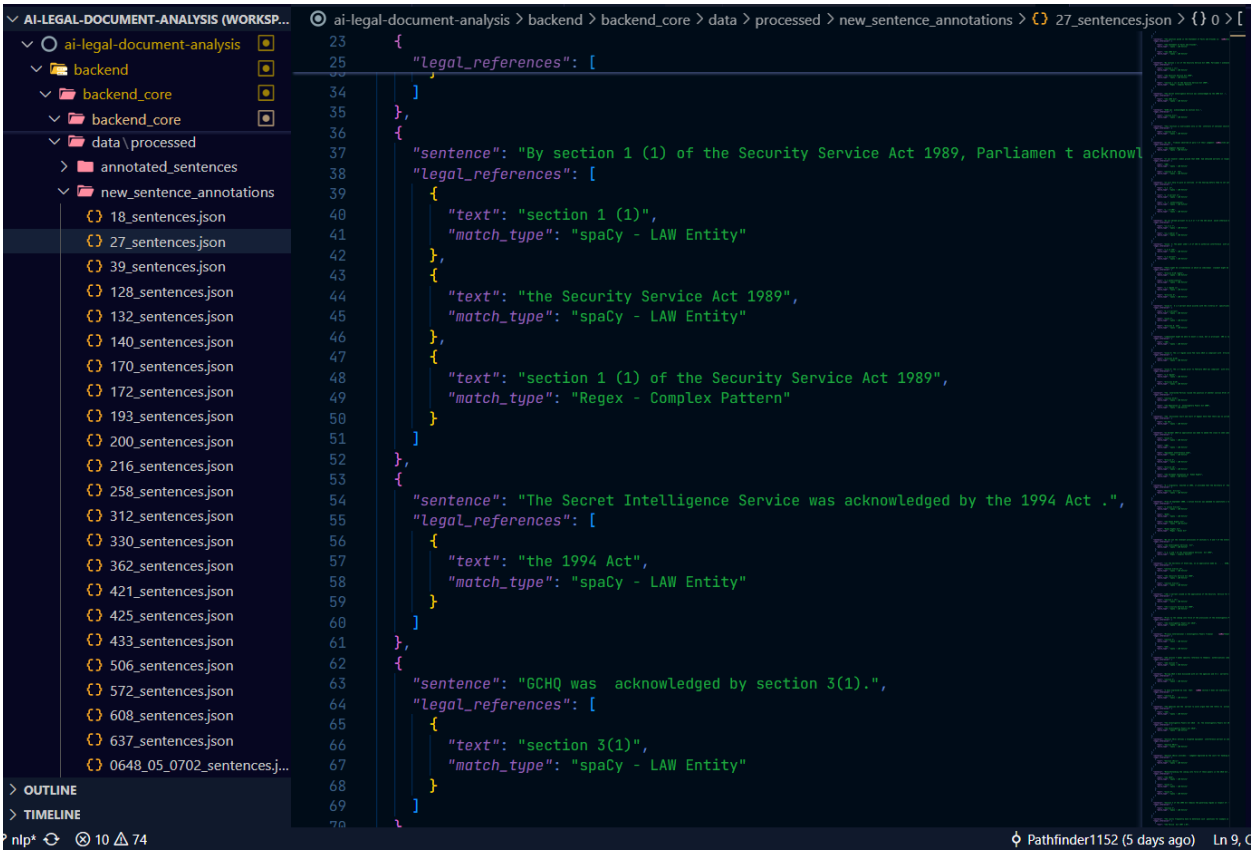
5.4 Sentences and Legal References Extraction

Following text cleaning, each document goes through a python script that utilizes LegalBERT for tokenization (as it is trained on legal text corpora, it was selected) and SpaCy's en_core_web_trf model that is trained on a vast amount of text data but is also capable of identifying different entities including **LAW.** This is one of the key parts of this project, although it isn't the best way of doing things, it could still be implemented as most of the population knows that the way that a legal document is referenced in case text is extremely structured. There can be different variations according to jurisdiction and etc. but there are only a few defined ways to cite a legal document, and it is a rule followed by everyone in the legal industry. So even though it isn't being done using a fine-tuned model, through a hybrid approach using the SpaCy library and REGEX functions, this function was implemented.



Several REGEX patterns were used including the ones above to make sure the system captures as much information as possible. And, since the system is also using the SpaCy library, the system is able to extract almost every instance where an “object” is identified as a LAW entity, which is pretty convenient.
Ex-
Act references:
e.g., "Computer Misuse Act 1990", "Terrorism Act 2006"
Detected using named patterns that capture common UK act phrasing, including year anchors and variants like “the 1990 Act”.
Section references:
e.g., "s. 3(1)", "Section 44", "section 11(b) of the Act"
These require syntax-aware patterns that handle nested parentheses, case insensitivity, and references to “this Act” or “that Act” when contextually relevant.

Which results in storing these extracted instances in JSON files ready to be passed onto another process. There were other variations of this same process to see which one produces the more valuable, hence explaining the number of different folders in the picture below. It can be observed from the right side of the screen that, this script flagged so many instances thus resulting in it being used as the main script for now. Further analysis could have improved this drastically; however it would require a more structured, labelled dataset.



5.5 Text Chunking, Vector Embedding and Semantic Search

To aid effective semantic search, retrieval-augmented generation (RAG), and document-level query, each case document is divided into overlapping segments of text before embedding. This allows for context completeness as well as proper matching in spite of varied sentence structures in legal documents.

Chunking Strategy:
Each document is divided into chunks of approximately 200–300 words, and a 50-token overlap is provided between two successive chunks

- Preserve sentence continuity across chunk boundaries
- Maintain context for references that span across paragraphs
- Improve recall during semantic search, especially for complex legal reasoning or cross-referenced clauses

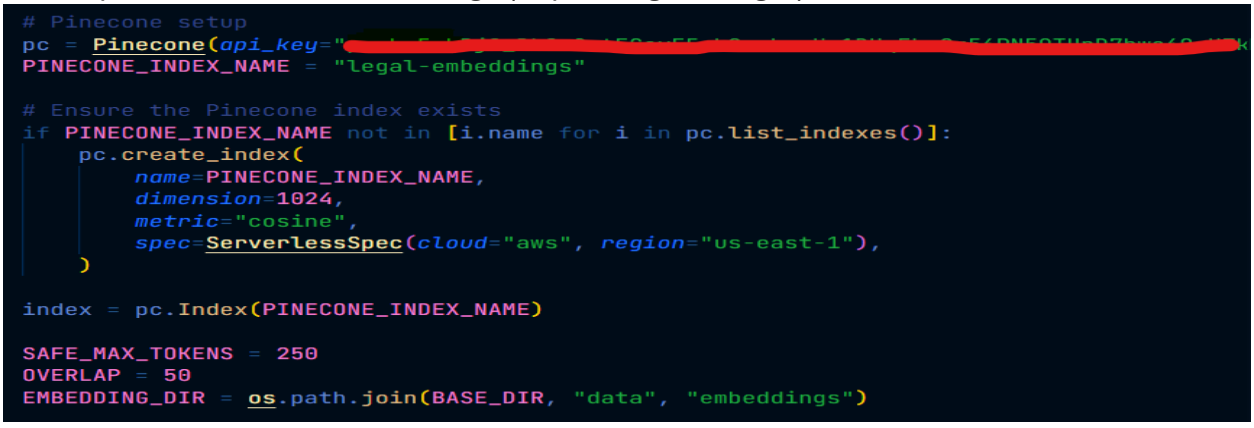
Chunk size was tuned to strike a balance between:

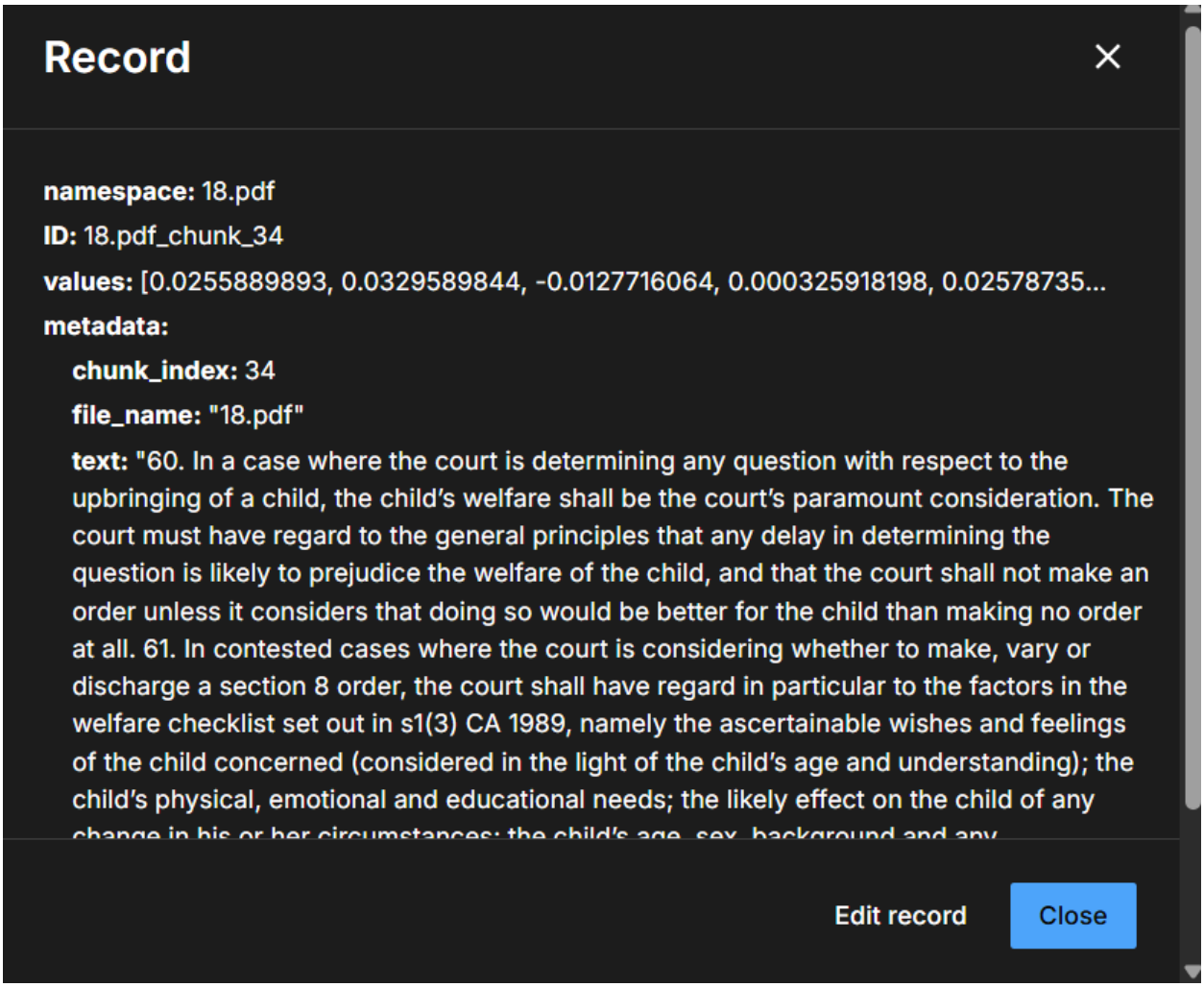
- **Semantic coherence** (longer chunks capture full legal arguments however it could be harder to interpret with the size of the chunk)
- **Token limits of the embedding model** (2048-token cap)
- **Inference speed and downstream latency**

Embedding with LLaMA Model
Each chunk is passed through **Meta's llama-text-embed-v2 model**, chosen for its performance in legal/technical domains and its compatibility with production inference environments.

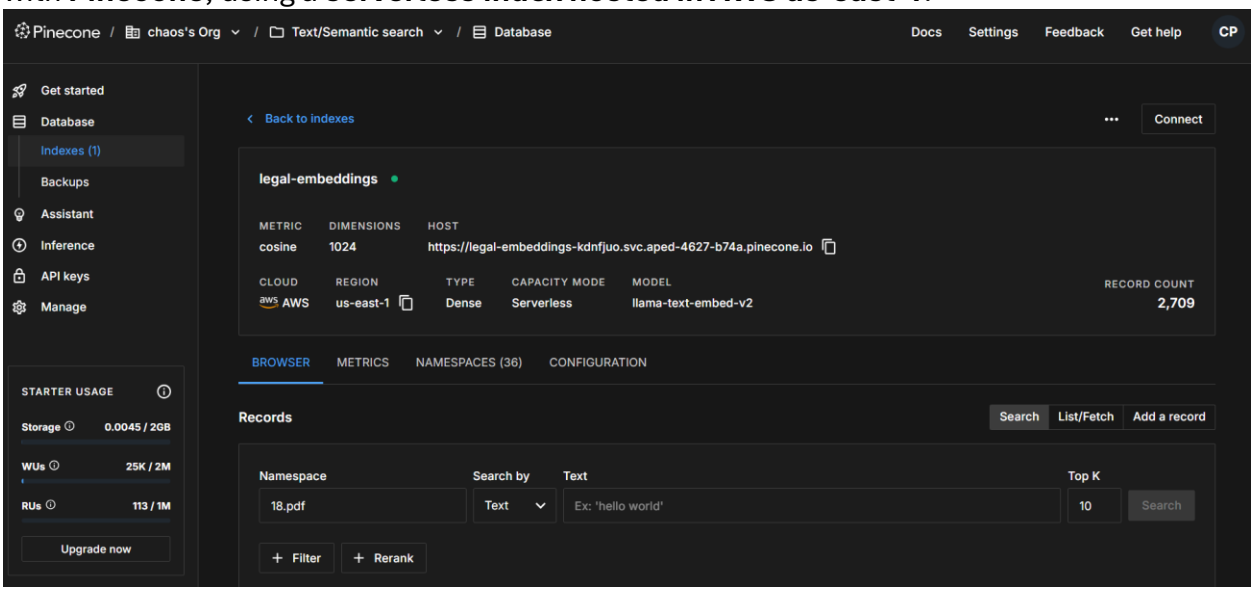
Key model characteristics:

- **1024-dimensional embeddings**
- **Supports up to 2048 input tokens per chunk**
- **Cosine similarity-optimized vectors** for better alignment with modern vector DBs
- **Latency**: ~1.6 seconds per chunk on a T4 GPU, which translates to 20–40 seconds per full document on average (depending on length)



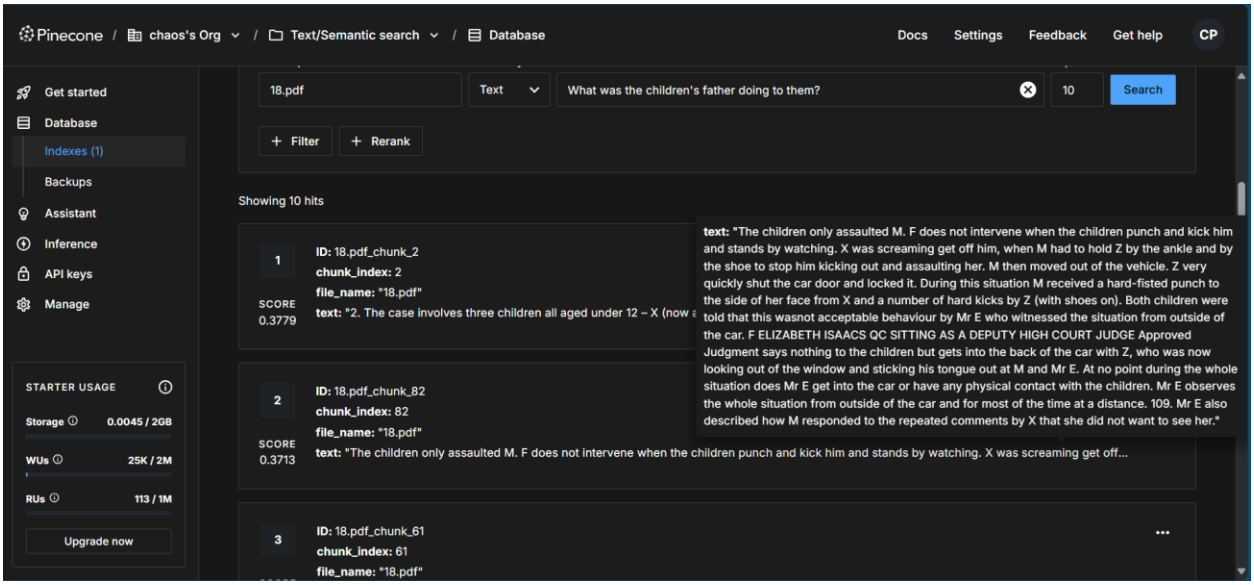


For persistent and scalable storage of these vector embeddings, the system integrates with **Pinecone**, using a **serverless index hosted in AWS us-east-1**.

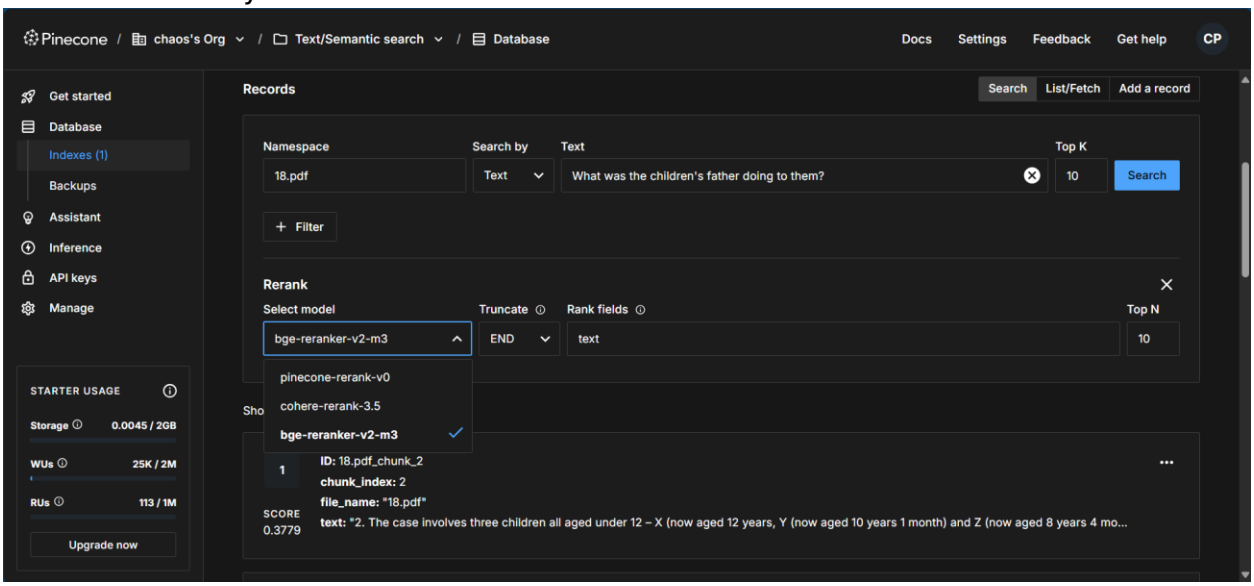


This has been fully wired into the Django backend and Next.js frontend to allow a user to upload a document and query inside their own document using semantic search.

Ex-



The scores represent how similar or relevant is the retrieved chunk is with the query, and pinecone also ranks the top answers using a reranking model (bge-reranker-v2-m3) to help the user to identify relevant answers.



5.6 Document Summarization

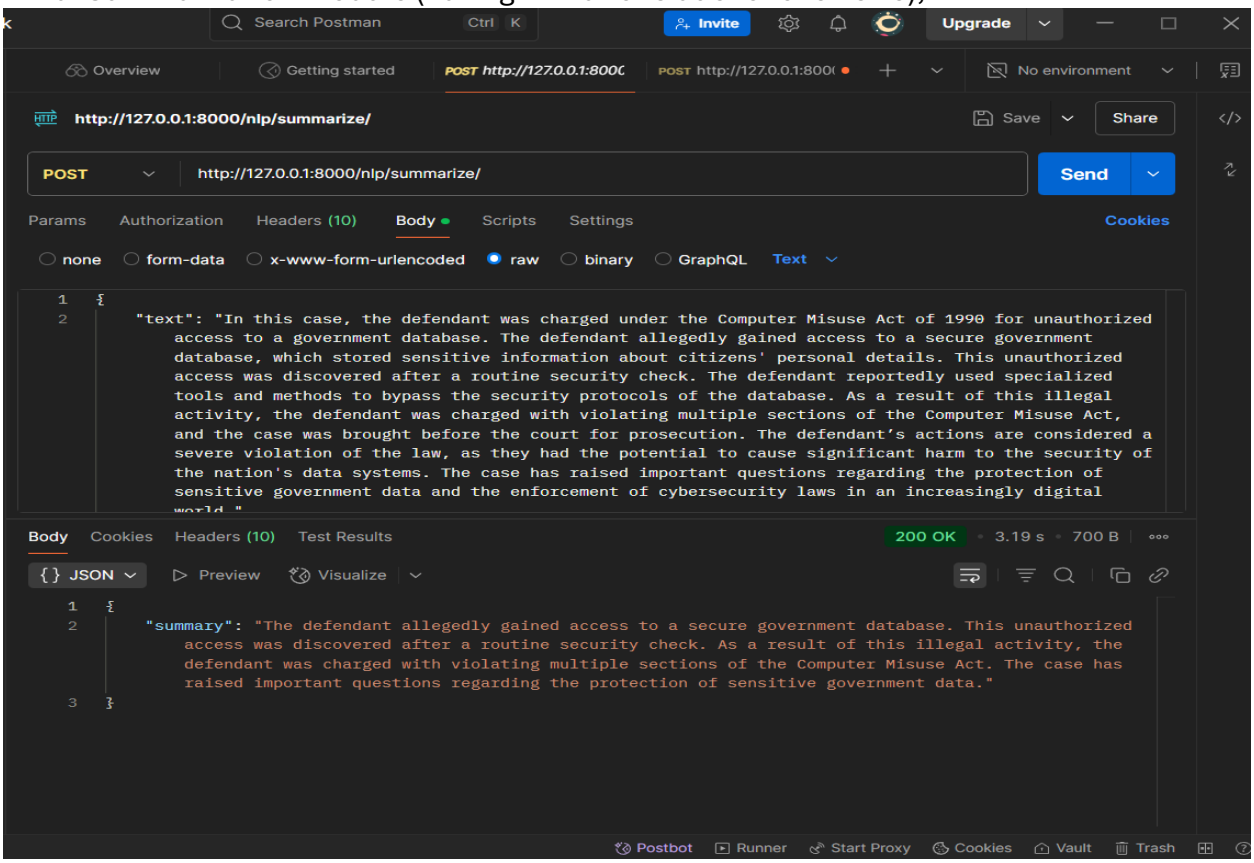
Once sentence-level data is extracted and stored, a recursive summarization technique is applied to long-form case documents. The goal is to generate a high-level summary that captures the most relevant information regarding the specific case.

Approach**:**

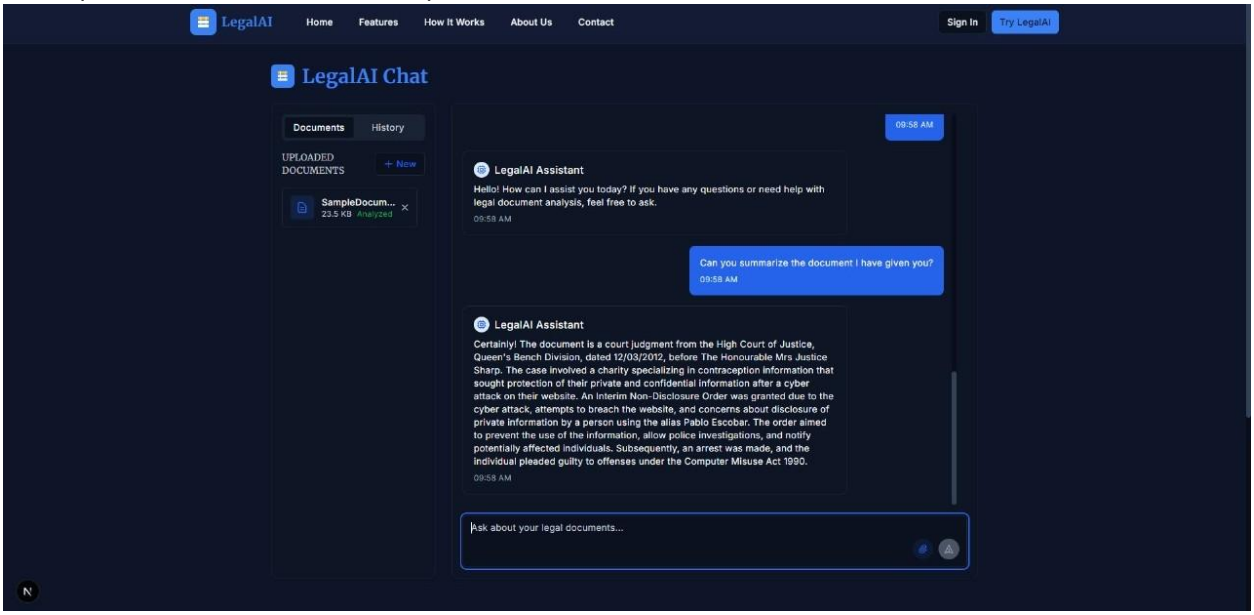
- The full document is chunked using the same logic as for embedding (approx. 200–300 word segments).
- Each chunk is summarized using an LLM (Facebook-BART), focusing on legal reasoning and factual content.
- These summaries are then recursively summarized again -like a *summary of summaries* - until a concise yet representative overview is produced (until the input token count matches the maximum input token count of the model).

This multi-level approach ensures that even very long case documents (10,000+ words) are distilled without losing core detail, and it outperforms naive summarization that misses deeper structure like issue framing or judgment logic. Summaries are stored alongside each document and exposed in the front-end viewer as an optional overview for users or they can just be called anytime.

Initial Summarization module (having limitations due to token size),



After (recursive summarization),



5.8 Text Simplification

Legalese is notoriously inaccessible. For greater useability, especially among non-lawyers, there is an added layer of simplification following processing with two complementary methods:

1. Glossary-Based Substitution
A specially curated set of UK legal definitions and procedural terms (Ex - "fraudulent training", "cyber espionagge") substitutes complex phrases with simple English equivalents. It is used at render time through regex and a map dictionary.

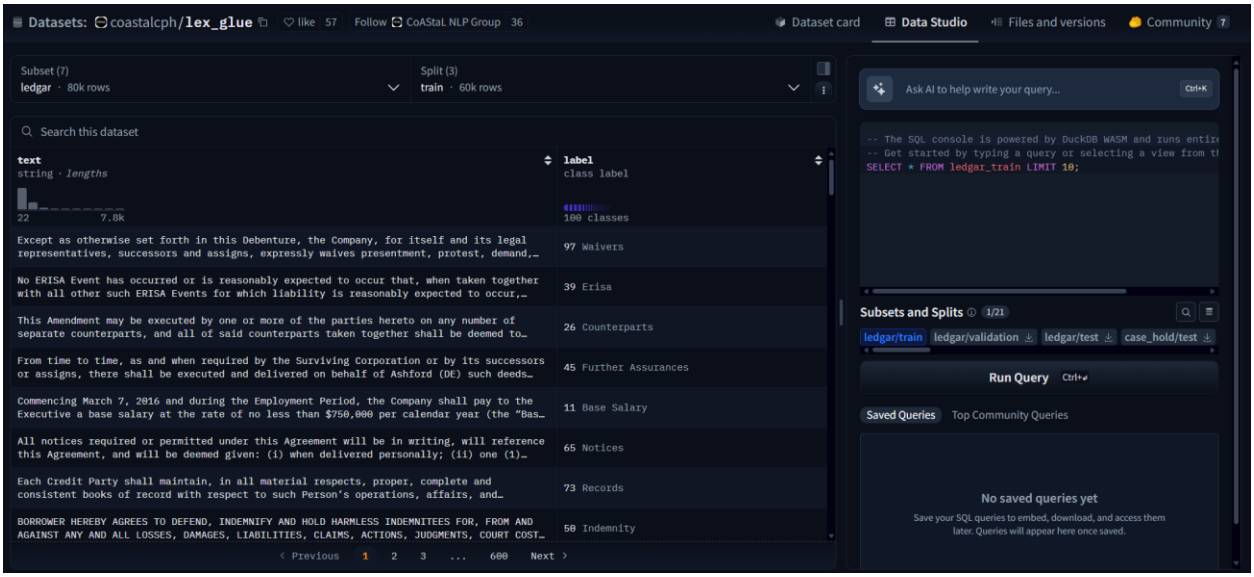
2. Sentence Simplification based on GPT
For boundary cases where simplification under rule is not possible, a GPT-4 prompt is used to paraphrase sentences preserving legal semantics. This is used selectively for long or highly nested sentences (determined by token length and parse depth.

5.9 Legal Text Classification (Fine-Tuned RoBERTa model)

To improve both retrieval precision and overall quality of downstream interactions, each sentence from the processed legal documents is labeled with a transformer model derived from RoBERTa, which was fine-tuned for this particular task. Training was accomplished by combining the LEDGAR dataset with additional domain-specific annotations present in UK court judgments.

The primary objective of this classification layer is to assign functional tags to each sentence, categorizing it into various legal clause categories such as Judgment, Fact, Charge, Reference, Argument, etc. These tags enable semantically meaningful filtering, i.e., retrieving sentences related to judicial reasoning, and enhance the contextual relevance of information passed to generative components such as the GPT-based chatbot.

Model Training and Implementation:
The model was trained over a dataset of approximately 80,000 labeled legal sentences with 60,000 used for training, 10,000 for validation, and the remainder for testing. Training was carried out using PyTorch together with HuggingFace Transformers.



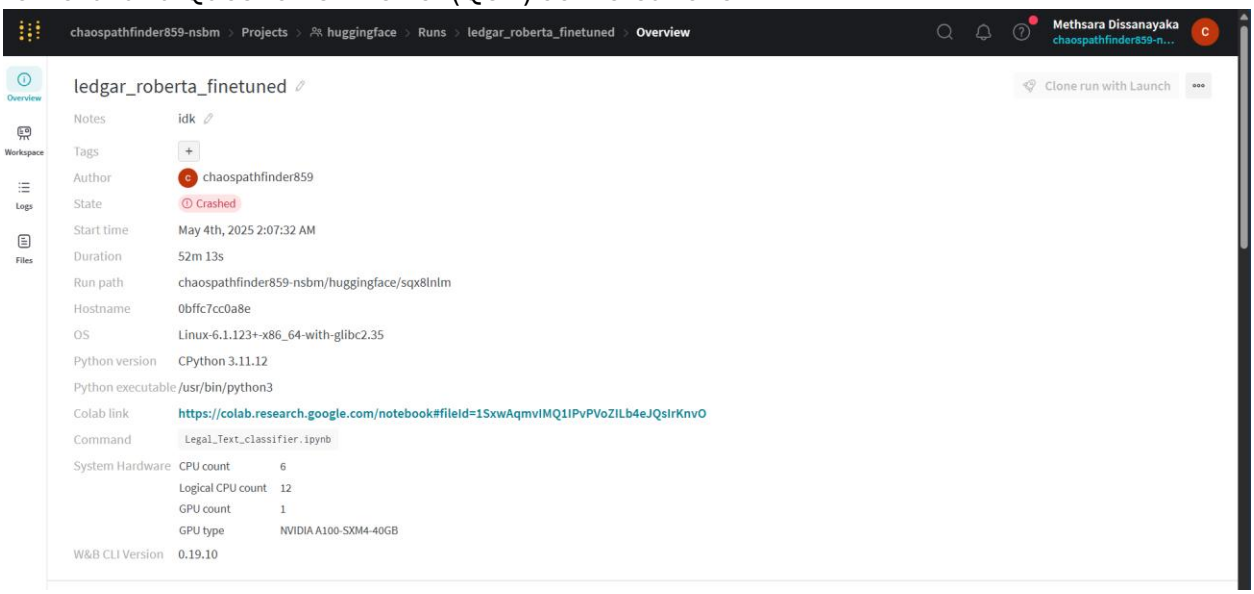
(LEDGAR Dataset)

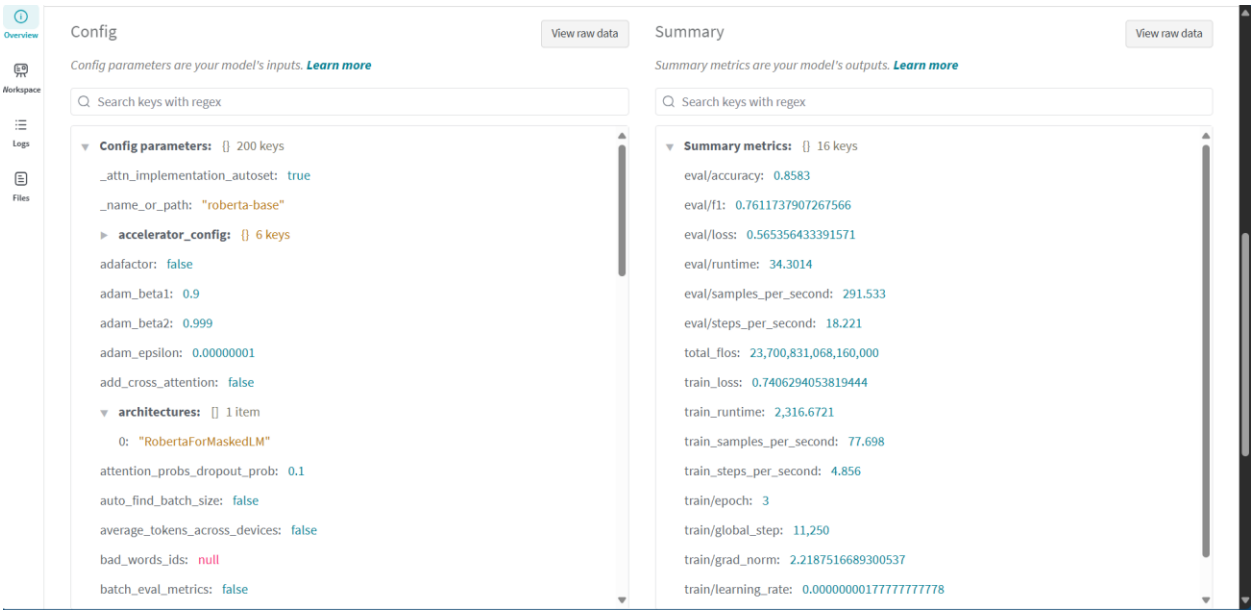
Validation on a held-out UK-specific test set resulted in more than 87% classification accuracy, demonstrating good generalization over varying case types and writing styles.

This model is also fully integrated into the backend pipeline of processing. At the sentence extraction stage, each sentence is tagged, and its predicted tag is encoded directly into per-sentence JSON output. Along with providing dynamic filtering and frontend interaction, these tags provide additional context while constructing prompts for large language

models (LLMs) like GPT-4, thereby making generated responses more targeted and interpretable.

In effect, the classification module provides semantic scaffolding for unstructured legal text so that it becomes presentable in a way that aligns with human legal reasoning. This enhancement significantly improves the system's responsiveness in legal information retrieval and Question & Answer (Q&A) conversations.



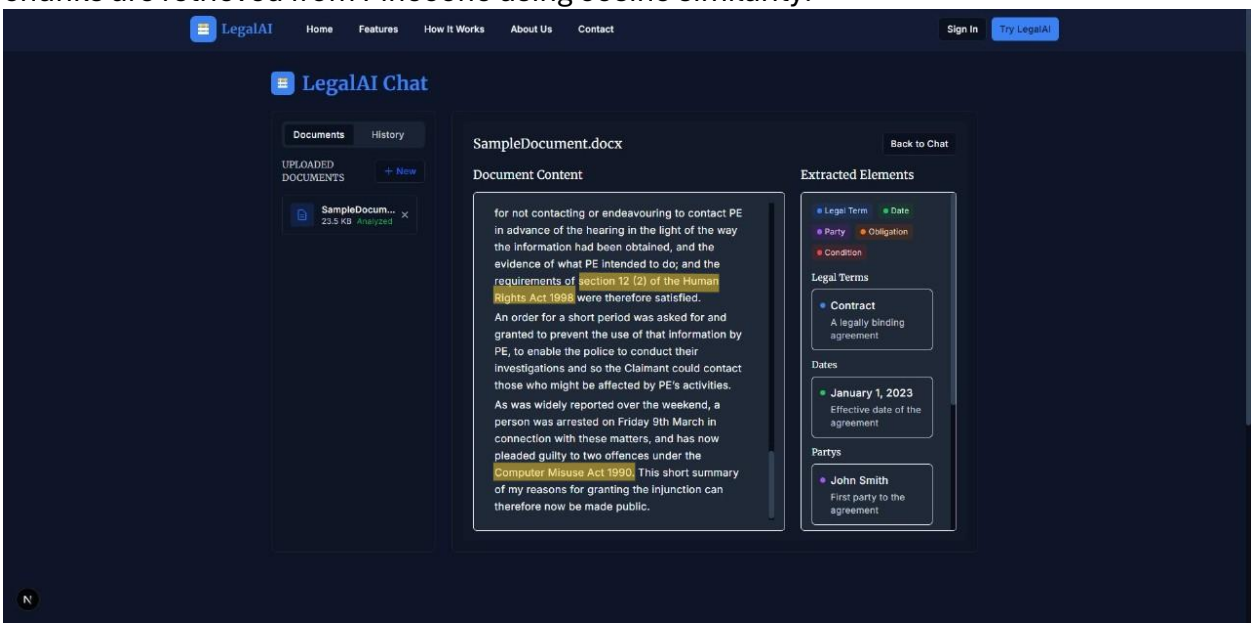


5.10 GPT - Powered Legal Chatbot

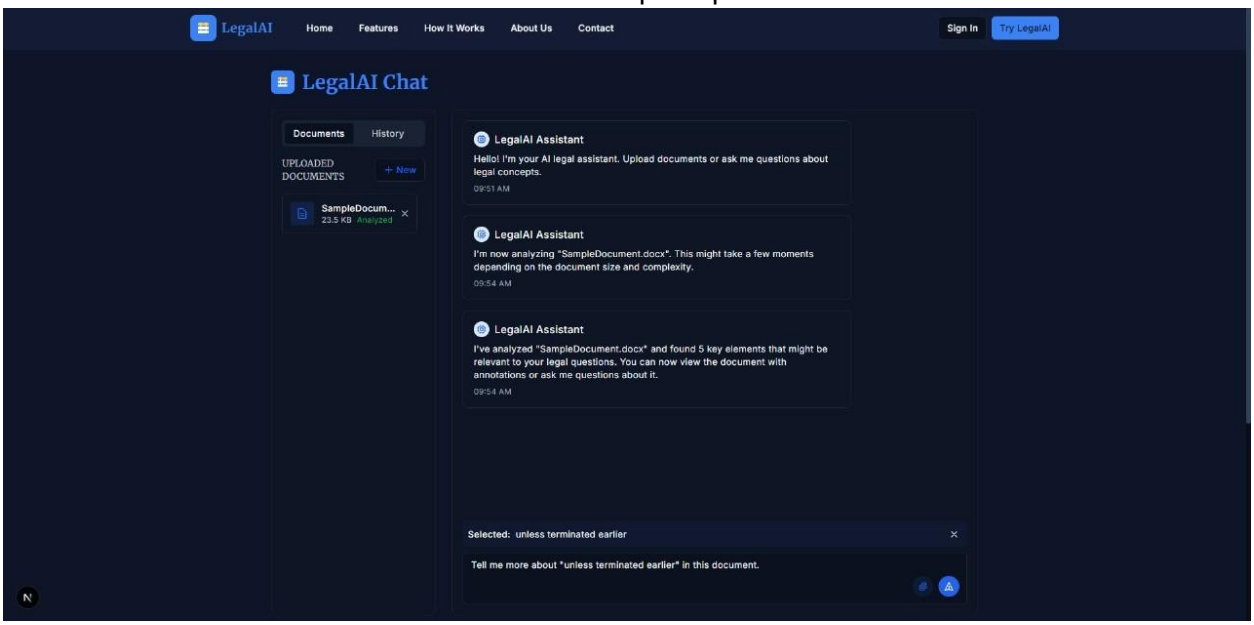
The system's final interaction layer is a GPT-4 (OpenAI) based conversational legal assistant, implemented via a Retrieval-Augmented Generation (RAG) approach.

Pipeline:

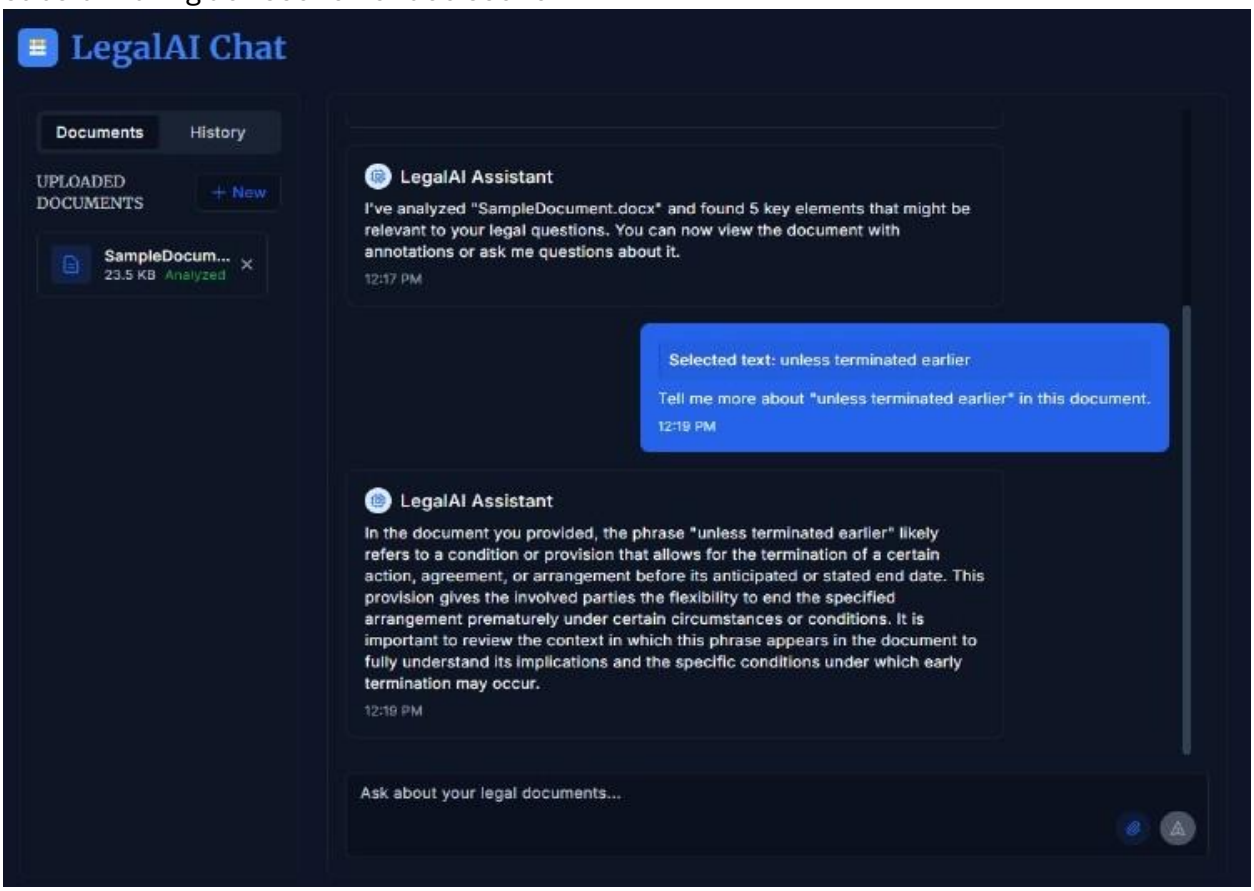
User Query → Semantic Search: The user queries a question about a document. Relevant chunks are retrieved from Pinecone using cosine similarity.



Prompt Construction: Retrieved text chunks, act references, sentence classifications, and metadata are combined into a structured GPT prompt.



GPT-4 delivers a contextually aware, grounded response with references to the substantiating act/section or doc section.



Follow-Up Support: Users can ask follow-up queries in the same context window. All interactions are document-specific but stateless.

The chatbot is the key user interface to complex legal documents, offering natural language access to highly structured, sentence-level legal information. It supports use cases such as:
- "What was the charge against the defendant?"
- "Was the Computer Misuse Act referred to?"
- "Summarize the court's rationale in plain language."

In contrast to generic chatbots, this system is based on per-document indexed facts, summaries, and legal metadata — making it reliable, traceable, and interpretable.am

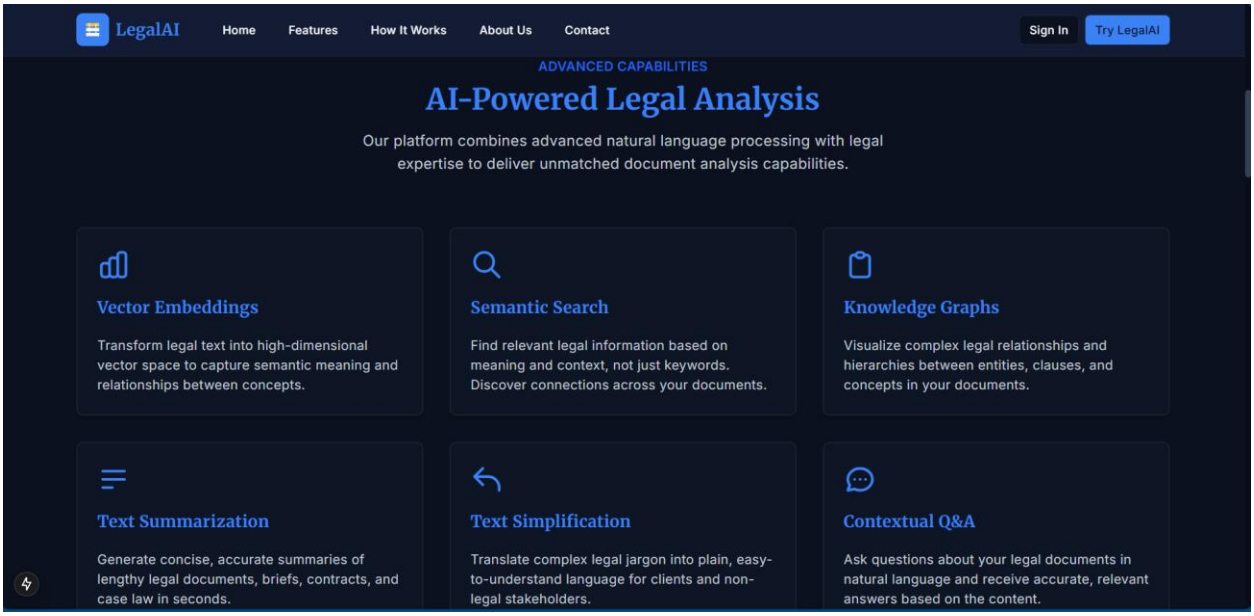
5.11 Frontend Development

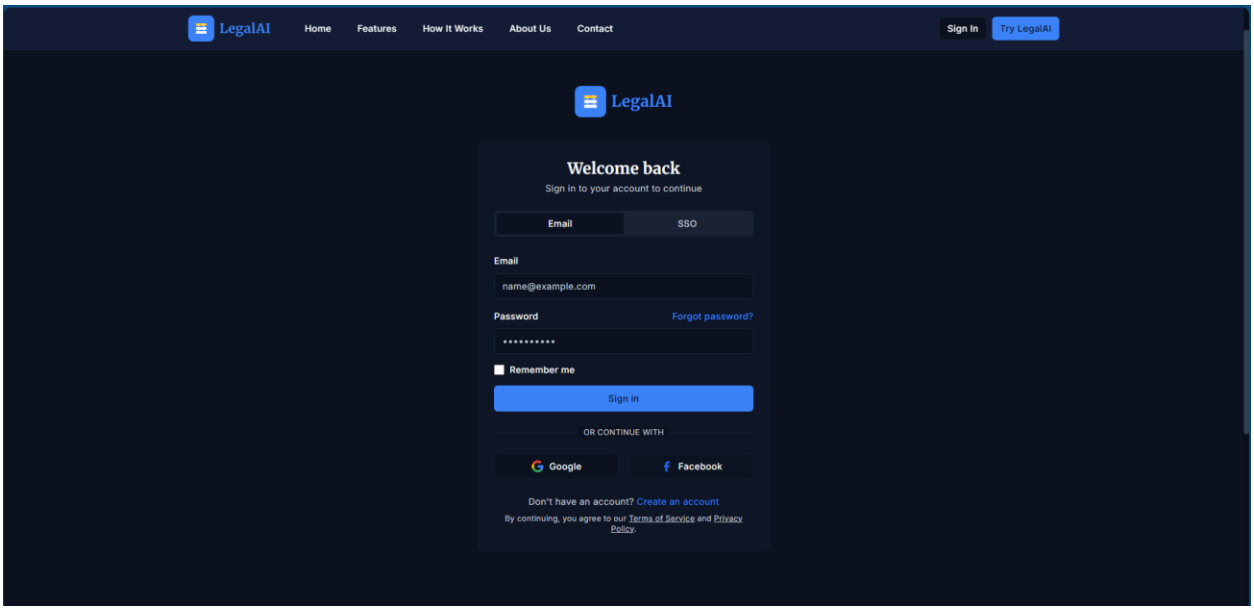
The frontend was developed using **Next.js** (built on React and TypeScript) to ensure a fast, responsive, and maintainable user interface. The goal was to build a clean, minimal interface that could surface complex legal insights without overwhelming users.

Key components of the frontend:

- **Document Viewer** – Allows users to upload legal documents and see the full text rendered with sentence-level highlighting and interactive elements.
- **Login / Signup Pages** – Implements secure user authentication using Django REST API with JWT tokens, enabling access control for document uploads, saved analyses, and personalized case history.
- **Highlighted Legal References** – Sentences with detected acts, statutes, or entities are highlighted. Hovering reveals tooltips with simplified explanations or related act details.
- **Search and Semantic Retrieval Interface** – Enables users to perform keyword or semantic queries over uploaded documents using the Pinecone index.

The frontend was styled using **Tailwind CSS** to keep the interface responsive, clean, and consistent across different screen sizes and devices. Animations and transitions (e.g., when hovering over case highlights or loading knowledge graphs) were handled using Framer Motion to enhance the user experience without bloating the interface.
This frontend setup ensures legal professionals and researchers can interact seamlessly with NLP-driven insights without needing technical expertise.





5.12 Technologies Used

5.12.1 Programming Languages and Core Technologies

Core Languages

- **Python** – Primary language for all NLP and backend tasks (spaCy, Transformers, Django).
- **JavaScript / TypeScript** – Used in the Next.js frontend for building reactive user interfaces.
- **SQL (PostgreSQL)** – Stores structured metadata about documents and cases.

5.12.1 Key Frameworks and Libraries

Natural Language Processing

- **spaCy** – Used for sentence segmentation, dependency parsing, and NER.
- **NLTK** – Supports custom legal text normalization (handling abbreviations, broken lines, archaic punctuation).
- **Hugging Face Transformers** – Framework for fine-tuning LegalBERT and RoBERTa-based models for legal domain tasks.

Sentence Embedding and Semantic Search

- **Sentence-Transformers** – Used to encode legal case chunks for semantic retrieval.
- **Pinecone** – Serverless vector database storing 1024-dimensional sentence embeddings using LLaMA/Legal-BERT encoders.

Models Utilized

- **Facebook_large_BART** – Used to perform text summarization.
- **Legal-BERT (Hugging Face Transformers)** – Domain-specific transformer model used for legal NER, sentence embeddings, and fine-tuned classification tasks.
- **RoBERTa (Fine-Tuned on LEDGAR)** – Used for clause classification within legal texts (e.g., obligations, rights, penalties).
- **LLama-2 Embedding Model** – Generates high-dimensional vector embeddings for legal documents to support semantic search.
- **spaCy Transformer (en_core_web_trf)** – Used for tokenization, dependency parsing, and entity recognition using transformer-backed pipelines.
- **AllenNLP OpenIE / Babelscape REBEL** – Integrated for zero-shot relation extraction from legal case sentences without additional training.

Backend and Storage

- **Django** – Handles backend APIs, user auth, and upload workflows.
- **Django REST Framework** – Exposes endpoints for document ingestion, search, and clause-level interaction.
- **PostgreSQL** – Main relational database for user data, document metadata, and NLP outputs.

Frontend and User Interaction

- **Next.js (React + TypeScript)** – Frontend interface for document analysis and interaction.
- **Tailwind CSS** – UI styling framework used for responsive, clean layouts.

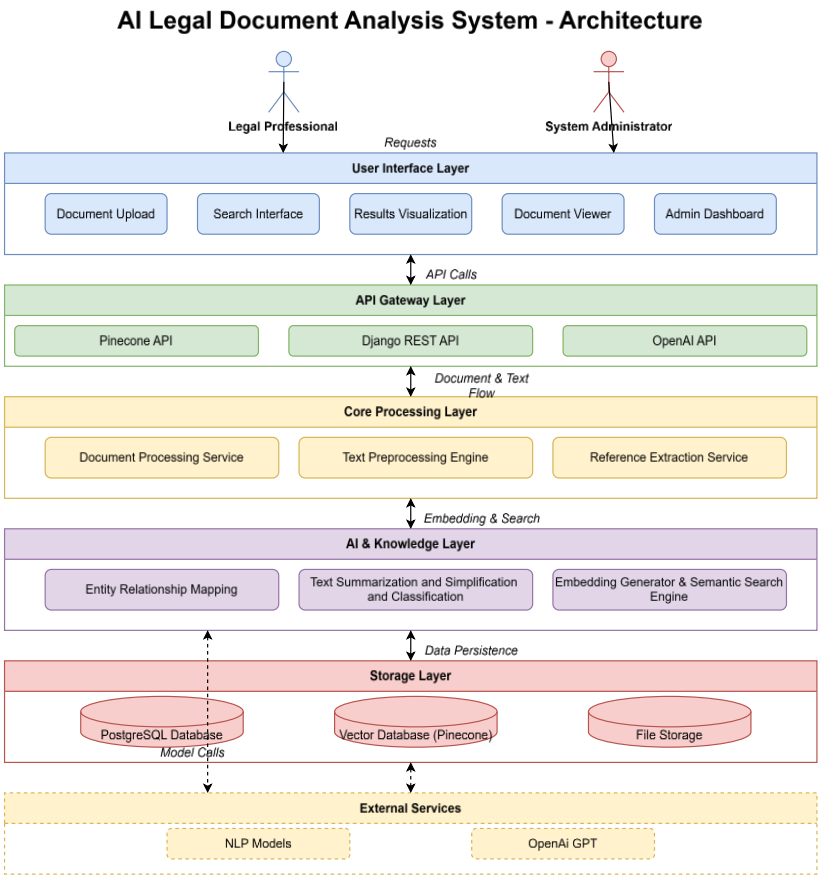
Development and Infrastructure

- **VS Code –** Used for development of the solution.
- **Google Collab –** Used for model training and testing and extracting.
- **GitHub** – Used for version control and collaborative tracking.
- **Postman** – For testing API endpoints and monitoring backend responses.

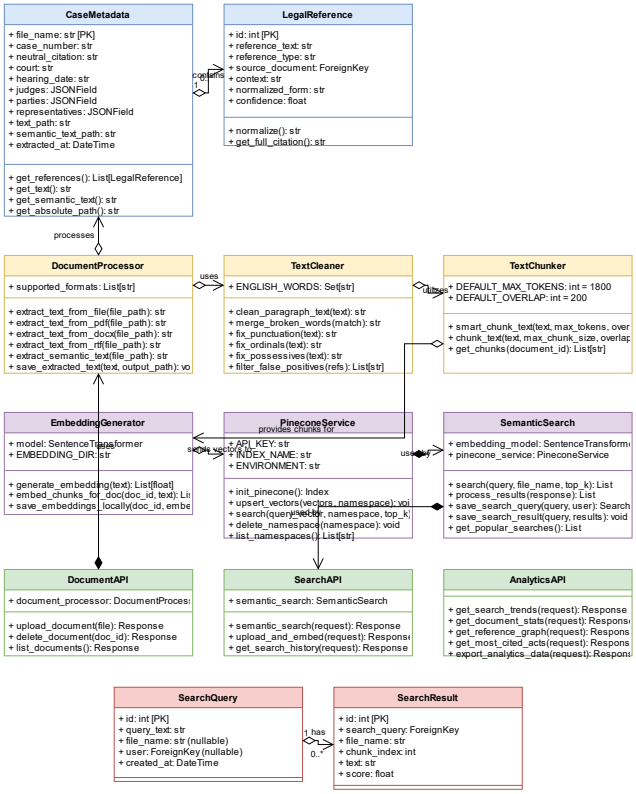
5.12 Diagrams

Use Case Diagram

The high level architectural diagram is shown below,


AI Legal Document Analysis System - Architecture
Legal Professional
System Administrator
Requests
User Interface Layer
Document Upload
Search Interface
Results Visualization
Document Viewer
Admin Dashboard
API Calls
API Gateway Layer
Pinecone API
Django REST API
OpenAI API
Document & Text Flow
Core Processing Layer
Document Processing Service
Text Preprocessing Engine
Reference Extraction Service
Embedding & Search
AI & Knowledge Layer
Entity Relationship Mapping
Text Summarization and Simplification and Classification
Embedding Generator & Semantic Search Engine
Data Persistence
Storage Layer
PostgreSQL Database
Vector Database (Pinecone)
File Storage
Model Calls
External Services
NLP Models
OpenAi GPT

AI Legal Document Analysis System - Class Diagram



Class diagram is shown above

6. Requirements

6.1 Functional Requirements

Document Management

- **Support for Multiple Formats**: For analysis, users ought to be allowed to upload several document formats.
- **Automated Content Structuring**: Upon upload, documents should undergo preprocessing to organize the text and get rid of irrelevant content, ensuring a clean dataset for analysis.

Model Fine-Tuning and Performance Enhancement

- **Domain-Specific Model Adaptation**: Existing models should be fine-tuned using specialized datasets.

Natural Language Processing (NLP)

- **Simplification and Summarization**: The system should employ advanced NLP techniques to simplify complex legal jargon and provide concise summaries, making the content more accessible.
- **Contextual Search Capabilities**: Implement semantic search functionalities to enable context-aware retrieval, surpassing traditional keyword-based search limitations.

User Interface and Interaction

- **Comprehensive Display**: The interface should present a summary of the document, simplified terms, the knowledge graph, and relevant sections from semantic searches, all in a cohesive view.
- **Query Refinement Options**: Provide users with capabilities to refine their queries or searches, allowing for follow up questions.

Additional Considerations

- **User Authentication and Access Control**: Implement robust authentication mechanisms and role-based access controls to protect sensitive legal data.

6.2 Non-Functional Requirements

- **Performance**: Ensure the system processes and analyzes documents within an acceptable timeframe, maintaining minimal latency to support usability.
- **Scalability**: The system should handle multiple users simultaneously without performance degradation and manage large documents and numerous queries efficiently.
- **Usability**: Design an intuitive user interface that facilitates easy navigation, interaction with knowledge graphs, and viewing of summaries.
- **Reliability and Accuracy**: Maintain high accuracy and consistent performance, incorporating robust error handling and continuous monitoring mechanisms.
- **Security**: Uphold confidentiality through strong authentication and authorization protocols, secure data storage and transmission via encryption, and compliance with relevant data protection regulations.
- **Maintainability**: Ensure the system is designed for easy maintenance, allowing for updates and modifications as needed.
- **Compliance**: Adhere to all applicable legal and regulatory standards, such as the UK General Data Protection Regulation (GDPR), ensuring lawful processing and protection of personal data.

7. End – Project Report

The project sought to develop an AI-driven legal document analysis system for the UK legal market with a particular focus on computer and cybercrime case law. It integrates natural language processing (NLP), semantic vector search, and knowledge graph technologies to enable deep comprehension and intelligent interaction with advanced legal documents. The final system allows users to upload legal documents, extract structured metadata, find crucial references and acts, and perform semantic retrieval of case content based on embeddings in Pinecone.

Throughout the development process, a full-stack pipeline was set up, consisting of a Django backend with PostgreSQL and Neo4j integration, a React-based frontend with Next.js and TailwindCSS, and transformer-based NLP processing with Legal-BERT and other fine-tuned models.

7.1 Project Objectives vs Achievements

Objective	Status	Commentary
Build a legal document parser capable of extracting metadata and legal references.	Mostly achieved	Regex and NLP were used effectively; however, due to the inconsistent formatting of court documents, 100% extraction accuracy was not possible.
Implement sentence-level act and statute extraction.	Achieved	Integrated regex + spaCy pipeline for robust sentence segmentation and law tagging.
Store and manage extracted data using PostgreSQL and Neo4j.	Fully achieved	Structured storage with foreign key mappings, supporting scalable data architecture.
Enable semantic search using embeddings and vector databases.	Achieved	LegalBERT embeddings are stored and queried via Pinecone; semantic results are linked back to sentence-level context.
Develop an interactive frontend for document upload and analysis.	Delivered	Responsive UI with upload viewer, sentence-level hover interactions, and highlighting.
Train a classifier for clause or reference type detection using fine-tuned models.	Partially achieved	Basic sentence classification and NER were integrated, but deeper clause classification faced limitations due to labeled data scarcity.

7.2 Critical Evaluation

This project was successful in accomplishing most of its core objectives, especially in system functionality, document interaction, and legal reference processing. There were, however, several key challenges:

Data Annotation: Legal documents, especially case law, are difficult to annotate due to inconsistent formatting, nested references, and antiquated language. Creating labeled datasets for model training was manually time-consuming and labor-intensive, and the unavailability of datasets hindered some ML components.

Regex Limitations: Initially, reference extraction from the law employed extensive use of regex that turned brittle in the face of non-standard formating, PDF OCR errors, or nested clause citations (e.g., "Section 1(1)(a) of the 1990 Act").

Package Incompatibility: Exaggerated amounts of time were wasted repairing stale packages, stale APIs (especially with NLP libraries), and broken dependencies within the Python environment (e.g., issues around spaCy transformer pipeline updates, Neo4j drivers).

Model Training Obstacles: Legal-specific training of transformers (LegalBERT, RoBERTa) required more compute than expected. Colab GPU limits and unavailability of large pre-annotated corpora resulted in fine-tuning being partially successful.

Knowledge Graph Complexity: Mapping extracted entities to Neo4j nodes and with search was more complicated than expected due to similarity among acts, statutes, and collateral phrases.

Despite these, the system is tangible in its contribution and pragmatic for trial in the real world. It is a working model demonstrating the real-world viability of employing AI technology in legal tech, in particular, making legal documents more interactive and readable.

7.3 Changes Made During Project

Switch from Rule-based to ML+Rule Hybrid: Initial regex-only extraction was replaced by a hybrid model using spaCy's transformer pipeline plus fallback rules. This drastically improved reference recognition and enabled more robust sentence extraction.

Model Selection Changes: Blackstone was considered for legal NER, but was replaced with LegalBERT + spaCy due to model quality and compatibility.

Vector DB Choice: The final system uses Pinecone instead of Weaviate, due to simpler integration and faster performance with the llama-text-embed-v2 model.

Frontend Design Evolution: UI design was initially basic, but evolved with TailwindCSS and hover-based interactivity to suit the document exploration use case.

7.4 Realization of Business Objectives

The core business/research value of the system is to:

- Help researchers, students, and professionals interact with long, dense legal documents more intuitively.
- Non professionals who want to gain insight regarding a legal document they might have.
- Enable search and summarization over legal content, reducing the effort required to find relevant cases and clauses.
- Provide an open framework that can be adapted to different legal domains or extended with additional AI capabilities.

The system also lays the foundation for future academic research or commercial extensions, such as chat-based legal assistants or legal citation recommendation engines.

8. Project Post - Mortem

Were the Right Objectives Chosen?
Yes — the objectives addressed real pain points in navigating complex case law. However, more focused objectives around specific user scenarios (e.g., lawyer researching precedent, student preparing a case summary) might have helped narrow development focus earlier.

Was the Product Properly Specified?
Partially. The high-level goals were clear — build an interactive AI tool for legal documents — but some of the subcomponents (like clause classification or relationship extraction) were underspecified. More upfront exploration of feasible ML components would have helped.

Relationship with the Client/Stakeholders
While this was an academic project, the system was built with legal professionals and students in mind. Several feedback sessions were held with law students and legal researchers, which led to interface and language simplification improvements. Their feedback particularly shaped the frontend and the glossary/simplification features.

Was Agile the Right Process?
Absolutely. Without Agile, the frequent pivots (e.g., switching NER libraries, adjusting vector DBs, frontend redesign) would've been chaotic. Weekly goals and continuous feedback loops enabled steady progress despite unexpected blockers.

Were the Technologies Chosen Correctly?
Mostly. Python, Django, spaCy, and LegalBERT were solid choices. Neo4j introduced some learning curve, but was effective for representing legal relationships. Pinecone proved better suited than Weaviate for fast, lightweight semantic search. Some tools like Google Drive API or Hugging Face inference endpoints introduced rate limits, which were frustrating during heavy usage/testing.

Performance and Challenges

- **Your Own Performance**: The project demanded deep learning across NLP, backend, and frontend — a serious full-stack engineering effort. Task-switching between annotation, debugging, and model evaluation was mentally draining. That said, significant progress was made in research, coding, and integrating unfamiliar systems, which reflects strong learning agility.
- **Labelled Data Shortage**: This was the biggest bottleneck. Most legal tasks don't have pre-labeled corpora, so creating training data for classification or relation extraction was a manual grind, limiting scope.

- **Toolchain Pain**: Installing and maintaining transformer-based models with GPU dependencies across environments was a constant source of errors and environment rebuilding.
- **Incomplete Metadata Extraction**: Despite regex and rule-based strategies, some legal references (especially nested or implied ones) were missed. Full comprehension requires context beyond what regex or sentence-based NLP can offer.

Lessons Learned

- **Legal text processing is hard.** Even with transformer models, edge cases and document inconsistencies create noise and ambiguity.
- **Don't underestimate frontend work.** Making AI outputs usable, readable, and trustworthy requires careful UI decisions.
- **Start with the data.** Before building pipelines or training models, understand what your data *actually* looks like and what labels you *can* realistically produce.
- **Hybrid > Pure ML or Rules.** A mix of NLP and regex delivered better results than either in isolation.
- **Be ready to pivot.** Many initial ideas (like using Blackstone, or fine-tuning full models) didn't pan out. Accepting and adapting quickly saved the project from getting stuck.

9. Conclusion

This project demonstrates that while AI for the legal environment is promising, it is not a plug-and-play situation. It demands careful engineering, annotation, iterative design, and hybrid methods to deal with legal language flexibility and complexity.

The final system offers an executable and extensible platform for parsing documents, referencing acts, and interactive querying — functionality of real value to law students, legal researchers, and legal tech practitioners. It is a balance of NLP innovation and practical application, embracing transformers, semantic embeddings, knowledge graphs, and a modern web interface.

The challenges faced — notably data labeling, integration of ML models, and NLP constraints — have provided valuable lessons for subsequent projects. While there is room for enhancement in clause categorization as well as deep relation extraction, the system is well positioned for future expansion, including GPT-based agent integration, higher summarization, and conversational interfaces.

Finally, the project exhibits not only technical capability, but also the ability to cope with uncertainty, complexity, and ambiguity — all of which are inherent in law and AI.

10. References

Chalkidis, I. *et al.* (2020) 'LEGAL-BERT: The Muppets straight out of Law School'. Available at: http://arxiv.org/abs/2010.02559.

Dragoni, M. *et al.* (2016) *Combining NLP Approaches for Rule Extraction from Legal Documents Combining NLP Approaches for Rule Extraction from Legal Documents. 1st Workshop on MIning and REasoning with Legal texts Combining NLP Approaches for Rule Extraction from Legal Documents*. Available at: https://wordnet.princeton.edu/.

Imogen, P.V., Sreenidhi, J. and Nivedha, V. (2024) 'AI-Powered Legal Documentation Assistant', *Journal of Artificial Intelligence and Capsule Networks*, 6(2), pp. 210–226. Available at: https://doi.org/10.36548/jaicn.2024.2.007.

Merchant, K. and Pande, Y. (2018) *NLP Based Latent Semantic Analysis for Legal Text Summarization*. IEEE.

Sabrina Univ-Prof Axel P, A.K. (2021) *Knowledge Graphs for Analyzing and Searching Legal Data*.

Sachidananda, V., Kessler, J.S. and Lai, Y. (2021) 'Efficient Domain Adaptation of Language Models via Adaptive Tokenization'. Available at: http://arxiv.org/abs/2109.07460.

Vayadande, K. *et al.* (2024) 'AI-Powered Legal Documentation Assistant', in *Proceedings - 2024 4th International Conference on Pervasive Computing and Social Networking, ICPCSN 2024*. Institute of Electrical and Electronics Engineers Inc., pp. 84–91. Available at: https://doi.org/10.1109/ICPCSN62568.2024.00022.

Yang, W. *et al.* (2019) 'Legal Judgment Prediction via Multi-Perspective Bi-Feedback Network'. Available at: https://doi.org/10.24963/ijcai.2019/567.

Zakir, M.H. *et al.* (2024) 'Artificial Intelligence and Machine Learning in Legal Research: A Comprehensive Analysis', *Qlantic Journal of Social Sciences*, 5(1), pp. 307–317. Available at: https://doi.org/10.55737/qjss.203679344.
Zheng, J. *et al.* (2024) 'Fine-tuning Large Language Models for Domain-specific Machine Translation'. Available at: http://arxiv.org/abs/2402.15061.
Zhong, H. *et al.* (2020) 'How Does NLP Benefit Legal System: A Summary of Legal Artificial Intelligence'. Available at: http://arxiv.org/abs/2004.12158.

11. Appendices

11.1 User guide

AI Legal Document Analyzer

An advanced web application that helps users upload legal documents, analyze them with AI, and interact with them through natural language chat.

Features

- Upload and analyze legal documents (PDF, DOC, DOCX, TXT)
- Automatically extract and highlight key elements:
  - Legal terms
  - Dates
  - Parties involved
  - Obligations
  - Conditions
- Chat with AI about document content
- Select specific clauses to ask targeted questions
- Interactive document viewer with annotations

Project Structure

The project consists of two main components:

- Backend (Django/Python): Handles document processing, analysis, and chat functionality
- Frontend (Next.js/React): Provides the user interface and document viewer

Setup Instructions

Prerequisites

- Node.js (v16+)
- Python (v3.11+)
- PostgreSQL

Backend Setup

1. Create a PostgreSQL database named `legal_doc_analyzer`

2. Navigate to the backend directory:
   ```
   cd backend
   ```

3. Create a virtual environment:
   ```
   python -m venv venv
   ```

4. Activate the virtual environment:
   - Windows: `venv\Scripts\activate`
   - macOS/Linux: `source venv/bin/activate`

5. Install dependencies:
   ```
   pip install -r requirements.txt
   ```

6. Configure environment variables:
   - Copy `.env.example` to `.env`
   - Add your OpenAI API key to the `.env` file

7. Apply migrations:
   ```
   cd backend_core
   ```

```
python manage.py migrate
```

8. Run the development server:
```
python manage.py runserver
```

Frontend Setup

1. Navigate to the frontend directory:
```
cd ai-legal-document-analysis-frontend
```

2. Install dependencies:
```
npm install
```

3. Configure environment variables:
   - Copy `.env.local.example` to `.env.local`
   - Ensure the API URL points to your Django backend

4. Run the development server:
```
npm run dev
```

5. Open your browser and navigate to `http://localhost:3000`

Usage

1. Upload a legal document using the sidebar
2. Wait for AI analysis to complete
3. View the document with annotations
4. Select annotations to ask specific questions
5. Chat with the AI about the document content

Technologies Used

- Frontend:
  - Next.js

- React
- TypeScript
- Tailwind CSS

- Backend:
  - Django
  - Django REST Framework
  - PostgreSQL
  - Pinecone
  - OpenAI API

11.2 PID

PUSL3190 - Computing Individual Project

Project Initiation Document (PID)

AI for Legal Document Analysis

Supervisor: Dr. Mohamed Shafraz

Name: Weerasinghe Dissanayakalage Methsara Nisaga Dissanayaka
Plymouth Index Number: 10899302
Degree Program: BSc. (Hons) in Data Science

Table of Contents

1. Introduction

The legal sector of a nation serves as the backbone of maintaining societal order, providing the framework for governance, rights, and responsibilities. In order to uphold justice, guarantee compliance and perform these very important tasks, legal institutions rely on a vast array of documents, including statutes, case law, contracts, and regulations. These documents are constructed with detailed language and specific jargon to the Legal field to capture complex legal ideas, therefore requiring interpretation by skilled professionals (Lawyers, Judges and Legal Scholars). However, in the digital era, traditional approaches to legal document analysis are increasingly inadequate due to the sheer volume and complexity of legal writing. This has created a growing demand for precise, scalable technologies capable of large-scale legal text analysis and interpretation.

Legal institutions are now facing unprecedented challenges in processing and analyzing the vast amounts of legal documentation. Laws, legislations, case verdicts, and arguments are not only intricate but are also subject to constant change, driven by evolving societal norms, digital advancements, and global connectivity. Additionally, legal texts are often ambiguous, requiring significant effort to interpret. Ambiguities in language can lead to inconsistencies in understanding and application, further complicating the process for legal professionals. These challenges are amplified when working within specialized legal

domains such as intellectual property law, digital privacy, or computer crimes, where terminology is technical, dynamic, and often unfamiliar.

People who work in the legal field such as judges, attorneys and analysts must review tons of documents by hand to retrieve relevant precedents, decipher complex clauses, and apply legal principles to particular cases accordingly. This manual process is labor-intensive, time-consuming, and prone to human error, especially in high-stakes scenarios. Professionals must also contend with the pressure of responding within tight timeframes, where delays in retrieving relevant legal information can have serious repercussions. Furthermore, the inefficiencies of manual analysis often lead to missed precedents or misinterpretations, particularly when outdated legal frameworks are applied to contemporary challenges.

Recognizing these limitations, this project aims to address the core issues plaguing legal document analysis, including:

- **Complexity and Ambiguity**: Unclear language in legal documents can lead to multiple interpretations, necessitating further clarification.

- **Data Volume and Time Constraints**: The exponential growth of legal texts, combined with time-sensitive demands, exacerbates inefficiencies in locating critical information.

- **Domain-Specific Challenges**: Specialized legal domains like computer crimes require targeted tools to manage their unique terminology and complexities.

- **Inefficiencies in Manual Analysis**: Traditional methods lack scalability, increasing the risk of errors and delays.

To tackle these challenges, the project leverages artificial intelligence (AI) and advanced natural language processing (NLP) techniques. The proposed system will focus on document summarization, legal term simplification, semantic search, and entity relationship mapping, tailored specifically to the domain of computer crimes. By narrowing the scope to cybercrime and referencing cases under the **Computer Crime Act No. 24 of 2007** (Certified on 09th July, 2007 and Published as a Supplement to Part II of the Gazette of the Democratic Socialist Republic of Sri Lanka of July 13, 2007), this tool provides a more focused solution that enhances accuracy and relevance.

This targeted approach allows for precise fine-tuning of NLP models, enabling the system to generate summaries, simplify legal jargon, and visualize knowledge graphs. By training on a specialized dataset of computer crime cases, the system can deliver deeper insights into this subset of law, helping legal professionals save time and improve decision-making. Moreover, this project not only aims to address current inefficiencies but

also lays the foundation for expanding AI-driven solutions to other legal domains in the future, bridging the gap between legal expertise and technological innovation.

2. Business Case

2.1 Business Need

The legal industry has several difficulties as a result of the growing incidence of cybercrime, particularly when it comes to evaluating and interpreting case files. Therefore, legal experts have to invest a lot of time in manually going through dense documents, cross referencing statutes, and pinpointing vital case elements. Inefficiency in this regard is compounded in the case of cybercrime cases due to the technical nature of the subject matter and the lack of specialized tools to handle such documentation.

Most of the tools available in the legal domain are still very generic, based on keyword searches with little contextual understanding, even with advancements in AI and NLP. These tools, even while being comprehensive, such as Westlaw and LexisNexis, often fail in providing domain-specific insights, especially when it comes to the cases on cybercrime. So, legal practitioners are forced to manually determine what connections there are between instances and statutes or waste time sorting through pointless results.

Not only that, the benefit of a system such as this for non-legal experts such as the normal public is a huge thing to notice as well. As they would be able to efficiently navigate and interpret complex legal documents, saving time and reducing the necessity of extensive legal consultations which can be both inaccurate (depending on the consultation) and expensive.

This project addresses these gaps by creating an AI-driven tool tailored to cybercrime case analysis. By automating tasks such as document summarization, semantic search, and entity relationship mapping, the tool aims to improve efficiency, reduce human error, and support faster decision-making in cybercrime litigation. Evidence from AI adoption in other sectors highlights its potential to increase efficiency by up to 40%, making a compelling case for its application in law.

2.2 Business Objectives

- Efficiency Improvement:
  Reduce the time required to review cybercrime-related legal documents by 50% at least through automated summarization and advanced search functionalities.

- Enhanced Information Retrieval:
  Implement semantic search to ensure at least 90% accuracy in retrieving contextually relevant sections of documents, reducing reliance on keyword-based systems.

- Simplified Legal Language:
  Provide simplified explanations for at least 85% of complex legal terms in the analyzed documents, improving accessibility for non-specialists.

- Interactive Data Visualization:
  Enable users to visualize relationships between cases, statutes, and entities through a knowledge graph interface, increasing contextual understanding by about 60%.

- Scalable Solution Development:
  Design a modular system architecture capable of scaling to additional legal domains, supporting future expansions beyond cybercrime law.

3. Project Objectives

7. To develop a Legal Document Processing Pipeline. (**Creating**)
   - Create an efficient pipeline to handle various document formats, structure them, and prepare them for analysis, also including text extraction, metadata processing and document normalization.
   - Ensure everything related is handled with at least 95% accuracy.

8. To integrate enhanced NLP methods for Advanced Summarization and Simplification of Legal Text. (**Integrating**)
   - Employ transformer-based NLP models (Ex: Legal - BERT) fine-tuned on legal data to extract key points, provide summaries and simplify complex legal writing, streamlining document review.
   - Target a summary length reduction of up to 70% while retaining critical legal information.

- Implement a glossary with dynamic linking to the processed document for easy reference

9. To implement Semantic Search with Vector embeddings. (**Applying**)
   - Enable retrieval of information based on meaning rather than exact matches using vector embeddings, significantly enhancing user navigation through complex documents.
   - Achieve a precision and recall rate of at least 85% for user queries.

10. To visualize legal document relationships through Knowledge Graphs. (if feasible)
    - Build a knowledge graph model that maps out relationships and dependencies within the text. This is crucial for identifying primary entities, connections and providing an interactive layer of insight into legal networks.
    - Visualize connections and relationships with clear labels and metadata for at least 80% of entities in the dataset.

11. To customize the AI Tool for Computer Crime case law using domain-specific Model Training and Evaluate Tool Performance. (**Analyzing & Evaluating**)
    - Develop an AI tool that is fine-tuned to address the specific requirements and nuances of a particular domain (cybercrime cases) and carry out performance evaluations on a dataset specific to the selected domain. Thus, creating a precise and functional AI solution for a targeted legal area.

12. To design an Intuitive & Interactive User interface. (**Developing**)
    - Craft an intuitive User interface for effortless navigation and engagement with complex legal data.

4. Literature Review

One doesn't need to be a smart individual to understand that the Legal Sector is the framework that keeps everything in its rightful place, from the highest echelons of authority to the most fundamental societal functions. This structured hierarchy ensures that order, justice, and rights (not to mention that they are also defined by the law) are preserved, safeguarding society from chaos and fostering a foundation upon which all individuals and institutions can rely. Carefully crafted Documents by legal experts which contain legal statutes, case law, contracts, and regulations uphold this structure, ensuring every detail is meticulously accounted for. But as these documents try to take everything into account, it means that these documents become extremely complex and often quite lengthy. And as the world and human activities continue to expand, the volume of legal data will also continue to increase, thus making these documents, which contain relevant legal sentences, frequently subjected to change. Therefore, thorough Legal Document Analysis is pivotal for the application and evolution of not just the legal sector but for any human endeavor.

However, current methods for Legal Document Analysis which are generally man-powered faces a lot of challenges in this modern era. The reliance on these human-driven document processing introduces issues such as time constraints, inefficiencies in information retrieval, susceptibility to errors, and vulnerable to various other problems. Apart from the need of human labor (experts in relevant fields), even the computer based legal document management tools has many shortcomings, as most of them primarily focus on document storage and retrieval through keyword-based search, which fails to capture the nuanced relationship and context-specific language often present in legal texts. This is particularly true in fields like cybercrime, where cases and statues are frequently intricately linked.

With the introduction of Artificial Intelligence (AI) in this digital era to many sectors, new possibilities have emerged in automating and enhancing various tasks in the Legal Sector as well, specifically in Legal Document Analysis. AI technologies such as Natural Language Processing (NLP), Deep Learning and Machine Learning have already been applied to the Legal Sector in order to transform raw legal text into structured, searchable and interpretable data (Zhong *et al.*, 2020). Pretrained language models such as BERT and RoBERTa are widely used for NLP tasks among a variety of industries and have demonstrated potential for legal applications when fine-tuned and optimized on domain-specific data. This gave birth to models such as Sci-BERT (Trained on biomedical and computer science literature), Fin-BERT (Focused on financial services), and Bio-BERT (Specialized in biomedical literature) etc., and among these specialized models, LEGAL-BERT stands out as particularly relevant this project. Although it is quite self-explanatory, this model is designed specifically for the Legal Domain, offering tailored solutions such as,

- Legal Document Classification
- Named Entity Recognition
- Legal Judgement Prediction
- Legal Statute Identification
- Semantic Segmentation
- Court Judgement Prediction

...that aligns perfectly with the direction of this project (Chalkidis *et al.*, 2020). Additionally, techniques like adaptive tokenization have been developed to modify pretrained models' tokenizers to better handle specialized vocabulary in legal documents. Research by (Sachidananda, Kessler and Lai, 2021) highlights that adapting tokenizer to legal terminology significantly boosts model performance on specialized tasks without the costs of training a new model from scratch.

AI is being utilized more and more in the legal industry for tasks ranging from document classification, rule extraction to even predicting case verdicts (Yang *et al.*,

2019). There are already existing applications that can perform document analysis. Research from (Zakir *et al.*, 2024) shows how much AI has advanced the processing of legal documents. Transformation models like Legal-BERT are proof that AI can have significant impact on the legal sector. Another major component that is believed to be not utilized to its full potential yet is knowledge graphs. Named Entity Recognition (NER) along with knowledge graphing offer a promising approach to organizing legal data, linking statutes, cases, and other legal entities into a structured network that facilitates query-based exploration. Erwin Flitz's dissertation on knowledge graphs (Sabrina Univ-Prof Axel P, 2021) underlines the benefits of linking legal data through standardized identifiers (like ELI and ECLI) and ontologies, which enables complex queries and cross-references across legal systems. These methods can help legal professionals and non-experts by visualizing relationships between legal entities and providing a comprehensive view of case law and statutory connections.

Numerous research offers insights into the application of AI in the legal sector, showcasing techniques for semantic searching using vector embeddings that goes beyond the limitations of keyword-based searching, vector databases, extracting and summarizing of legal texts etc. For instance, (Dragoni *et al.*, 2016) discuss methods of rule extraction from legal documents using NLP, combining syntactic parsing with semantic analysis to identify regulatory rules. Meanwhile (Merchant and Pande, 2018) describes the usage of latent semantic analysis for legal text summarization. Such approaches demonstrate how multi-step NLP techniques can parse dense legal terminology into comprehensible terms, making legal reasoning more accessible through AI. Similarly, adaptive domain training techniques such as those proposed by (Sachidananda, Kessler and Lai, 2021) and (Zheng *et al.*, 2024), show that fine-tuning models with domain-specific tokens or language data can significantly enhance NLP results for specific domains inside the legal sector itself.

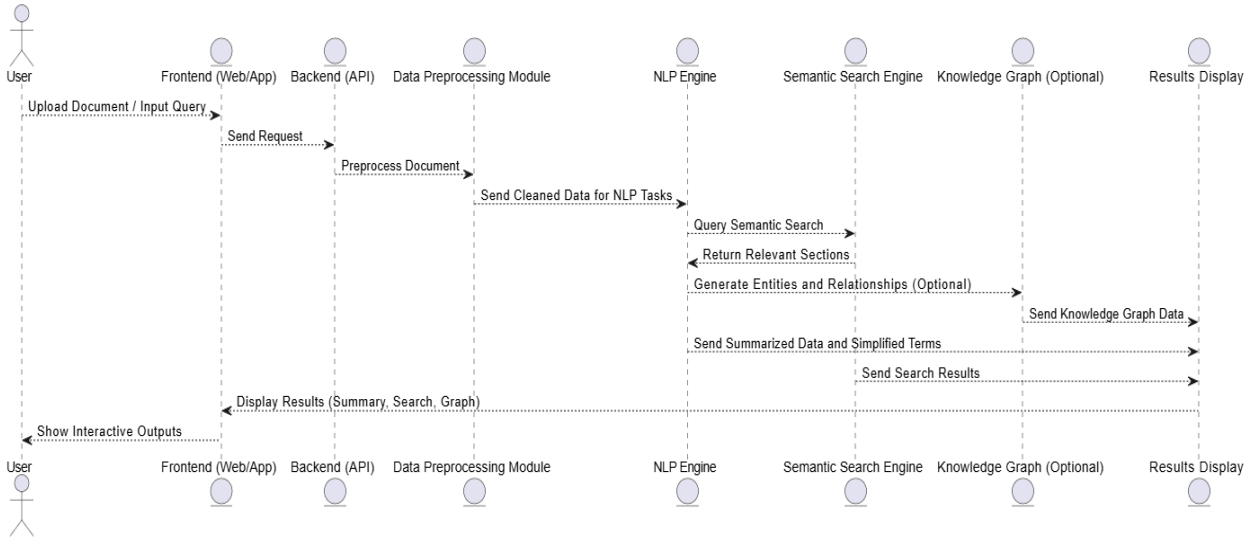
Reflecting on these findings, there is a clear **Research Gap** and Need for systems that goes beyond simple document retrieval to incorporate contextual understanding, semantic search, and entity graphing, particularly in a specific domain as we found out the benefits in using fine-tuned models for a specific domain. Current solutions primarily cater to broad legal applications, offering generalized search functions, summarization and basic information retrieval, which fall short in addressing the specific nuances of various domains within the legal sector (because law covers every aspect in the world, even the simplest ones an individual can think of), particularly when it comes to highly specialized / specific fields such as cybercrimes. For example, complex domain-specific language can be misinterpreted by generic NLP models, even the ones that are fine-tuned for the Legal Sector itself, making precise entity recognition, relationship mapping, and contextual summarization challenging (Zhong *et al.*, 2020). Furthermore, only a small number of current tools completely incorporate sophisticated capabilities such as semantic search, vector-based embeddings, and knowledge graph-based visualization tailored to legal contexts, leaving a clear gap for tools that integrate these capabilities into a unified, domain-specific platform (cybercrimes). Thus, the proposed project addresses these gaps by building a domain-specific AI tool focused on cybercrime, which will,

1) Utilize fine-tuned NLP models trained on cybercrime case data, enhancing the accuracy of summarization and term interpretation.
2) Incorporate semantic search powered by vector embeddings, improving the tool's relevance and retrieval precision.
3) Use knowledge graphs to map and visualize the relationship between cases, statutes, and involved entities specific to cybercrime.

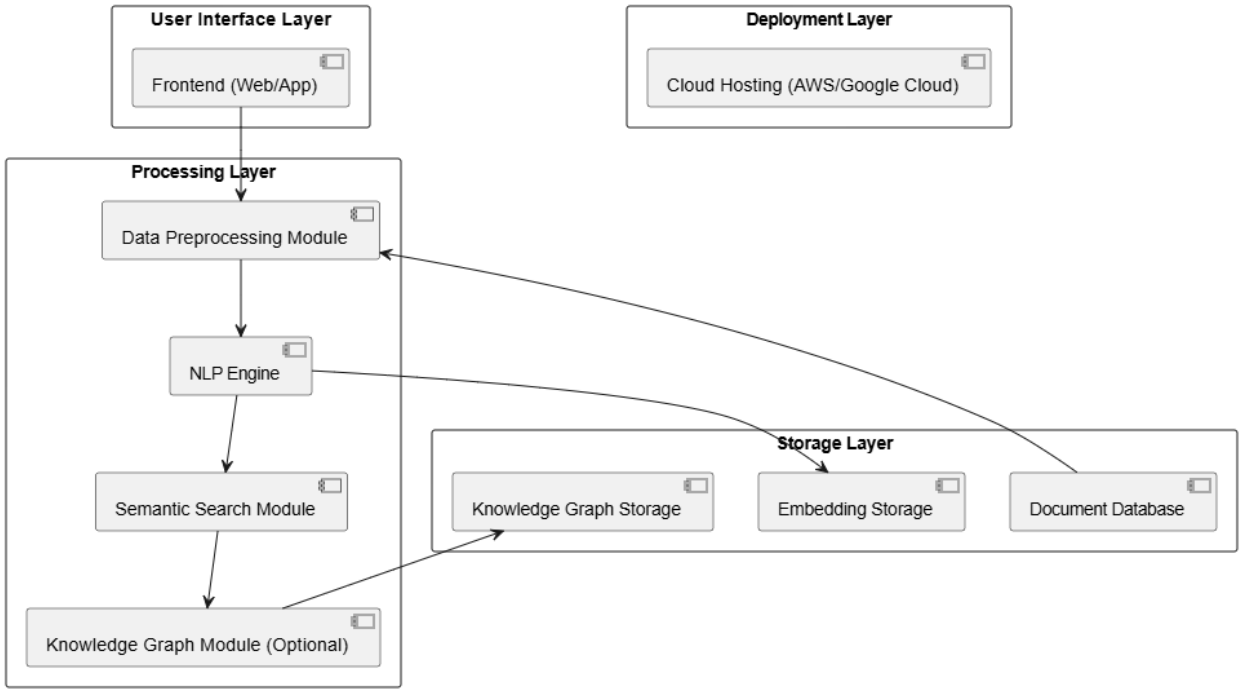
This combination of domain-specialized NLP and advanced visualization will offer a more nuanced, contextual approach than current general-purpose legal AI tools, meeting an unaddressed gap within the legal AI landscape. This outcome will also provide a replicable framework adaptable to other areas within the legal sector, ultimately contributing to the evolution of legal towards more intuitive and precise AI-Assisted legal research. (Imogen, Sreenidhi and Nivedha, 2024)(Vayadande *et al.*, 2024)

5. Diagrams

5.1 Conceptual Diagram



5.2 Layered Architecture Diagram



Divides the system into layers (Ex- User Interface, Processing, Storage, Deployment) with components and data flow within each layer, this diagram shows the top-down view of the architecture.

5.3 Workflow Diagram

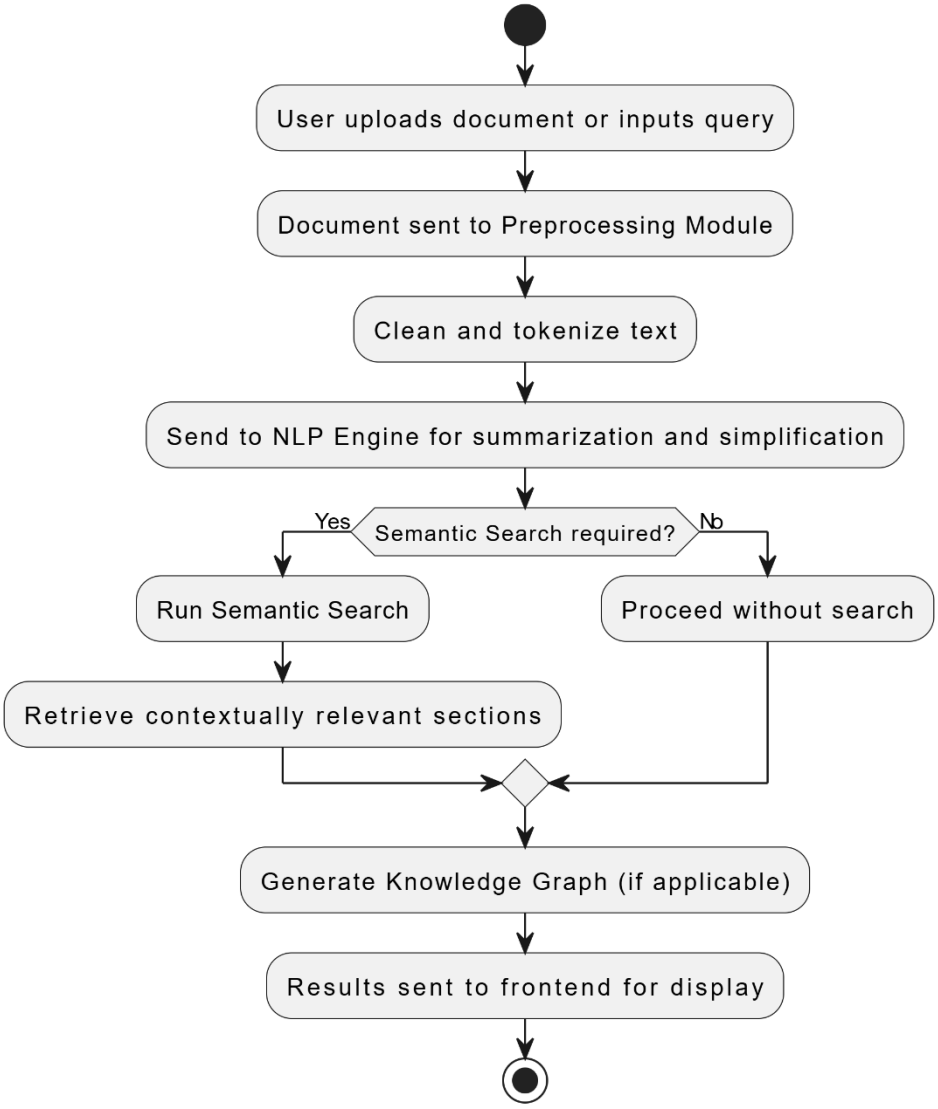
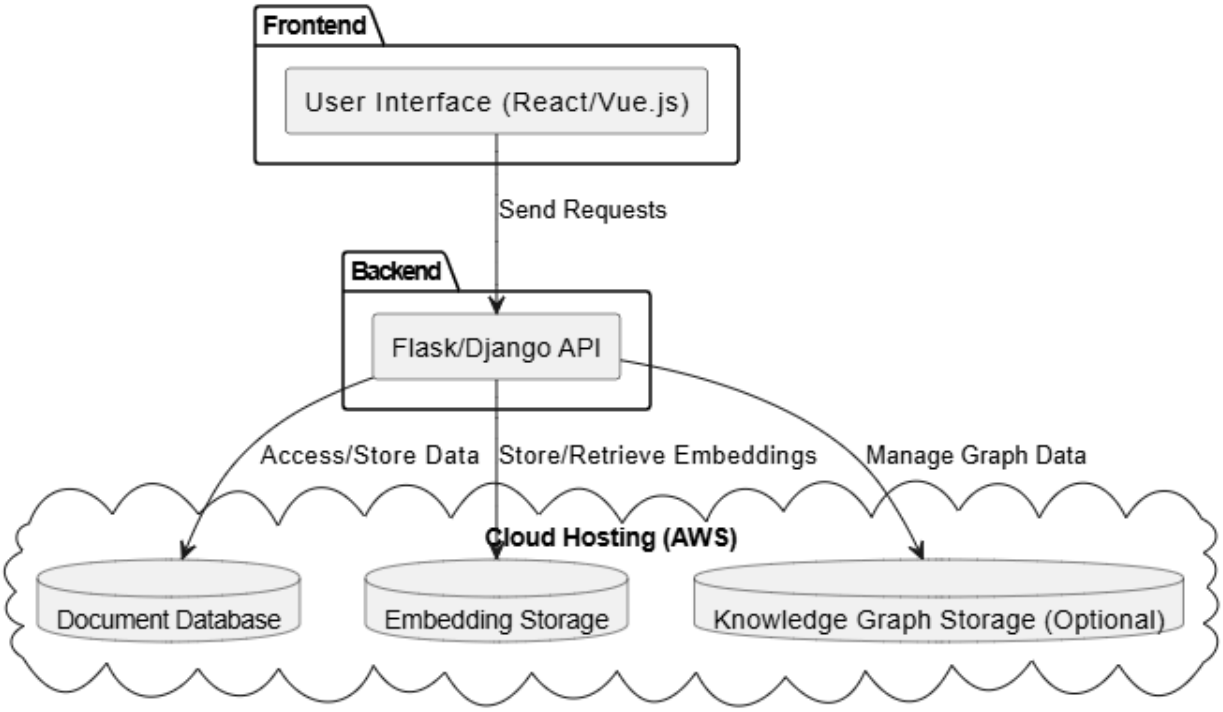


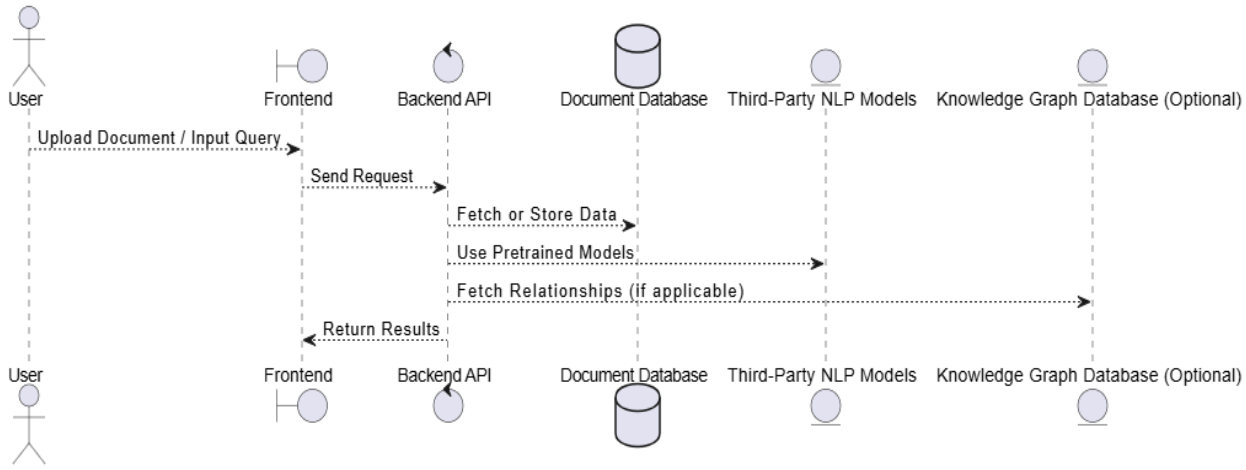
Diagram to show the sequential flow of data and tasks from input (user) to output (results).

5.4 Deployment Diagram



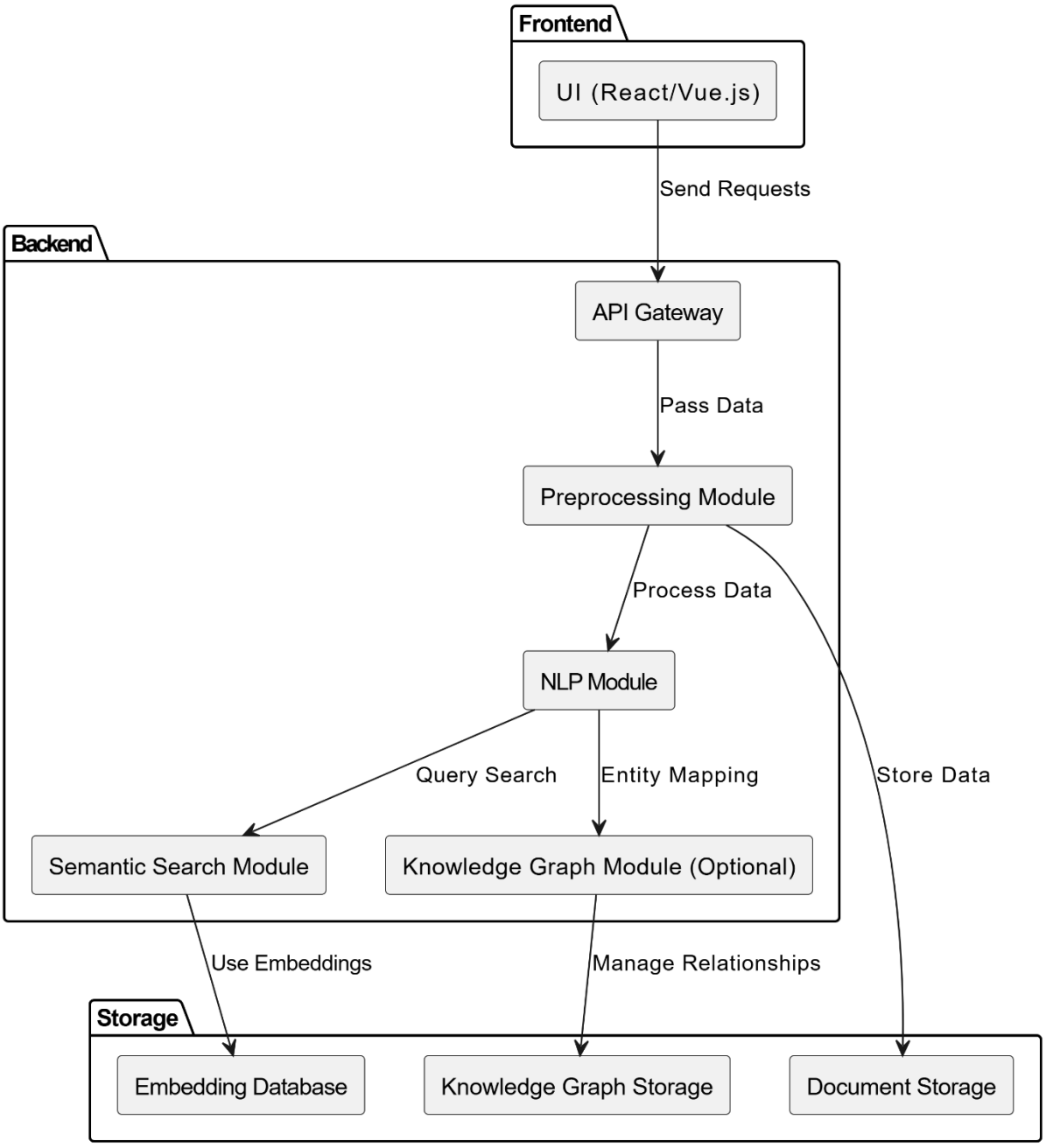
Displays how components are distributed across servers, clouds, or other infrastructure. Includes databases, APIs, and user-facing applications

5.5 System Context Diagram



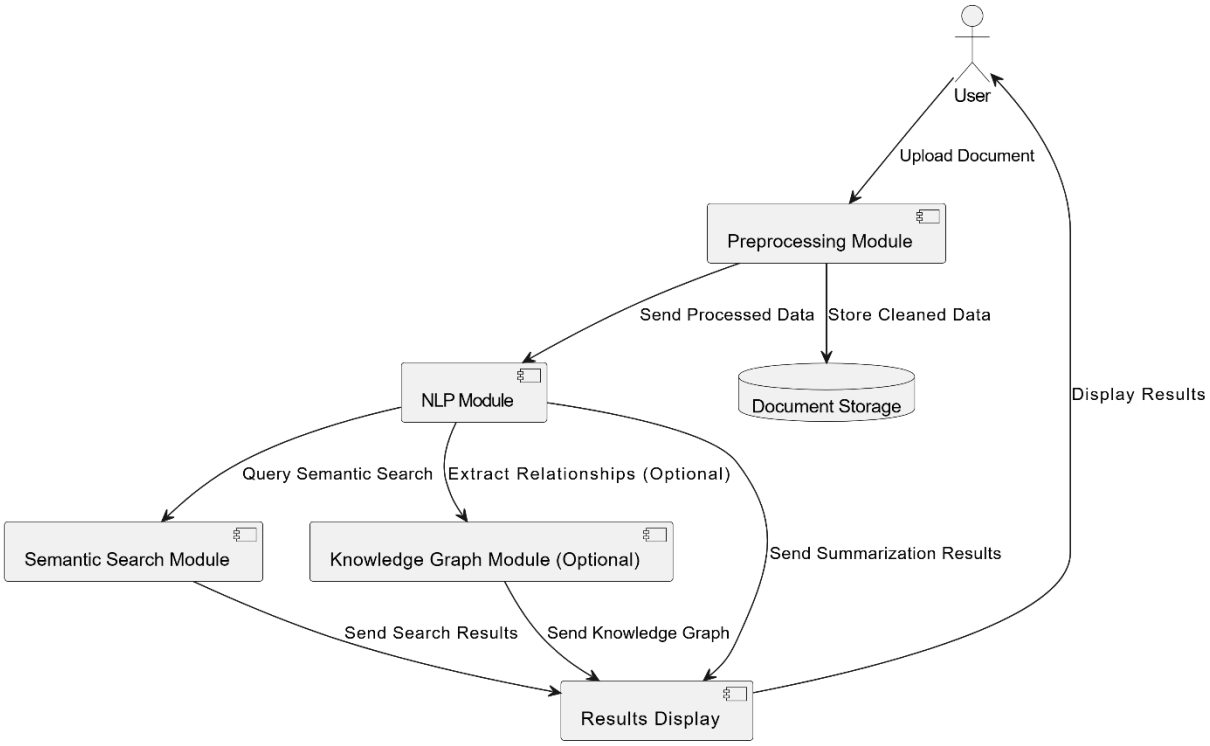
How the system interacts with external entities such as users, databases, third-party APIs are highlighted from this diagram.

5.6 Component Diagram



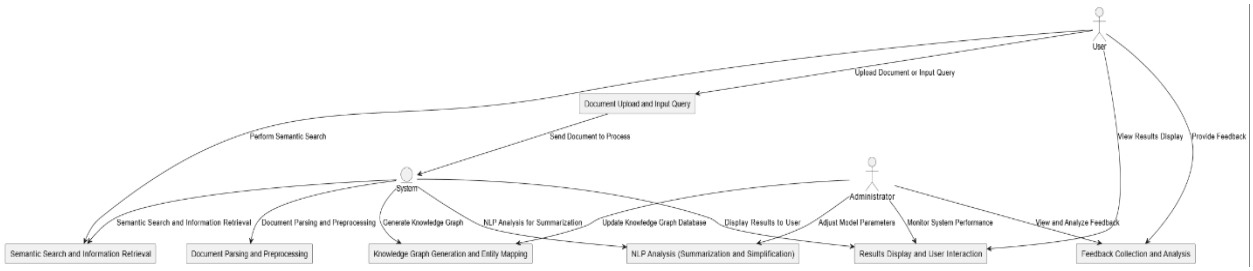
This diagram shows how the key functional modules interact with each other.

5.7 Data Flow Diagram (DFD)

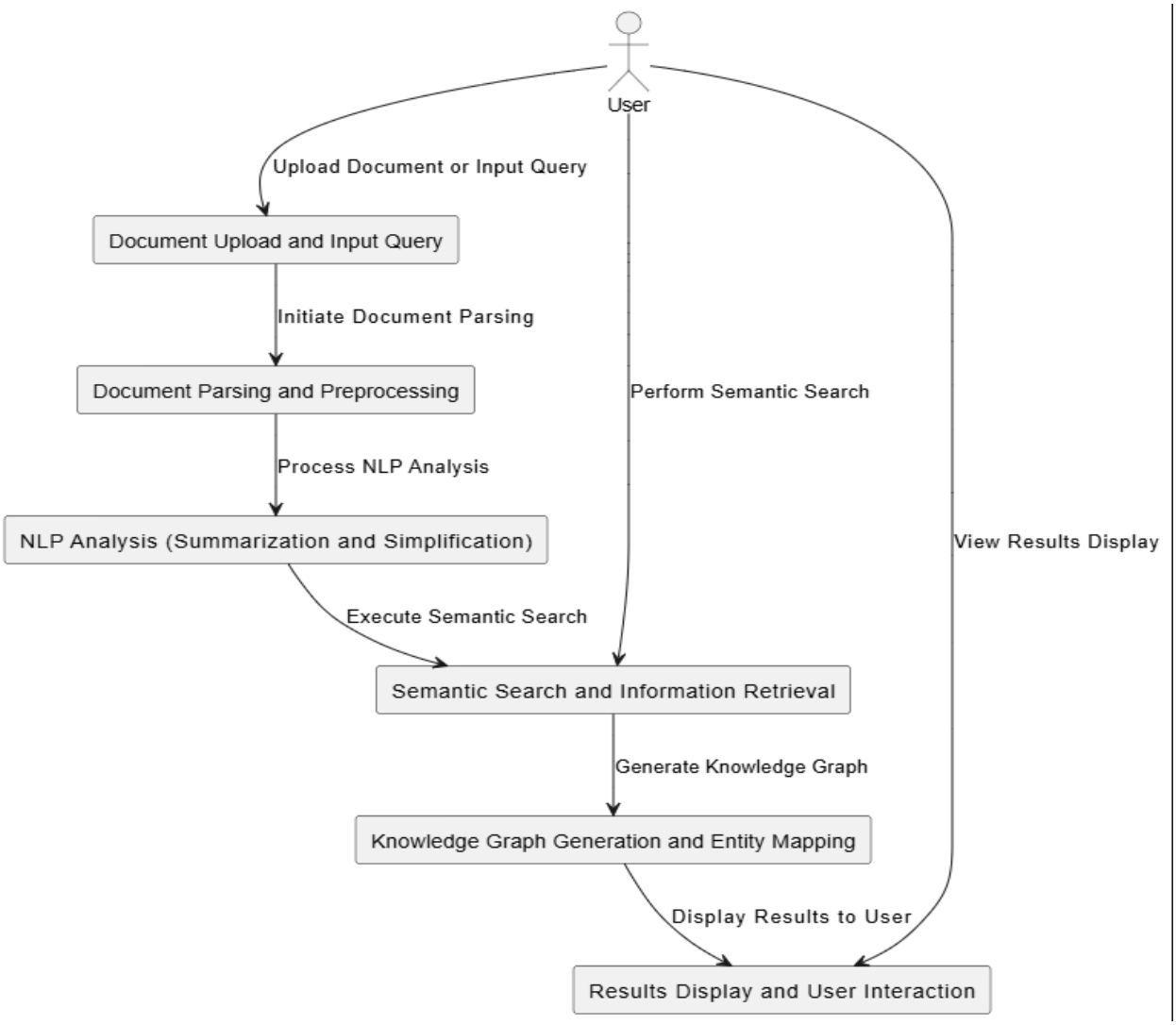


Tracks the movement of data within the system, from input sources to processing modules and storage.

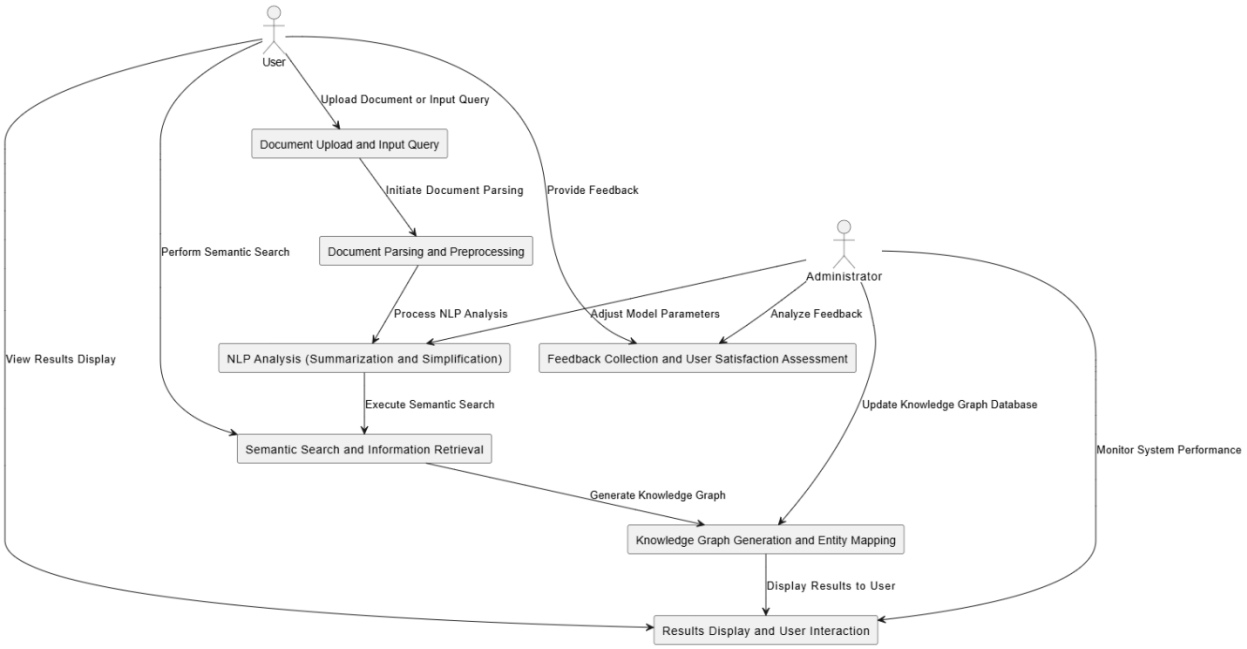
5.8 Use Case Diagrams



(Use Case Diagram with System and Administrator – zoom in)



(Use Case Diagram with just the User)



(Use Case Diagram with just the User and Administrator)

PS – Concepts / Frameworks demonstrated by these figures can be subject to change.

6. Method of Approach

Data Preprocessing: This phase ensures raw legal documents are ready for NLP processing.

- **Text Extraction / Data Collection**:
  Tools like PyPDF2 or pdfplumber extract text from various formats (PDF, Word). For robust extraction, use OCR libraries like Tesseract for scanned documents, and getting access to existing datasets of legal cases related to cybercrimes.
- **Cleaning**:
  Remove unnecessary elements (e.g., headers, footers, page numbers) and normalize text by handling whitespace, punctuation, and encoding issues.
- **Tokenization**:
  Split the text into smaller units (words, sentences). Libraries like spaCy ensure domain-specific tokenization, preserving legal terms.
- **Metadata Extraction**:
  Capture key information (e.g., case name, date) using regular expressions (REGEX) or rule-based methods.

NLP Processing

Core NLP tasks for legal document analysis include:

- **Summarization**:
  - Fine-tune a pretrained transformer (e.g., **Legal-BERT**, **T5**) on cybercrime datasets.
  - Use abstractive methods to create concise summaries, retaining critical legal details.
- **Term Simplification**:
  - Implement a glossary of legal terms using dictionaries (e.g., Black's Law Dictionary).
  - Use NLP models to generate simplified explanations dynamically, ensuring clarity for non-legal experts.

Advanced Techniques:

- Use multi-task learning to combine summarization and simplification tasks in one model.
- Add highlight generation to mark key sections in the document.

Semantic Search

Improve search capabilities using context-aware techniques:

- **Vector Embeddings**:
  Use **Sentence-BERT** to convert text and queries into vector embeddings.
- **Similarity Scoring**:
  Implement cosine similarity to rank document sections based on relevance to user queries.

- **Query Refinement**:
  Use NLP models to suggest refined queries based on initial user input.

Advanced Techniques:

- Add hybrid search (combining keyword-based and vector-based search) for enhanced precision.
- Train embeddings on legal corpora for improved domain-specific performance.

Knowledge Graph (Optional)

If feasible, creation of graphs to visualize relationships between entities:

- **Entity Recognition and Linking**:
  Use pretrained models for Named Entity Recognition (NER) to identify entities (e.g., laws, parties).
- **Graph Database**:
  Use Neo4j to store entities and relationships. Query data dynamically for exploration.
- **Visualization**:
  Render graphs interactively using pyvis or NetworkX.

Application Development

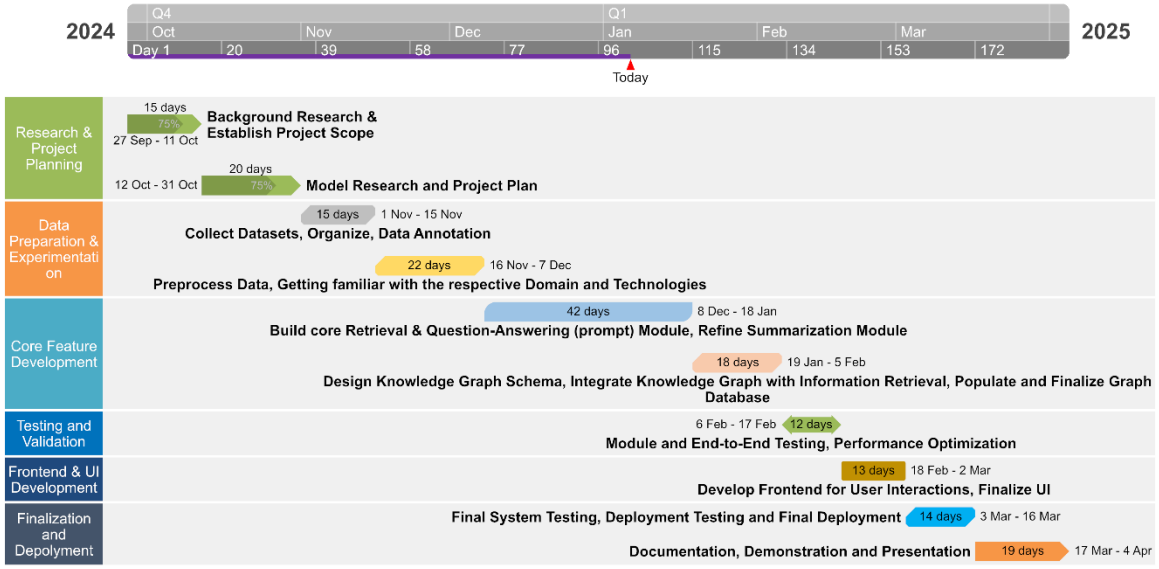
To create a public-facing tool:

- **Frontend**:
  - Use **React** or **Vue.js** for dynamic, user-friendly interfaces.
  - Features: File upload, search bar, result display, and interactive graphs.
- **Backend**:
  - Use **Flask** or **Django** for processing workflows, connecting the frontend to NLP and database modules.
  - Make sure endpoints can be expanded to support new features in the future.
- **Deployment**:
  - Deploy on **Heroku**, **AWS**, or **Google Cloud** for scalability and public accessibility.

Project Management

- **Framework**: Agile methodology with bi-weekly sprints for flexibility and regular feedback.
- **Task Breakdown**:
  - Sprint 1: Preprocessing pipeline and initial model setup.
  - Sprint 2: Implement semantic search.
  - Sprint 3: Integrate NLP tasks with the frontend.
  - Sprint 4: Testing, optimization, and final deployment.

7. Initial Project Plan



8. Risk Analysis

Data Availability,

Difficulty accessing cybercrime related legal documents due to restricted availability or insufficient public datasets.

Likelihood: High

Impact: High – Insufficient data can hinder the model training and system validation.

Mitigation Strategies:

- Use publicly available legal repositories or government case archives to gain access to data.
- Create synthetic datasets / dummy datasets based on laws and regulations, hypothetical scenarios.
- Collaborate with university law departments or legal professionals for anonymized case data.

Technical Complexity,

Challenges in implementing advanced features such as semantic search and knowledge graph generation, due to inexperience and immaturity within the field of NLP.

Likelihood: Medium

Impact: Very High – Failure to implement core functionalities will jeopardize the entire project.

Mitigation Strategies:

- Break down tasks into smaller, manageable modules, starting with essential features.
- Get / Seek extra assistance from experienced professionals / individuals in the related fields.

Time Constraint,

Being unable to complete all the tasks within the desired / designated period or timeline.

Likelihood: Medium

Impact: High – Delays could branch into different aspects of the project.

Mitigation Strategies:

- Prioritizing core features such as Summarization and Semantic search over difficult implementations such as knowledge graphs.
- Using agile methodology with bi-weekly sprints to tackle everything.

Resource Availability

Inadequate availability of computer resources (such as GPUs) for testing and training models.

Likelihood: Low

Impact: Medium – It will take more time but will not directly affect the project's functionality.

Mitigation Strategies:

- Using cloud services for development.
- Code optimization to reduce computational power.

Budget Constraints

Limited budget may restrict access to premium tools, APIs, or datasets.

Likelihood: Low

Impact: Medium – May incur limitations but it is still possible to achieve project goals.

Mitigation Strategies:

- Using academic license for software when available.
- Using Open-Source tools and technologies.

Model Performance Issues

There is a chance on models underperforming on cybercrime-specific legal texts due to various reasons, main one lacking domain-specific training data.

Likelihood: High

Impact: Medium – May result in lower accuracy in search and summarization tasks.

Mitigation Strategies:

- Fine tuning pre trained models and Transfer learning.
- Regularly evaluate models using metrics like BLEU, ROGUE.

Ethical and Privacy Concerns

Inappropriate handling of private or sensitive legal data may result in moral dilemmas or regulatory violations.

Likelihood: Low

Impact: High - Legal or ethical violations could damage the project's credibility.

Mitigation Strategies:

- Anonymize any sensitive data and make sure all data utilized conforms with public information standards.

Summary Table of Risks,

Risk	Likelihood	Impact	Mitigation Strategies
Data Availability	High	High	Use public datasets, create synthetic data.
Technical Complexity	Medium	Very High	Break tasks into modules, use pretrained models, seek help.
Time Constraints	Medium	High	Prioritize core features, follow Agile.
Resource Availability	Low	Medium	Use free-tier services, optimize code for efficiency.
Budget Constraints	Low	Medium	Leverage free tools, apply for educational licenses
Model Performance Issues	High	Medium	Fine-tune models, use transfer learning, evaluate iteratively/
Ethical and Privacy concerns	Low	High	Anonymize data, comply with policies.

9. Stakeholder Analysis

Primary Stakeholders

Legal Professionals
Role: End Users of the System. They rely on the tool for quick summaries, contextual search and visualization that maps relationships entities. Also plays a part in the project itself to provide accurate legal information.

Involvement: Provide feedback on usability, relevance of results, and accuracy from a legal standpoint.

Academic Supervisors
Role: Guide the project from a research and technical perspective.

Involvement: Validate methodologies, provide feedback on implementation, and assess project outcomes.

University Law Department
Role: Collaborators for accessing domain- specific knowledge, legal documents, and evaluation of project relevance.

Involvement: Offer insights into various legal statutes and potential datasets.

Non-Legal Experts / Public
Role: End user of the system. They also rely on summaries, and simpler interpretations of legal texts including visualizations for their own benefits.

Involvement: Providing feedback on the system in a public eye perspective.

Secondary Stakeholders

AI and NLP Researchers
Role: Offer advice on certain technologies and methods.

Future Developers or Contributors
Role: Build upon / Extend the project after completion to expand features or adapt new domains.

Involvement: Requires comprehensive documentation for seamless handover.

10. Scalability and Future Work

Scalability
The tool will be modular, allowing easy integration of new features, such as additional legal domains or advanced visualization techniques.

Future Work
- Extending to other areas of the law such as IP rights law (Intellectual Property), contract laws etc.
- Enhancing the knowledge graph to include temporal relationships (Ex - case timelines).
- Adding platform support for different languages.

11. Key Performance Indicators (KPIs)

Objective	KPI	Target
Document Summarization	BLEU or ROGUE score	Achieve >85% accuracy
Semantic Search	Precision and Recall Metrics	Precision and Recall >85%
Legal Term Simplification	User comprehension feedback	>90% satisfaction rate
Knowledge Graphs	Relationship accuracy in graphs	>80% correctness
UI Usability	Time taken to upload and retrieve results	<10 seconds average response time

12. References

Chalkidis, I. *et al.* (2020) 'LEGAL-BERT: The Muppets straight out of Law School'. Available at: http://arxiv.org/abs/2010.02559.

Dragoni, M. *et al.* (2016) *Combining NLP Approaches for Rule Extraction from Legal Documents Combining NLP Approaches for Rule Extraction from Legal Documents. 1st Workshop on MIning and REasoning with Legal texts Combining NLP Approaches for Rule Extraction from Legal Documents*. Available at: https://wordnet.princeton.edu/.

Imogen, P.V., Sreenidhi, J. and Nivedha, V. (2024) 'AI-Powered Legal Documentation Assistant', *Journal of Artificial Intelligence and Capsule Networks*, 6(2), pp. 210–226. Available at: https://doi.org/10.36548/jaicn.2024.2.007.

Merchant, K. and Pande, Y. (2018) *NLP Based Latent Semantic Analysis for Legal Text Summarization*. IEEE.

Sabrina Univ-Prof Axel P, A.K. (2021) *Knowledge Graphs for Analyzing and Searching Legal Data*.

Sachidananda, V., Kessler, J.S. and Lai, Y. (2021) 'Efficient Domain Adaptation of Language Models via Adaptive Tokenization'. Available at: http://arxiv.org/abs/2109.07460.

Vayadande, K. *et al.* (2024) 'AI-Powered Legal Documentation Assistant', in *Proceedings - 2024 4th International Conference on Pervasive Computing and Social Networking, ICPCSN 2024*. Institute of Electrical and Electronics Engineers Inc., pp. 84–91. Available at: https://doi.org/10.1109/ICPCSN62568.2024.00022.

Yang, W. *et al.* (2019) 'Legal Judgment Prediction via Multi-Perspective Bi-Feedback Network'. Available at: https://doi.org/10.24963/ijcai.2019/567.

Zakir, M.H. *et al.* (2024) 'Artificial Intelligence and Machine Learning in Legal Research: A Comprehensive Analysis', *Qlantic Journal of Social Sciences*, 5(1), pp. 307–317. Available at: https://doi.org/10.55737/qjss.203679344.

Zheng, J. *et al.* (2024) 'Fine-tuning Large Language Models for Domain-specific Machine Translation'. Available at: http://arxiv.org/abs/2402.15061.

Zhong, H. *et al.* (2020) 'How Does NLP Benefit Legal System: A Summary of Legal Artificial Intelligence'. Available at: http://arxiv.org/abs/2004.12158.

THANK YOU!

IN PARTNERSHIP WITH PLYMOUTH UNIVERSITY

Name: Weerasinghe Dissanayakalge Methsara Nisaga Dissanayaka
Student Reference Number: 10899302

Module Code: PUSL 3190	Module Name: Computing Individual Project
Coursework Title: AI for Legal Document Analysis	
Deadline Date: 05/05/2024	Member of staff responsible for coursework: Dr. Mohomed Shafraz

Program: BSc. (Hons) in Data Science
Please note that University Academic Regulations are available under Rules and Regulations on theUniversity website www.plymouth.ac.uk/studenthandbook.
Group work: please list all names of all participants formally associated with this work and state whether the work was undertaken alone or as part of a team. Please note you may be required to identify individual responsibility for component parts. ***We confirm that we have read and understood the Plymouth University regulations relating toAssessment Offences and that we are aware of the possible penalties for any breach of theseregulations. We confirm that this is the independent work of the group.*** Signed on behalf of the group:
Individual assignment: ***I confirm that I have read and understood the Plymouth University regulations relating to Assessment Offences and that I am aware of the possible penalties for any breach of these regulations. I confirm that this is my own independent work.*** Signed: *Weerasinghe Dissanayaka*
Use of translation software: failure to declare that translation software or a similar writing aid has been used will be treated as an assessment offence. I *have used/not used translation software. If used, please state name of software……

UNIVERSITY OF PLYMOUTH

PUSL3190 - Computing Individual Project

Interim Report

AI for Legal Document Analysis

Supervisor: Dr. Mohamed Shafraz

Name: Weerasinghe Dissanayakalage Methsara Nisaga Dissanayaka
Plymouth Index Number: 10899302
Degree Program: BSc. (Hons) in Data Science

Table of Contents

1. Introduction

In today's world, legal documents are a massive maze of dense language and endless details, which makes life hard for both legal pros and anyone trying to understand the law. This interim report dives into the early stages of this project—AI for Legal Document Analysis—that tackles these challenges head-on.

At its core, this project is driven by the need to simplify and streamline how legal texts are handled. Traditional methods require tons of manual effort and often leave people sifting through confusing jargon and overly complex language. The aim is to create an intelligent system that not only speeds up the process but also makes it way easier to extract meaningful insights from legal documents.

A clear overview of the challenges that are being addressed and the conceptual strategies that are being deployed to reshape legal document analysis will be demonstrated in this report.

1.1 Problem Definition

The challenges of Legal Document Analysis in a Digital Age

The legal sector of a nation serves as the backbone of maintaining societal order, providing the framework for governance, rights, and responsibilities. In order to uphold justice, guarantee compliance and perform these very important tasks, legal institutions rely on a vast array of documents, including statutes, case law, contracts, and regulations. These documents are constructed with detailed language and specific jargon to the Legal field to capture complex legal ideas, therefore requiring interpretation by skilled professionals (Lawyers, Judges and Legal Scholars). However, in the digital era, traditional approaches to legal document analysis are increasingly inadequate due to the sheer volume and complexity of legal writing. This has created a growing demand for precise, scalable technologies capable of large-scale legal text analysis and interpretation.

Legal institutions are now facing unprecedented challenges in processing and analyzing the vast amounts of legal documentation. Laws, legislations, case verdicts, and arguments are not only intricate but are also subject to constant change, driven by evolving societal norms, digital advancements, and global connectivity. Additionally, legal texts are often ambiguous, requiring significant effort to interpret. Ambiguities in language can lead to inconsistencies in understanding and application, further complicating the process for legal professionals. These challenges are amplified when working within specialized legal domains such as intellectual property law, digital privacy, or computer crimes, where terminology is technical, dynamic, and often unfamiliar.

People who work in the legal field such as judges, attorneys and analysts must review tons of documents by hand to retrieve relevant precedents, decipher complex clauses, and apply legal principles to particular cases accordingly. This manual process is labor-intensive, time-consuming, and prone to human error, especially in high-stakes scenarios. Professionals must also contend with the pressure of responding within tight timeframes, where delays in retrieving relevant legal information can have serious repercussions. Furthermore, the inefficiencies of manual analysis often lead to missed precedents or misinterpretations, particularly when outdated legal frameworks are applied to contemporary challenges.

Recognizing these limitations, this project aims to address the core issues plaguing legal document analysis, including:

- **Complexity and Ambiguity**: Unclear language in legal documents can lead to multiple interpretations, necessitating further clarification.

- **Data Volume and Time Constraints**: The exponential growth of legal texts, combined with time-sensitive demands, exacerbates inefficiencies in locating critical information.

- **Domain-Specific Challenges**: Specialized legal domains like computer crimes require targeted tools to manage their unique terminology and complexities.

- **Inefficiencies in Manual Analysis**: Traditional methods lack scalability, increasing the risk of errors and delays.

To tackle these challenges, the project leverages artificial intelligence (AI) and advanced natural language processing (NLP) techniques. The proposed system will focus on document summarization, legal term simplification, semantic search, and entity relationship mapping, tailored specifically to the domain of computer crimes. By narrowing the scope to cybercrime and referencing cases under the **Computer Misuse Act 1990 - UK** (Main legislation that criminalizes unauthorized access to computer systems and data, and the damaging or destroying of these in the UK), this tool provides a more focused solution that enhances accuracy and relevance.

This targeted approach allows for precise fine-tuning of NLP models, enabling the system to generate summaries, simplify legal jargon, and visualize knowledge graphs. By training on a specialized dataset of computer crime cases, the system can deliver deeper insights into this subset of law, helping legal professionals save time and improve decision-making. Moreover, this project not only aims to address current inefficiencies but also lays the foundation for expanding AI-driven solutions to other legal domains in the future, bridging the gap between legal expertise and technological innovation.

1.2 Project Objectives

13. Develop a Legal Document Processing Pipeline. (**Creating**)
    - Construct a robust pipeline capable of handling multiple document formats (Ex – PDF, DOCX, TXT). This pipeline will extract text, process metadata, and normalize documents to prepare them for analysis.
    - Ensure everything related is handled with at least 95% accuracy.

14. Integrate enhanced NLP methods for Advanced Summarization and Simplification of Legal Text. (**Integrating**)
    - Utilize transformer-based NLP models (Ex: Legal - BERT) fine-tuned on legal datasets to extract key points, generate concise summaries and simplify complex legal writing, streamlining document review.
    - Target a summary length reduction of up to 70% while retaining critical legal information.
    - Implement a dynamic glossary with contextual linking to the processed document for quick reference of complex legal terms.

15. Implement Semantic Search with Vector embeddings. (**Applying**)
    - Enable retrieval of information based on contextual meaning rather than strict keyword matches using vector embeddings, significantly enhancing user navigation through complex documents.
    - Achieve a precision and recall rate of at least 85% for user queries.

16. Visualize legal document relationships through Knowledge Graphs. (if feasible)
    - Construct a knowledge graph model that maps out relationships and dependencies within the text. This is crucial for identifying primary entities, connections and providing an interactive layer of insight into legal networks.
    - Visualize connections and relationships with clear labels and metadata for at least 80% of entities in the dataset.

17. Customize the AI Tool for Computer Crime case law using domain-specific Model Training and Evaluate Tool Performance. (**Analyzing & Evaluating**)
    - Fine-tune the AI system to address the specific requirements and nuances of a particular domain (cybercrime cases) and carry out performance evaluations on a dataset specific to the selected domain. Thus, creating a precise and functional AI solution for a targeted legal area.

18. Design an Intuitive & Interactive User interface. (**Developing**)
    - Craft an intuitive User interface for effortless navigation and interaction with complex legal data.

2. System Analysis

2.1 Facts Gathering Techniques

2.1.1 Literature Review

One doesn't need to be a smart individual to understand that the Legal Sector is the framework that keeps everything in its rightful place, from the highest echelons of authority to the most fundamental societal functions. This structured hierarchy ensures that order, justice, and rights (not to mention that they are also defined by the law) are preserved, safeguarding society from chaos and fostering a foundation upon which all individuals and institutions can rely. Carefully crafted Documents by legal experts which contain legal statutes, case law, contracts, and regulations uphold this structure, ensuring every detail is meticulously accounted for. But as these documents try to take everything into account, it means that these documents become extremely complex and often quite lengthy. And as the world and human activities continue to expand, the volume of legal data will also continue to increase, thus making these documents, which contain relevant legal sentences, frequently subjected to change. Therefore, thorough Legal Document Analysis is pivotal for the application and evolution of not just the legal sector but for any human endeavor.

However, current methods for Legal Document Analysis which are generally man-powered faces a lot of challenges in this modern era. The reliance on these human-driven document processing introduces issues such as time constraints, inefficiencies in information retrieval, susceptibility to errors, and vulnerable to various other problems. Apart from the need of human labor (experts in relevant fields), even the computer based legal document management tools has many shortcomings, as most of them primarily focus on document storage and retrieval through keyword-based search, which fails to capture the nuanced relationship and context-specific language often present in legal texts. This is particularly true in fields like cybercrime, where cases and statues are frequently intricately linked.

With the introduction of Artificial Intelligence (AI) in this digital era to many sectors, new possibilities have emerged in automating and enhancing various tasks in the Legal Sector as well, specifically in Legal Document Analysis. AI technologies such as Natural Language Processing (NLP), Deep Learning and Machine Learning have already been applied to the Legal Sector in order to transform raw legal text into structured, searchable and interpretable data (Zhong *et al.*, 2020). Pretrained language models such as BERT and RoBERTa are widely used for NLP tasks among a variety of industries and have demonstrated potential for legal applications when fine-tuned and optimized on domain-specific data. This gave birth to models such as Sci-BERT (Trained on biomedical and computer science literature), Fin-BERT (Focused on financial services), and Bio-BERT

(Specialized in biomedical literature) etc., and among these specialized models, LEGAL-BERT stands out as particularly relevant this project. Although it is quite self-explanatory, this model is designed specifically for the Legal Domain, offering tailored solutions such as,

- Legal Document Classification
- Named Entity Recognition
- Legal Judgement Prediction
- Legal Statute Identification
- Semantic Segmentation
- Court Judgement Prediction

...that aligns perfectly with the direction of this project (Chalkidis *et al.*, 2020). Additionally, techniques like adaptive tokenization have been developed to modify pretrained models' tokenizers to better handle specialized vocabulary in legal documents. Research by (Sachidananda, Kessler and Lai, 2021) highlights that adapting tokenizer to legal terminology significantly boosts model performance on specialized tasks without the costs of training a new model from scratch.

AI is being utilized more and more in the legal industry for tasks ranging from document classification, rule extraction to even predicting case verdicts (Yang *et al.*, 2019). There are already existing applications that can perform document analysis. Research from (Zakir *et al.*, 2024) shows how much AI has advanced the processing of legal documents. Transformation models like Legal-BERT are proof that AI can have significant impact on the legal sector. Another major component that is believed to be not utilized to its full potential yet is knowledge graphs. Named Entity Recognition (NER) along with knowledge graphing offers a promising approach to organizing legal data, linking statutes, cases, and other legal entities into a structured network that facilitates query-based exploration. Erwin Flitz's dissertation on knowledge graphs (Sabrina Univ-Prof Axel P, 2021) underlines the benefits of linking legal data through standardized identifiers (like ELI and ECLI) and ontologies, which enables complex queries and cross-references across legal systems. These methods can help legal professionals and non-experts by visualizing relationships between legal entities and providing a comprehensive view of case law and statutory connections.

Numerous research offers insights into the application of AI in the legal sector, showcasing techniques for semantic searching using vector embeddings that goes beyond the limitations of keyword-based searching, vector databases, extracting and summarizing of legal texts etc. For instance, (Dragoni *et al.*, 2016) discuss methods of rule extraction from legal documents using NLP, combining syntactic parsing with semantic analysis to identify regulatory rules. Meanwhile (Merchant and Pande, 2018) describes the usage of latent semantic analysis for legal text summarization. Such approaches demonstrate how multi-step NLP techniques can parse dense legal terminology into comprehensible terms,

making legal reasoning more accessible through AI. Similarly, adaptive domain training techniques such as those proposed by (Sachidananda, Kessler and Lai, 2021) and (Zheng *et al.*, 2024), show that fine-tuning models with domain-specific tokens or language data can significantly enhance NLP results for specific domains inside the legal sector itself.

Reflecting on these findings, there is a clear Research Gap and Need for systems that goes beyond simple document retrieval to incorporate contextual understanding, semantic search, and entity graphing, particularly in a specific domain as we found out the benefits in using fine-tuned models for a specific domain. Current solutions primarily cater to broad legal applications, offering generalized search functions, summarization and basic information retrieval, which fall short in addressing the specific nuances of various domains within the legal sector (because law covers every aspect in the world, even the simplest ones an individual can think of), particularly when it comes to highly specialized / specific fields such as cybercrimes. For example, complex domain-specific language can be misinterpreted by generic NLP models, even the ones that are fine-tuned for the Legal Sector itself, making precise entity recognition, relationship mapping, and contextual summarization challenging (Zhong *et al.*, 2020). Furthermore, only a small number of current tools completely incorporate sophisticated capabilities such as semantic search, vector-based embeddings, and knowledge graph-based visualization tailored to legal contexts, leaving a clear gap for tools that integrate these capabilities into a unified, domain-specific platform (cybercrimes). Thus, the proposed project addresses these gaps by building a domain-specific AI tool focused on cybercrime, which will,

4) Utilize fine-tuned NLP models trained on cybercrime case data, enhancing the accuracy of summarization and term interpretation.
5) Incorporate semantic search powered by vector embeddings, improving the tool's relevance and retrieval precision.
6) Use knowledge graphs to map and visualize the relationship between cases, statutes, and involved entities specific to cybercrime.

This combination of domain-specialized NLP and advanced visualization will offer a more nuanced, contextual approach than current general-purpose legal AI tools, meeting an unaddressed gap within the legal AI landscape. This outcome will also provide a replicable framework adaptable to other areas within the legal sector, ultimately contributing to the evolution of legal towards more intuitive and precise AI-Assisted legal research. (Imogen, Sreenidhi and Nivedha, 2024)(Vayadande *et al.*, 2024)

2.1.2 Case Study Analysis: Reviewing Cybercrime Cases and Relevant Legal Documents

An extensive case study analysis was conducted, primarily using cybercrime cases available on BAILII (British and Irish Legal Information Institute – provided relevant case data regarding Computer Misuse Act 1990 upon request). This study was created to comprehend not only the procedural and factual elements of cybercrime litigation, but also the complicated language and legal reasoning that support these cases, given the enormous complexity of legal papers. Recognizing the importance of data availability, the focus was shifted from Computer Crime Act No. 24 of 2007 (Sri Lanka) to UK law, ensuring that there are enough cases to be reviewed (for model training etc.) and are also directly relevant.

The case studies provided deep insights into the handling of cyber offences such as unauthorized access, digital fraud, and data breaches. By dissecting judgments and opinions, the analysis highlighted how legal practitioners interpret technical evidence, apply statutory provisions, and navigate the nuances of cybercrime under the UK framework. This understanding has been critical in modelling the development of this AI-based legal document analysis system, particularly in shaping the natural language processing models to capture and simplify complex legal terminologies without losing the essence of judicial reasoning.

In parallel with case analysis, a comprehensive review of other relevant legislative and regulatory documents was undertaken. The analysis included key legal frameworks that influence how cybercrime is approached, including:

- **Data Protection Act 2018 (DPA) and UK General Data Protection Regulation (UK-GDPR):** These documents underscore the importance of data privacy and set stringent guidelines for handling personal data, aspects that are integral to the interpretation and processing of legal documents.
- **Network and Information Security Directive (NIS2) and Digital Operational Resilience Act (DORA):** These regulations highlight the critical role of operational resilience and cybersecurity, providing a contextual background for understanding how legal obligations are enforced in the digital realm.
- **UK Operational Resilience Framework:** This framework offers insight into the continuity and robustness expected of organizations in the face of cyber threats, further informing the legal narratives found in cybercrime cases.

- **EU Cybersecurity Act and EU Cyber Resilience Act:** These Acts provide a broader European perspective on cybersecurity standards and resilience, influencing legal arguments and regulatory expectations within the UK.
- **Computer Misuse Act 1990:** As one of the foundational legal documents dealing with unauthorized computer access and cyber offences, it remains a cornerstone in the analysis of cybercrime cases.
- **EU Artificial Intelligence Act:** This emerging legislative framework is beginning to shape discussions around the use of AI in legal settings, particularly in how automated tools process and analyze legal texts.
- **Telecommunications (Security) Act 2021 and Privacy and Electronic Communications Regulations (PCER):** These documents address the security and privacy aspects of digital communications, further adding layers to the legal context of cybercrime investigations.

(Chin, 2025)

Our knowledge about legal landscape has been enhanced by this dual approach of examining legislative papers and case law. On one hand, the detailed case studies offer useful perspectives on the difficulties faced by attorneys in cybercrime matters as well as court reasoning. The legislative review, on the other hand, ensures that the technologies used not only correctly understand legal language but also complies with current legal standards and compliance requirements by grounding these insights in a structured regulatory framework.

Collectively, this extensive analysis has laid a solid foundation for the continued development of this AI-driven tool. By intertwining the practical insights from real-world cases with a deep understanding of the prevailing legal and regulatory environment, the project is well-positioned to address the challenges of legal document analysis in the realm of cybercrime.

2.1.3 Interviews and Discussions

Semi-structured interviews were conducted with diverse stakeholders, such as legal practitioners, academics, and industry practitioners by various means. Such interviews provided in-depth insights into the practical nuances and challenges of legal document analysis. The interviewees described their first-hand experience of legal research and

identified specific issues—such as complexity in legal language, inefficiency in current document review processes, and the need for improved clarity in case summaries. These discussions not only illuminated the issues of functioning in legal practice but also helped to identify the most critical areas where an AI-based solution could significantly impact. The iterative nature of these discussions helped us to constantly refine our approach so that our system remains aligned with real requirements.

2.2 Existing Systems and their Drawbacks

Current practices in legal document analysis rely predominantly on traditional methods that are increasingly inadequate for addressing the complexity and volume of modern legal texts.

Manual Legal Document Review:
Lawyers traditionally read and review documents manually - a time-consuming and labor-intensive process. Not only does this slow down the discovery of valuable insights, but it also opens the door to human error, with subtle but significant legal distinctions potentially being missed.

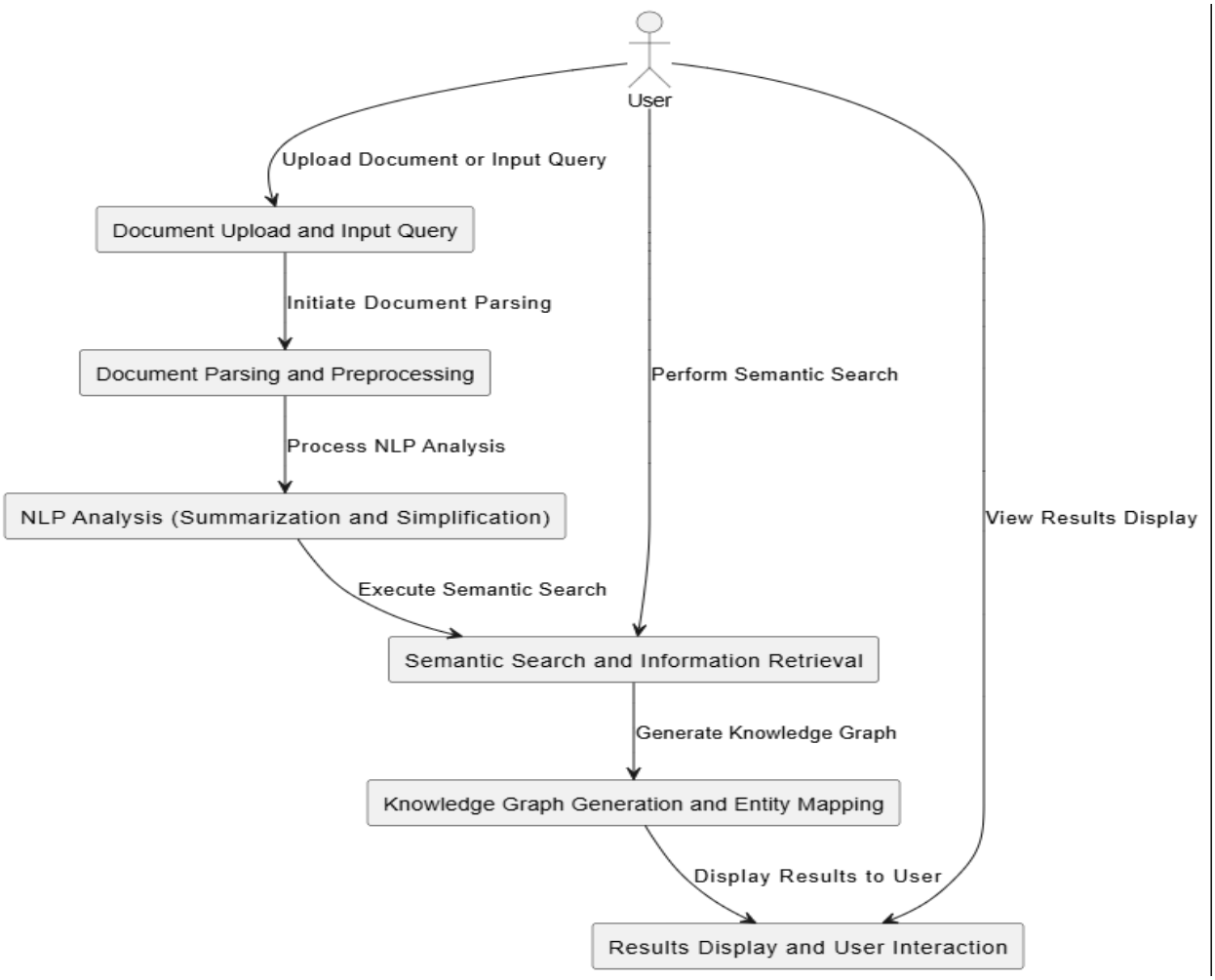
Keyword-Based Search:
Most legal databases employ keyword-based search techniques. While the systems succeed in retrieving documents containing target terms, they do not capture the general sense and underlying context expressed in the language of the law. The failure results in reduced accuracy while seeking to identify relevant cases or precedents because semantic relationships and implicit references are largely ignored.

Traditional Legal Research Tools:
Westlaw and LexisNexis have long been staples of legal research. Although they offer immense repositories of legal information, their methods are based primarily on traditional search and retrieval techniques. They typically don't include advanced AI capabilities, such as deep semantic analysis, auto-summarization, and relationship mapping, required to obtain deeper insights from complex legal content.

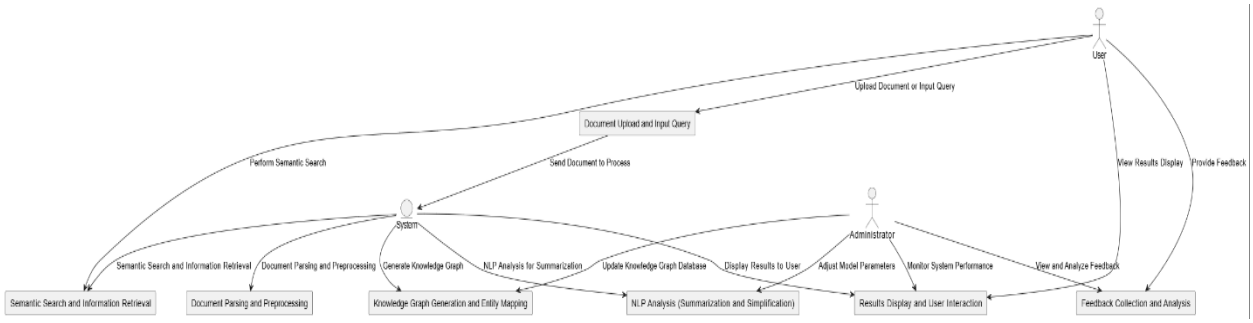
Lack of Automation for Specialized Needs:
There is a noticeable gap of automation in specialized areas like cybercrime case analysis. None of the existing systems are meant to process, summarize, and map relations in legal case documents involving cybercrimes automatically. The absence of AI-supported features—like dynamic summarization, interactive knowledge graphs, and semantic document retrieval—pinpoints the inability of existing systems to cater to modern legal challenges.

Cumulatively, these limitations highlight the need for a revolutionary solution. The development of an AI-driven legal document analysis system is meant to overcome these challenges by automating the extraction of key insights, enhancing the relevance of search results, and providing a comprehensive analysis that is aligned with the evolving requirements of legal professionals.

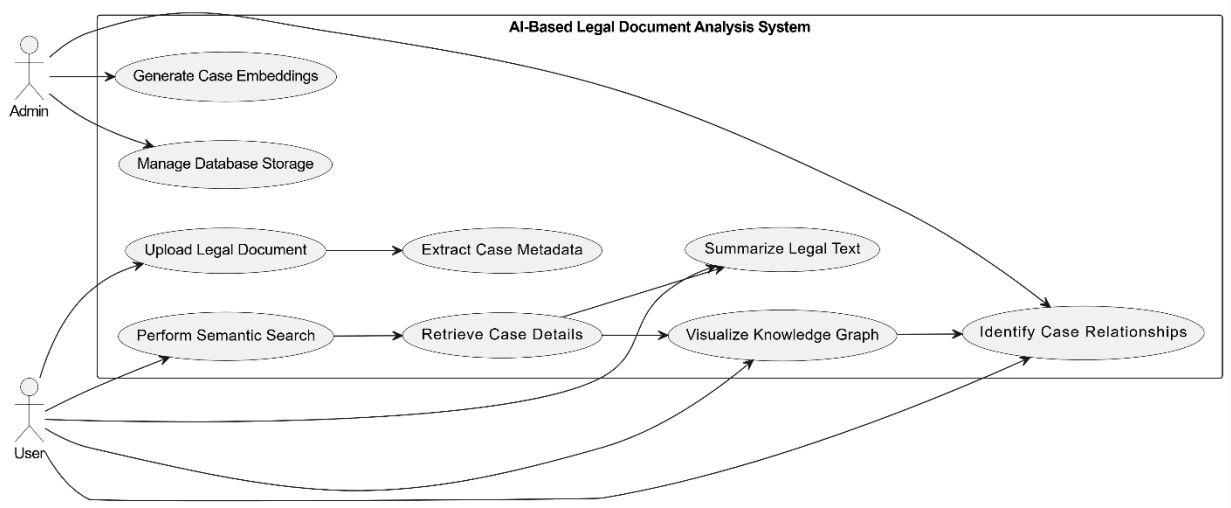
2.3 Use Case Diagram

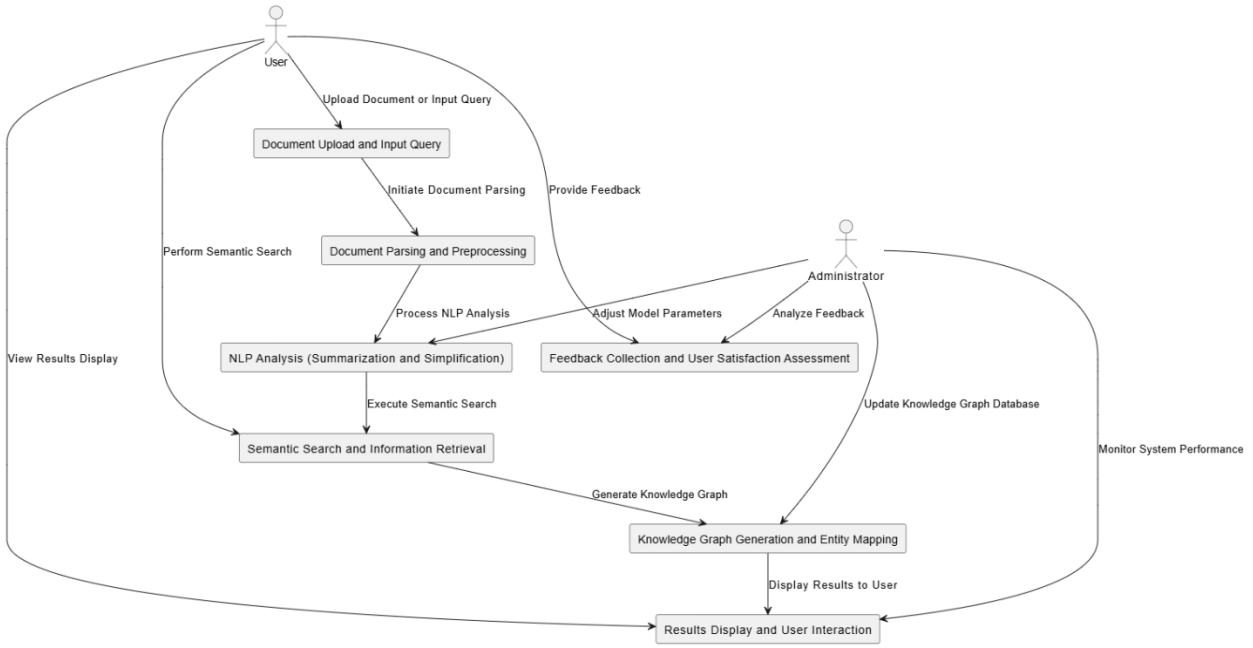


(Use Case Diagram with just the User)



(Use Case Diagram with System and Administrator – zoom in)





(Use Case Diagram with just the User and Administrator)

3. Requirement Specification

3.1 Functional Requirements

- Document Management

  Multi – format upload: For analysis, users ought to be allowed to upload several document formats. (Ex – PDS, DOCX, TXT)

  Automated Preprocessing: The system ought to preprocess these documents by structuring the text and removing unnecessary content.

- Fine-Tuning Models and Optimization

  Domain Specific Adaptation: Optimize existing models using specific datasets (cases related to cybercrime).

- Natural Language Processing (NLP)

  Text Simplification and Summarization: Legal terminology must be simplified and summarized by the system using advanced NLP techniques.

  Semantic Search: Semantic search should be supported so that context-based retrieval is possible moving beyond keyword-based search.

- Knowledge Graph Generation

  Entity Identification and Linking: The system should identify and link key entities (e.g., case names, statutory references, involved parties) generating a knowledge graph of relationships.

  Interactive Exploration: It should be possible for users to examine certain entities by interacting with the knowledge graph.

- Result Presentation and User Interaction

  Comprehensive Output Display: Display a summary of the document along with simplified terms, the knowledge graph, and relevant sections from semantic search.

  Interactive Query Refinement: Give users the option to refine their queries or search, ensuring that they can tailor the analysis to their specific needs.

- Additional Considerations

  User Authentication and Access Control: To protect sensitive legal data, the system should include robust user authentication and role-based access control features.

  Report Generation: The system may also support exploring detailed reports of the analysis, facilitating further review and record-keeping by legal professionals.

3.2 Non – Functional Requirements

- Performance

  The system should process and analyze documents within an acceptable timeframe (minimal latency) to maintain usability.

- Scalability

  Multiple users should be able to utilize the tool without any issues (lag), and it should be able to manage large documents and numerous queries.

- Usability

  The user interface should be intuitive, with easy navigation for interacting with knowledge graphs and viewing summaries.

- Reliability and Accuracy

  Ensure high accuracy through the system and consistent performance with robust error handling and continuous monitoring mechanisms.

- Security

  Confidentiality must be maintained throughout the system enforcing strong authentication and authorization protocols, secure data storage and transmission through encryption, and compliance with relevant data protection regulations.

- Maintainability

  Confidentiality must be maintained during the processing and storage of user queries and documents.

- Compliance

  The system must adhere to all applicable legal and regulatory standards, such as the UK General Data Protection Regulation (UK-GDPR) and other relevant industry-specific guidelines.

3.3 Hardware / Software Requirements (Technological Requirements)

NLP Models: Legal-BERT for legal text processing, GPT-4 for encoding documents into vectors and extracting entities/relationships.

Vector Database: Pinecone, Weaviate, or FAISS for embedding storage and semantic search.

Knowledge Graph Database: Neo4j or Stardog for storing and querying legal entities and relationships.

Python Libraries and Frameworks:

Libraries including spaCy, NLTK, Transformers (by Hugging Face), Gensim for text extraction, entity recognition, topic modeling, and other NLP-related tasks. Also supplemented with libraries such as pandas and scikit-learn for efficient data preprocessing and model evaluation.

Django and FastAPI will be utilized for backend connection and services.

Annotation Tools: Brat or Prodigy for annotating legal documents for training NER models.

Cloud Platforms: Google Cloud Natural Language API, AWS Comprehend, or Microsoft Azure's Text Analytics to provide scalable and reliable NLP functionalities. Also leverage cloud infrastructure to manage extensive datasets, support high-performance computing requirements, and ensure robust data storage and backup solutions.

Storage Solutions: Implement sufficient storage capacity with redundancy and high availability to manage extensive legal document repositories and system logs.

Containerization and Orchestration: Docker for containerization, Kubernetes for orchestration.

(PS- Above mentioned tools may be subjected to changes)

4. Feasibility Study

4.1 Operational Feasibility

The proposed system is highly feasible from an operational standpoint due to the increasing demand for automation in legal document analysis.

Integration with existing operations:
The legal industry relies primarily on manual document review and keyword-based search software that are both time-consuming and of limited effectiveness. The system integrates into the existing legal research processes seamlessly by offering automated document review, AI-based summarization, and semantic search capability. Legal researchers can continue to utilize their existing workflows while enhancing efficiency by leveraging the system's capabilities.

User Adoption
Since the system automates document processing, summarization, and legal relationship mapping, it requires no manpower. Legal professionals, researchers, and law enforcement officers can utilize the system with minimal training due to its ease of use and AI-generated reports. User acceptance is expected to be high given the increasing reliance on AI for legal research and the system's ability to simplify complicated legal documents.

Infrastructure Requirements
The system will be deployed with the assistance of cloud-based services (e.g., AWS, Google Cloud, Azure) to attain scalability and remote access. It does not require extensive local infrastructure, which reduces operational overhead. The use of pre-trained NLP models and vector databases also facilitates document retrieval effectively without straining computing resources.

Legal and Regulatory Considerations
The system development must comply with UK legal frameworks like UK-GDPR and the Data Protection Act 2018 to maintain safe data handling and anonymization. The Computer Misuse Act 1990 prevents unauthorized access, and the EU AI Act demands explainable AI to prevent biases. Compliance with the Telecommunications (Security) Act 2021 and PECR is required with cloud-based services to maintain safe data transmission and privacy protection. By including these legal safeguards, the project remains feasible while being compliant with regulation and minimizing risk.

The system is operationally feasible as it integrates smoothly into existing legal workflows, requires minimal staffing adjustments, leverages scalable cloud infrastructure, and aligns with legal regulations along with accessed data through BAILII. Its user-friendly design and

AI-driven enhancements ensure high adoption among legal professionals, making it a practical and effective solution for legal document analysis.

4.2 Technical Feasibility

The feasibility of deploying this AI-enabled legal document analytical system depends on various technical parameters like computational resources and infrastructure.

Computational Resources & Infrastructure
The architecture follows transformer models such as Legal-BERT and GPT-4, which use enormous computational power. While local deployment can be done for testing, large-scale processing and real-time inference are only possible using cloud-based infrastructure (e.g., AWS, Google Cloud, or Azure) with GPU or TPU for model training and execution.

System Architecture & Data Management
A hybrid storage strategy is a requirement for efficient document retrieval and processing:

- Vector Database (Pinecone, Weaviate, FAISS): Stores document embeddings for semantic search.
- Knowledge Graph Database (Neo4j, Stardog): Structures legal entities and relations.
- Relational Database (PostgreSQL): Manages user data, admin tasks, and document indexing.

The backend, implemented with Django, must facilitate seamless communication between these parts through properly optimized API endpoints.

Integration & Performance Optimization
For the best system performance, there should be interoperability between the NLP models, database, and frontend. Minimization of API calls, implementation of caching mechanisms, and latency reduction are crucial to handle large-scale legal datasets and provide real-time search results.

Security & Compliance
With the sensitive nature of legal information, strict security measures need to be put in place:

- Data Protection: Encryption of data stored, and data transmitted.
- Access Control: Role-Based Access Control (RBAC) for managing user permissions.
- Authentication & Logging: Secure authentication (OAuth, JWT) and audit logging to track data access and updates.

Compliance with UK law regulations, including UK-GDPR and the Data Protection Act 2018, is crucial to ensure that the system complies with industry best practices for privacy and security.

By implementing scalable cloud resources, effective data handling, and robust security controls, the technical viability of the system is well-supported, ensuring its sustainability for real-world legal use.

4.3 Budget Outline (Finances)

The project is designed to leverage predominantly free and open-source resources, resulting in minimal financial investment. The estimated budget breakdown is as follows:

a) Hardware Costs –

Expected to be zero, as all the required hardware materials are already in possession.

b) Software and Development Tools –

Python and required libraries including HuggingFace tools – Free and Open Source
Neo4j (for Knowledge Graphing) – Free version available
Weaviate or Pinecone (for Vector Databases) – Free tiers available

c) Data Acquisition and Preprocessing –

Public Legal Datasets – Gain access to cybercrime case datasets through legal institutions (BAILII). Already obtained access to the Computer Misuse Act and other relevant legislation documents.

Preprocessing costs – Automated cleaning of data using the latest technologies which will most likely be free of charge.

d) Cloud Computing Costs –

Cloud Services – Services like Google Cloud, Microsoft Azure or AWS offer free tiers, but some additional payment will be required based on resources need for model training and time. (Approx. £50 - £150)
However, as an alternative Google Colab Pro can be used which offers more computing and fewer restrictions. (Approx. £9/month)

e) Development and Research Resources –

Journal Subscriptions or Articles – Most research articles are available through the university account and through the university library.

f) Miscellaneous Costs –

API Access Fees – If paid APIs are needed or if the free limit surpasses that of the free APIs, additional payment will be needed. (Approx. £20 - £100)

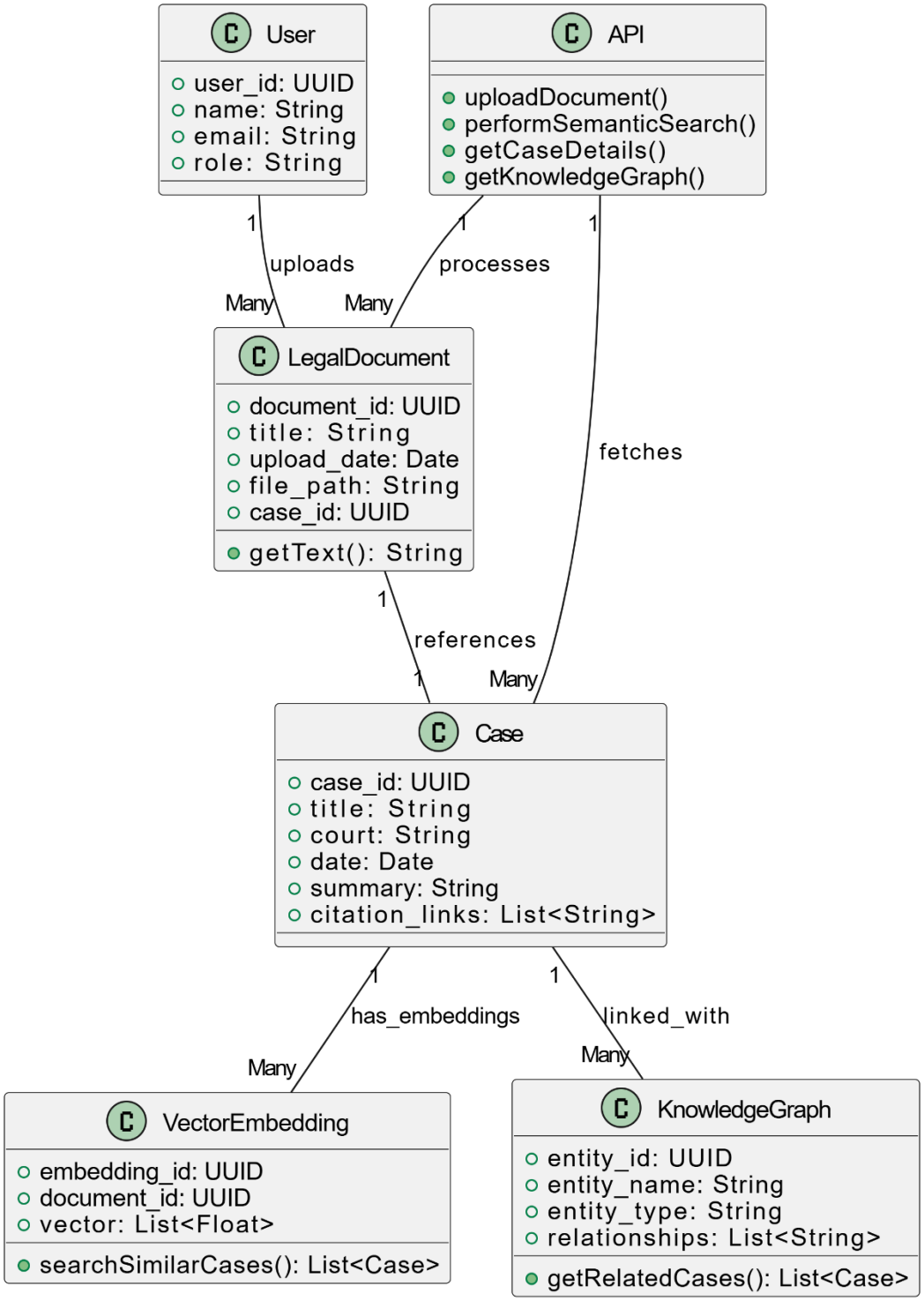
External Consultation Fees & Other Stuff – As this project concerns the legal sector, consultation of legal professionals will be needed in order to create a system that will be accepted by the public. Also, the system being endorsed by a legal practitioner will be a huge addition and will really separate this one from the existing general tools. That being said, this is hoped to be achieved through a legal professional that is personally known, that would mean it will most likely not be an expense. However, taking everything into account an Approx. of £30 - £100 seems most likely.

Category	Estimated Cost (£)
Hardware	0
Software	0
Data Acquisition	0
Cloud Computing Services	0 – 100
Development and Research	0
Miscellaneous	0 – 100
Total Estimated Costs	0 – 200

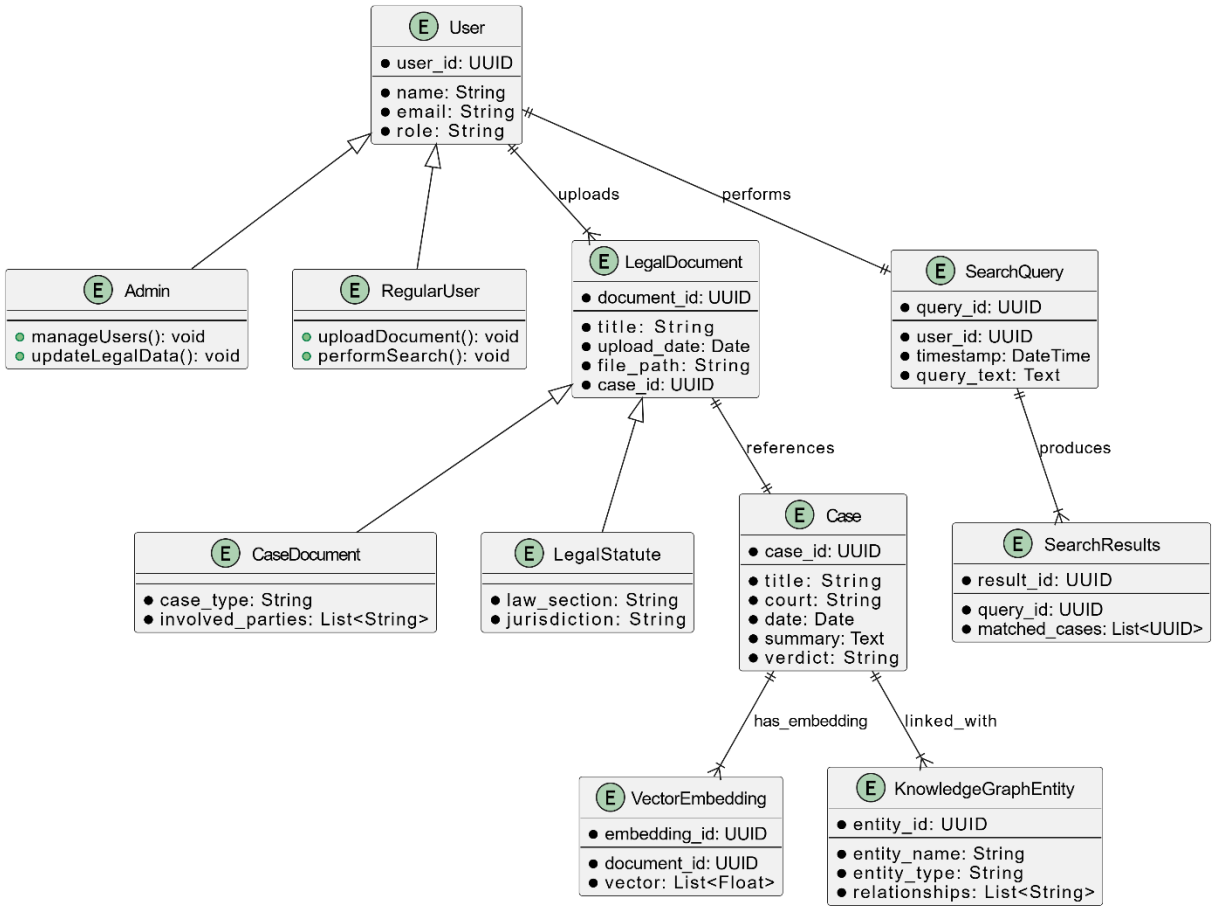
The total estimated costs for completing this are estimated to fall between £0 - £200, depending on specific cloud computing service needs, and potential API usage fees. A completely functional system can be created with minimal financial investment by using free tiers of widely accessible technologies (such as cloud services, pre-trained models, and public datasets).

5. System Architecture

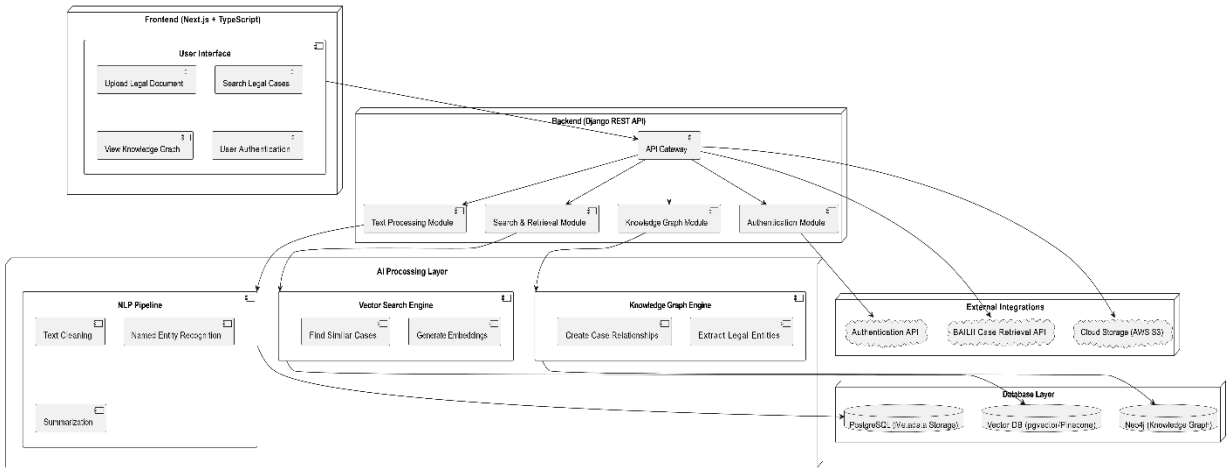
5.1 Class Diagram of Proposed System



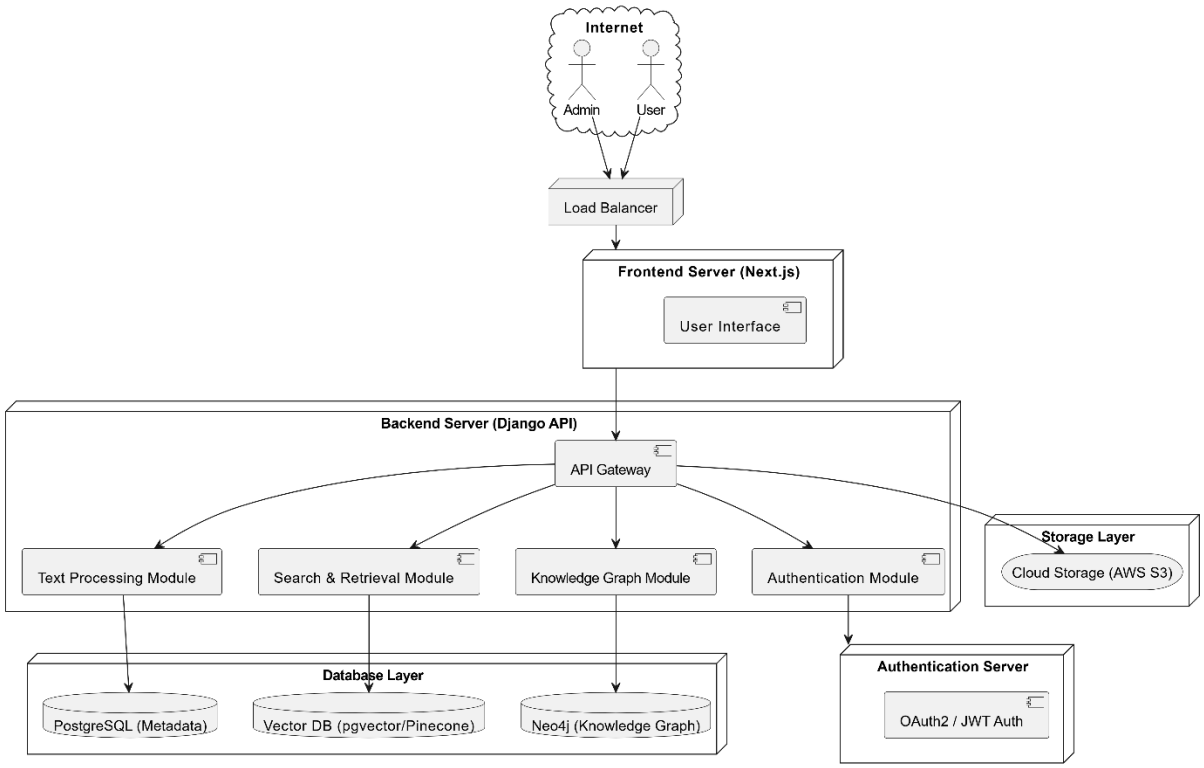
5.2 Enhanced Entity Relationship Diagram (EER)



5.3 High Level Architectural Diagram

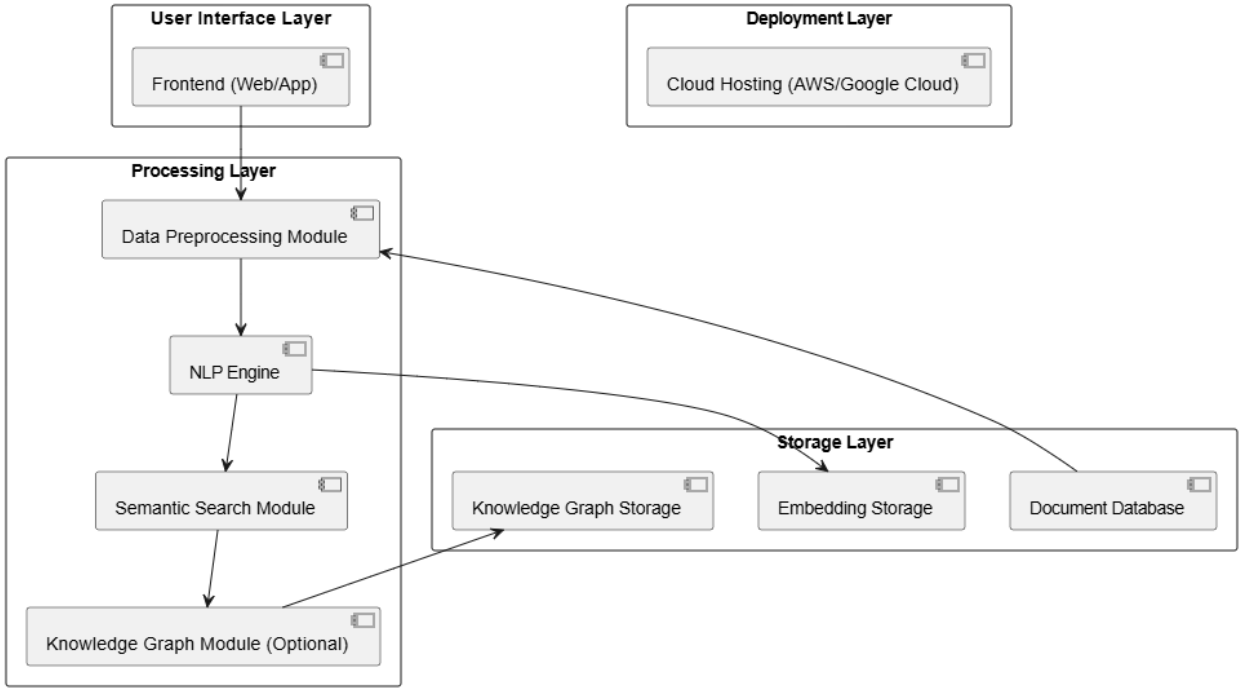


5.4 Networking Diagram



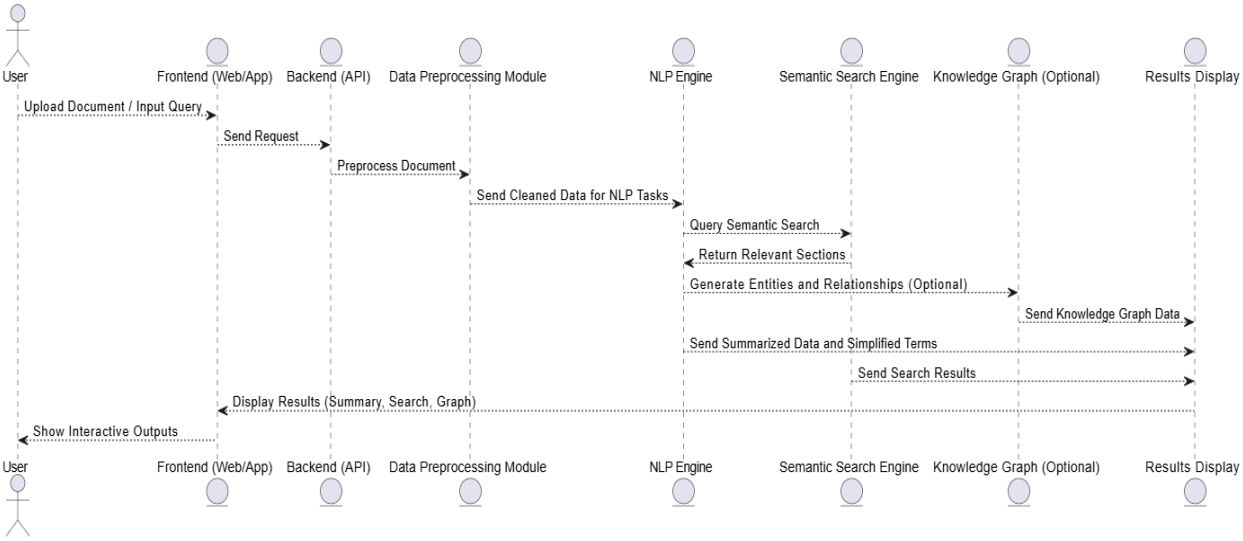
5.4 Other Diagrams

Layered Architecture Diagram



Divides the system into layers (Ex- User Interface, Processing, Storage, Deployment) with components and data flow within each layer, this diagram shows the top-down view of the architecture.

Conceptual Diagram



Workflow Diagram

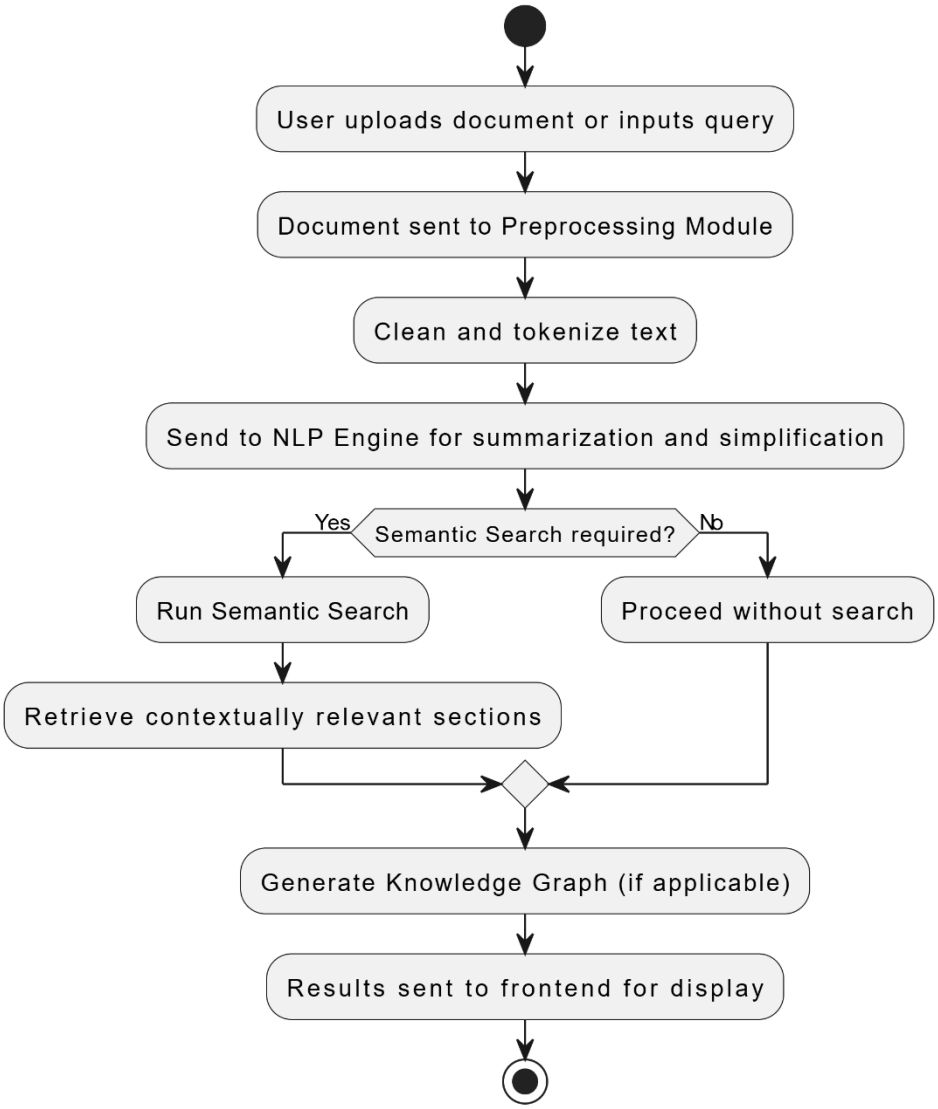
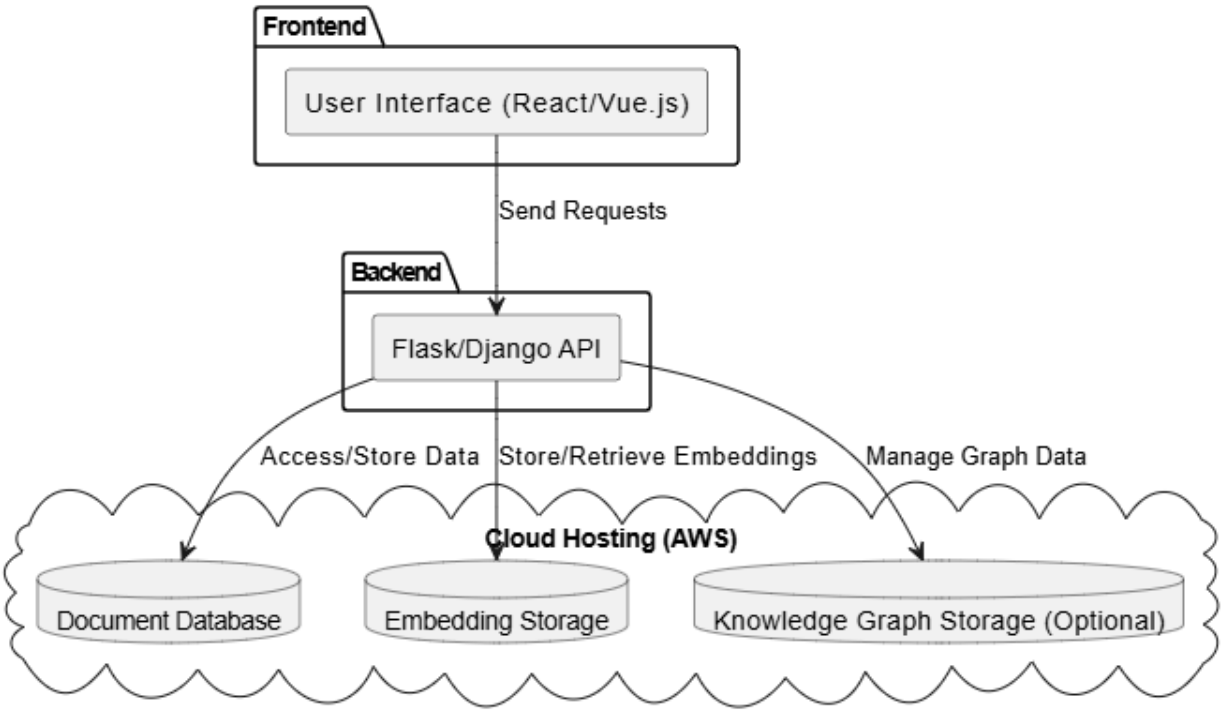


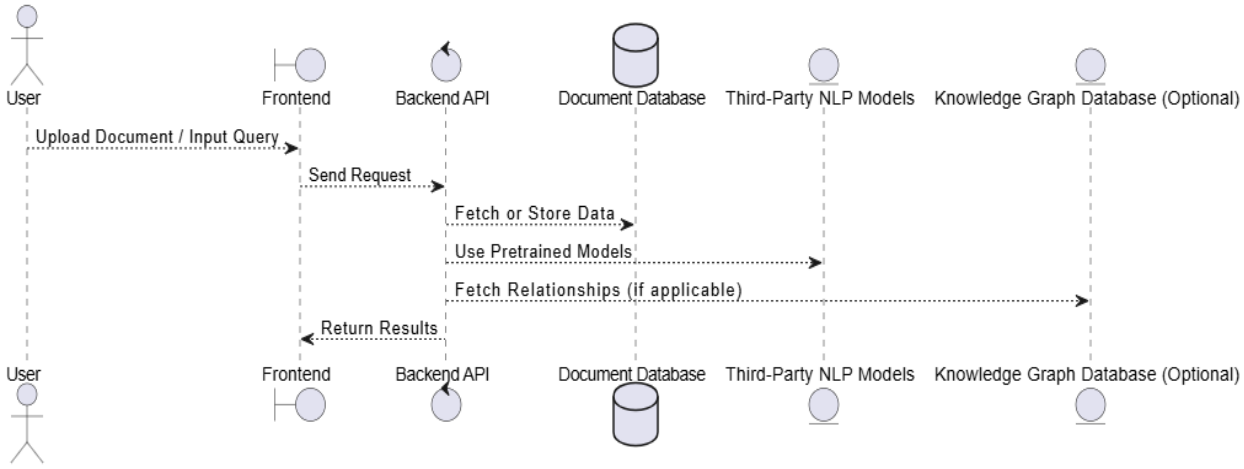
Diagram to show the sequential flow of data and tasks from input (user) to output (results).

Deployment Diagram



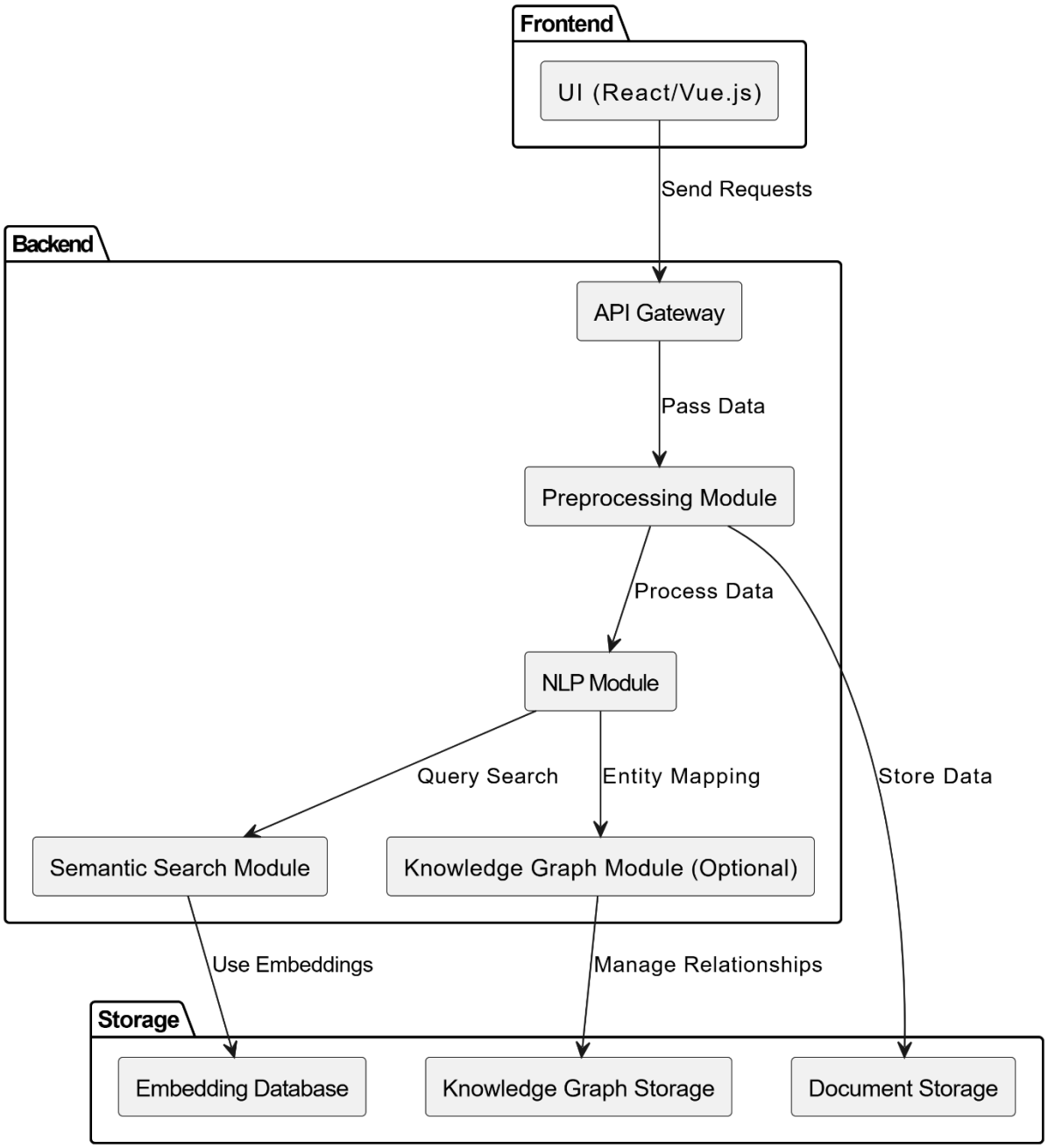
Displays how components are distributed across servers, clouds, or other infrastructure. Includes databases, APIs, and user-facing applications

System Context Diagram



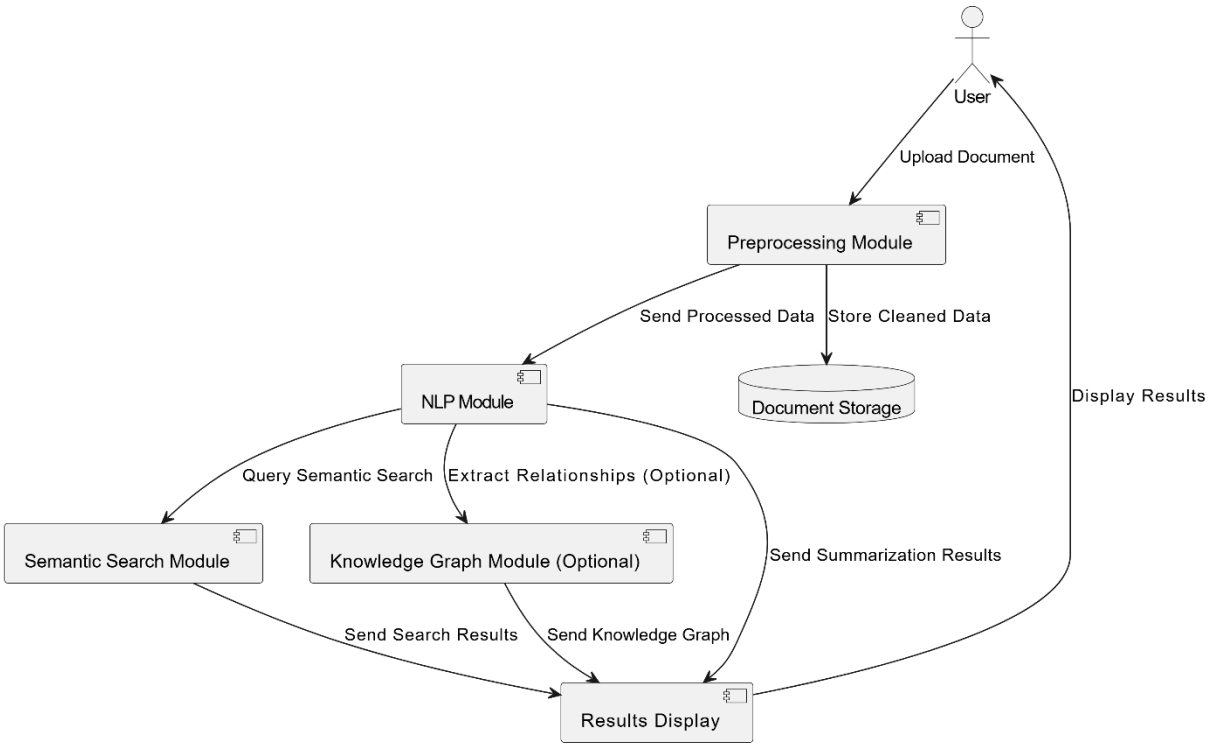
How the system interacts with external entities such as users, databases, third-party APIs are highlighted from this diagram.

Component Diagram



This diagram shows how the key functional modules interact with each other.

Data Flow Diagram (DFD)



Tracks the movement of data within the system, from input sources to processing modules and storage.

PS – Concepts / Frameworks demonstrated by these figures may be subjected to change.

6. Development Tools and Technologies

6.1 Development Methodology

Development Methodology defines the methodology followed during designing, building, and deploying this AI-fueled legal document analysis tool. Given the complexity of integrating NLP, knowledge graphs, and vector-based semantic search, a systematic and iterative approach must be followed to provide efficiency, flexibility, and accuracy.

Agile Development Approach

Therefore, this project takes an Agile Development Methodology, a Scrum-based iterative approach. Agile is chosen because it can handle changing requirements, continuous feedback, and the cycles of repeated improvement efficiently. This methodology ensures that the system is progressively developed on the basis of testing, evaluation, and feedback from the users.

Major Agile Principles Applied:

Incremental Development – The system is incrementally developed with functional components being built and tested in iterations.
Continuous Feedback – Regular testing and verification against legal databases guarantee performance and compliance.
Adaptability – Allows for changes according to experience gained from case studies, expert opinions, and real-world experiments.
Collaboration – Involves feedback from legal experts to refine accuracy and usability of the system.

System Development Stages
Development is divided into distinct stages to ensure an organized and efficient workflow.

Phase 1: Requirement Analysis & Planning

- Identify functional and non-functional requirements.
- Define the key components: document processing, NLP, knowledge graph, and semantic search.
- Assess feasibility in terms of computational resources and lawfulness.
- Select appropriate technologies (Neo4j, Pinecone/Weaviate, Hugging Face models, etc.).

Phase 2: Data Collection & Preprocessing

- Gather cybercrime case data from BAILII and other judicial sources.
- Clean and preprocess documents (e.g., removing extraneous text, structuring data).
- Generate embeddings for semantic search using models like Legal-BERT.

- Identify significant legal entities and relationships for constructing knowledge graphs.

Phase 3: Model Development & Fine-Tuning
- Use NLP pipelines for legal document summarization.
- Fine-tune Legal-BERT and other pre-trained models for legal text processing.
- Develop a named entity recognition (NER) model to recognize legal terms, case names, and significant parties.

Phase 4: Knowledge Graph & Vector Search Implementation
- Build a Neo4j-based knowledge graph linking legal cases, statutes, and significant entities.
- Use Weaviate or Pinecone for vector search to improve legal document retrieval.
- Enable hybrid search (semantic search and knowledge graphs integration).

Phase 5: System Integration & Backend Development
- Develop the backend using Django, integrated with Neo4j and PostgreSQL.
- Integrate APIs for document upload, semantic search, and knowledge graph operations.
- Secure user data handling (encryption, access control).

Phase 6: Frontend Development
- Enforce a Next.js-based frontend for simplicity in user interaction.
- Develop a dashboard for document analysis, knowledge graph visualization, and case recovery.
- UI/UX optimization for easy navigation and usability.

Phase 7: Testing & Evaluation
- Perform unit testing, integration testing, and user acceptance testing.
- Validate system accuracy with legal thresholds and expert rating.
- Tune NLP and vector search models according to performance criteria.

Phase 8: Deployment & Finalization
- Deploy the system using cloud services (AWS, Azure, or Google Cloud).
- Make final performance and security tuning.
- Document system architecture, API endpoints, and user manuals.

Justification for Agile Methodology
Agile offers rapid, iterative development with constant refinement based on real-world testing and specialist advice. Because of technical and legal complexity, this method prevents strict forms that can hinder system improvement. Additionally, Agile's incremental delivery emphasis guarantees that core functionalities such as NLP analysis, knowledge graphs, and vector search are tested and validated incrementally.

By this method, the project ensures a robust, scalable, and legally compliant AI-driven legal document analysis system.

6.2 Programming Languages and Tools

The development of the AI-based legal document analysis system relies on a combination of **programming languages, frameworks, and tools** to ensure efficient processing, storage, and retrieval of legal documents. Each component is chosen based on its suitability for NLP, knowledge graph construction, and web development.

Primary Programming Languages

- **Python** – The core language for NLP processing, machine learning model integration, and backend development (Django).
- **JavaScript (TypeScript)** – Used for the frontend, built with **Next.js**, ensuring a modern and efficient user interface.
- **Cypher Query Language** – Used to query and manipulate legal data stored in the **Neo4j** knowledge graph.
- **SQL (PostgreSQL)** – For structured data storage, such as user management and administrative features.

Development Tools & Platforms

- **Jupyter Notebook / Google Colab** – For model fine-tuning, testing, and data preprocessing.
- **Postman** – For API testing and validation.
- **Docker** – For containerized deployment, ensuring compatibility across different environments.
- **GitHub** – For version control and collaborative development.

By leveraging these languages and tools, the system ensures seamless integration of NLP models, database management, and a user-friendly interface.

6.3 Third-Party Components and Libraries

To enhance efficiency and accuracy, the system integrates several third-party libraries and frameworks specialized in NLP, knowledge graphs, and vector search. These components enable robust text processing, entity recognition, and efficient search mechanisms.

Natural Language Processing (NLP) Libraries

- **spaCy** – Used for tokenization of text, part-of-speech (POS) tagging, and Named Entity Recognition (NER).
- **NLTK (Natural Language Toolkit)** – Facilitates in basic text processing and legal text structuring.
- **Hugging Face Transformers** – Provides pre-trained models like **Legal-BERT**, fine-tuned for the analysis of legal documents.

Vector Search & Embedding Libraries

- **Sentence-Transformers** – Used for encoding legal texts into vector embeddings for **semantic search**.
- **FAISS (Facebook AI Similarity Search)** – An alternative for high-speed nearest-neighbor search.
- **Weaviate / Pinecone** – Manages and indexes document embeddings for real-time case retrieval.

Knowledge Graph Components

- **Neo4j** – Stores and visualizes legal case relationships through graphs, enabling **relation-based** querying.
- **NetworkX** – Utilized for graph-based computations and analysis.

Web Development Frameworks

- **Django (Python)** – Manages backend services, API endpoints, and authentication.
- **Next.js (React/TypeScript)** – Handles the frontend, providing an intuitive UI for legal document analysis.
- **FastAPI** – For smooth API communication among components.

Cloud & Storage Services

- **Google Cloud / AWS / Azure** – Provides computational power for fine-tuning of models.
- **Google Drive API / AWS S3** – For storing user-uploaded legal documents.

These third-party components significantly enhance the system's efficiency in legal text processing, knowledge representation, and user interaction.

(PS – These were already mentioned in this report and older documents repeatedly, refer chapter 4)

6.4 Algorithms

This AI-based legal document analysis system is developed using multiple algorithms to assist with text processing, semantic searching, and constructing a knowledge graph. These algorithms ensure high precision and accuracy in retrieving, summarizing, and structuring legal documents.

1. Document Preprocessing Algorithm

- **Text Cleaning & Tokenization**:
  - Remove unnecessary formatting, headers, and citations.
  - Tokenize sentences and words using **spaCy**.
- **Stop-word Removal & Lemmatization**:
  - Remove irrelevant words to enhance semantic search accuracy.
  - Convert words to their base form (lemmatization).

2. Named Entity Recognition (NER) Algorithm

- **Custom-trained NER model (using Hugging Face Transformers) detects legal entities**, such as case names, legal references, and key actors.
- Uses **Conditional Random Fields (CRF)** and **Transformer-based models (Legal-BERT)** for optimal accuracy.

3. Semantic Search Algorithm

- **Embedding Generation**:
  - Convert legal text to **vector embeddings** using **Sentence-Transformers**.
- **Nearest Neighbor Search**:
  - Uses **FAISS / Weaviate** for fast similarity matching between query documents and stored legal cases.

4. Knowledge Graph Construction Algorithm

- **Entity-Relationship Mapping**:
  - Extracts entities from legal documents and connects them based on case relationships.
  - Uses **Neo4j's Cypher Query Language** to represent structured knowledge representation.

5. Legal Document Summarization Algorithm

- **Abstractive Summarization (T5/BART Models)**:
  - Generates concise legal summaries while preserving critical case details.
- **Extractive Summarization (TextRank Algorithm)**:
  - Select key sentences from the document for a **fact-based summary**.

6. Hybrid Querying Algorithm

- **Combines vector search + knowledge graph queries (also keyword search)** to improve accuracy in retrieving legal cases.
- **Process:**
    - User query → Vector similarity search (Pinecone) → Retrieve related legal cases (Neo4j).
    - Apply **hybrid reranking** based on **semantic similarity + graph-based relevance scoring**.

With these advanced algorithms, the system guarantees accurate legal document retrieval, knowledge graph construction, and NLP-based summarization, and hence is a resilient AI-driven solution for legal professionals.

7. Discussion

7.1 Overview of the Interim Report

The report first establishes the problem definition and objectives, focusing on legal document analysis, notably in UK cybercrime law. It then establishes system requirements such as document management, NLP-based summarization, semantic search, and knowledge graph construction. The feasibility studies guarantee that the project is viable in operational, technical, and financial aspects. The system development process, programming languages, third-party tools, and algorithms used in the system are also specified, illustrating how various parts unite to form an integrated solution.

7.2 Summary of the Report

This paper outlines the development of an AI-driven legal document analysis system for UK cybercrime law. The project addresses inefficiencies of traditional legal research through preprocessing, summarization, semantic search, and constructing interactive knowledge graphs automatically. Leveraging state-of-the-art NLP models like Legal-BERT and GPT-4 and vector databases and Neo4j for relationship mapping, the system promises improved accuracy and faster insights. The agile development paradigm, coupled with robust technical and operational feasibility studies, emphasizes the project's feasibility using predominantly free, open-source tools and flexible cloud infrastructure. Retrieving structured legal data and tuning models to recognize domain-specific nuances are primary challenges. Future work will focus on refining model performance, optimizing the user interface, and expanding system capabilities through further integration with legal databases and APIs.

7.3 Challenges Faced

Several challenges have been encountered along the way while developing this project. One of the most important challenges is having organized cybercrime case data because legal cases are typically scattered in diverse sources. Another challenge is optimizing NLP models for legal text, which remains complicated by domain language and the need for high accuracy in summarization and entity recognition. Another challenge is to optimize the integration of the knowledge graph and vector search for effective hybrid querying. Resource limitations, particularly cloud computing costs and model training, also pose constraints that need to be handled with care.

7.4 Future Plans / Upcoming Work

The next phase of the project involves integrating pre-trained models, improving semantic search accuracy, and implementing knowledge graphs. Additionally, the UI will be refined and finalized to improve usability. Deployment and testing will be conducted to ensure the system meets performance and reliability expectations. Future work will also explore potential integrations with legal databases or APIs to expand case coverage and further validate the system’s effectiveness in real-world scenarios.

8. Progress and Development Review

8.1 Tasks Undertaken and Outcomes

During the project, several of the major activities have been carried out to lay a foundation for the AI-based legal document analysis system. Firstly, data acquisition was the main priority where efforts were made to obtain legal case documents for analysis. The initial aim was to obtain cybercrime-related case documents under Sri Lankan law utilizing the Computer Crime Act (2007). However, this direction was confronted with significant challenges due to limited availability of digital records of legal cases. Therefore, a strategic choice was made to shift the legal area of the project to UK cybercrime law, specifically cases under the UK Computer Misuse Act (1990), due to higher data availability and accessibility through sources such as BAILII.

To facilitate this collection of data, HTML scraping scripts were written to fetch case titles and links from BAILII. Not every case title, however, included case texts on the internet, so manual intervention was required for missing records. After metadata extraction (titles, dates, courts), the next milestone was to download and store full case texts, which have now been organized in a PostgreSQL database for ease of retrieval. Additionally, Neo4j has been set up for knowledge graph representation of case relationships.

Another major task involved structuring / rethinking of the project's architecture. The initial idea of using multiple databases was refined to focus on Neo4j for legal relationships and PostgreSQL for backend metadata storage while leveraging vector embeddings to enhance semantic search. A Next.js frontend has also been initialized to provide an intuitive interface for users.

Key Outcomes Achieved:

- **Legal domain adjusted** → Shifted focus from Sri Lankan to UK law for better data availability.
- **Case retrieval & storage set up** → Titles, metadata, and case texts now stored in PostgreSQL.
- **Graph representation initiated** → Legal entities and case relationships structured in Neo4j.
- **Backend & API groundwork laid** → Django backend with endpoints for document processing.
- **Frontend environment initialized** → Next.js with a structured project setup. (Development Halfway)

8.2 Products Produced and Product Quality

Product	Status	Quality Considerations
Data Retrieval System	Completed	Scraping and automated downloads with manual verification for missing records.
Database Schema (PostgreSQL & Neo4j)	In - Progress	Structured for scalability and optimized for legal data.
Backend APII (Django and FastAPI)	Initial Setup Done	Planned improvements for API security & error handling.
Frontend UI (Next.js)	Developed Halfway	Initial components in place (Landing page etc.), awaiting API integration.
Knowledge graph	In - Progress	Initial entity mapping complete, next step: visual integration.

Product Quality Considerations:
The initial versions of the components have been tested for **basic functionality**, with future refinements planned for **API security, efficient query processing, and UI responsiveness**. Data integrity remains a primary concern, especially in **legal case representation**, requiring **validation checks and structured formatting** for accurate case mapping. **Knowledge Graph System** is still under initial development phase.

8.3 Risks that Have Materialized and Responses

Risks Encountered	Impact Assessment	Response and Mitigation
Limited access to Sri Lankan legal case data	High	Shifted focus to **UK cybercrime cases (Computer Misuse Act 1990)** for better data availability.
Incomplete legal case texts on BAILII	Medium	Logged missing records for manual review and download.
Uncertainty regarding knowledge graph feasibility	High	Proceeding with **Neo4j for case relationships**, with further testing required for user interaction.
Model training complexity (semantic search & summarization)	Medium	Fine-tuning domain-specific **NLP models on UK legal texts** is planned but may require additional pre-trained legal datasets.

Risk List Updates:

- Data availability risk has shifted from accessibility concerns to complete verification.
- The knowledge graph remains under evaluation as we determine its practical user applications.
- Potential model training constraints may require additional external datasets or embeddings.

8.4 Schedule Progress: Planned vs Actual Progress

The original schedule accounted for legal data retrieval and processing within the first month. However, the unexpected challenge of Sri Lankan legal data scarcity led to delays in dataset acquisition. Switching to UK law required additional time, pushing other tasks back by approximately two weeks. Despite this, progress in backend API and frontend UI setup has remained on track, ensuring minimal overall delays.

Corrective Actions Taken:

- **Parallel development tasks introduced** → While legal data processing faced delays, backend and UI development continued.
- **Sprint adjustments** → The initial delay has been offset by prioritizing backend integrations to maintain momentum.

8.5 Needed and Acquired Learning Outcomes

The project has necessitated a deep understanding of **AI-driven legal text processing**, requiring proficiency in multiple areas:

Technical Learning Outcomes:

- **Database Optimization** - Efficient data structuring in PostgreSQL & Neo4j.
- **Legal NLP Models** - Exploring BERT-based models (LegalBERT, CaseLawBERT) for domain-specific processing.
- **Semantic Search Implementation** - Understanding vector embeddings and similarity search techniques.

Project Management & Research Outcomes:

- **Agile Workflow Adjustments** - Managing task dependencies amid changing requirements.
- **Legal Data Handling Ethics** - Understanding privacy policies and open-access legal databases.
- **Risk Management** - Adapting to unforeseen challenges in legal document access & processing.

Future Learning Requirements:

- Fine-tuning NLP models for summarization & legal search.
- Integrating advanced graph-based legal insights using Neo4j.
- Enhancing frontend UX for intuitive legal document interaction.

8.6 Updated Final Deliverables

Core Deliverables (Confirmed):

- Full AI-Driven Legal Document Processing System (with document upload, search, summarization).
- Neo4j-Based Legal Case Knowledge Graph. (Optional)
- Semantic Search Engine for Legal Case Retrieval.
- API & Frontend Interface for Public Access.

Potential Additions (Pending Feasibility Testing):

- Pre-trained Legal NLP Model Integration for Higher Accuracy.
- Graph-Based Legal Question Answering System.
- Automated Statutory Reference Extraction from Case Texts.

9. References

Chalkidis, I. *et al.* (2020) 'LEGAL-BERT: The Muppets straight out of Law School'. Available at: http://arxiv.org/abs/2010.02559.

Dragoni, M. *et al.* (2016) *Combining NLP Approaches for Rule Extraction from Legal Documents Combining NLP Approaches for Rule Extraction from Legal Documents. 1st Workshop on MIning and REasoning with Legal texts Combining NLP Approaches for Rule Extraction from Legal Documents*. Available at: https://wordnet.princeton.edu/.

Imogen, P.V., Sreenidhi, J. and Nivedha, V. (2024) 'AI-Powered Legal Documentation Assistant', *Journal of Artificial Intelligence and Capsule Networks*, 6(2), pp. 210–226. Available at: https://doi.org/10.36548/jaicn.2024.2.007.

Merchant, K. and Pande, Y. (2018) *NLP Based Latent Semantic Analysis for Legal Text Summarization*. IEEE.

Sabrina Univ-Prof Axel P, A.K. (2021) *Knowledge Graphs for Analyzing and Searching Legal Data*.

Sachidananda, V., Kessler, J.S. and Lai, Y. (2021) 'Efficient Domain Adaptation of Language Models via Adaptive Tokenization'. Available at: http://arxiv.org/abs/2109.07460.

Vayadande, K. *et al.* (2024) 'AI-Powered Legal Documentation Assistant', in *Proceedings - 2024 4th International Conference on Pervasive Computing and Social Networking, ICPCSN 2024*. Institute of Electrical and Electronics Engineers Inc., pp. 84–91. Available at: https://doi.org/10.1109/ICPCSN62568.2024.00022.

Yang, W. *et al.* (2019) 'Legal Judgment Prediction via Multi-Perspective Bi-Feedback Network'. Available at: https://doi.org/10.24963/ijcai.2019/567.

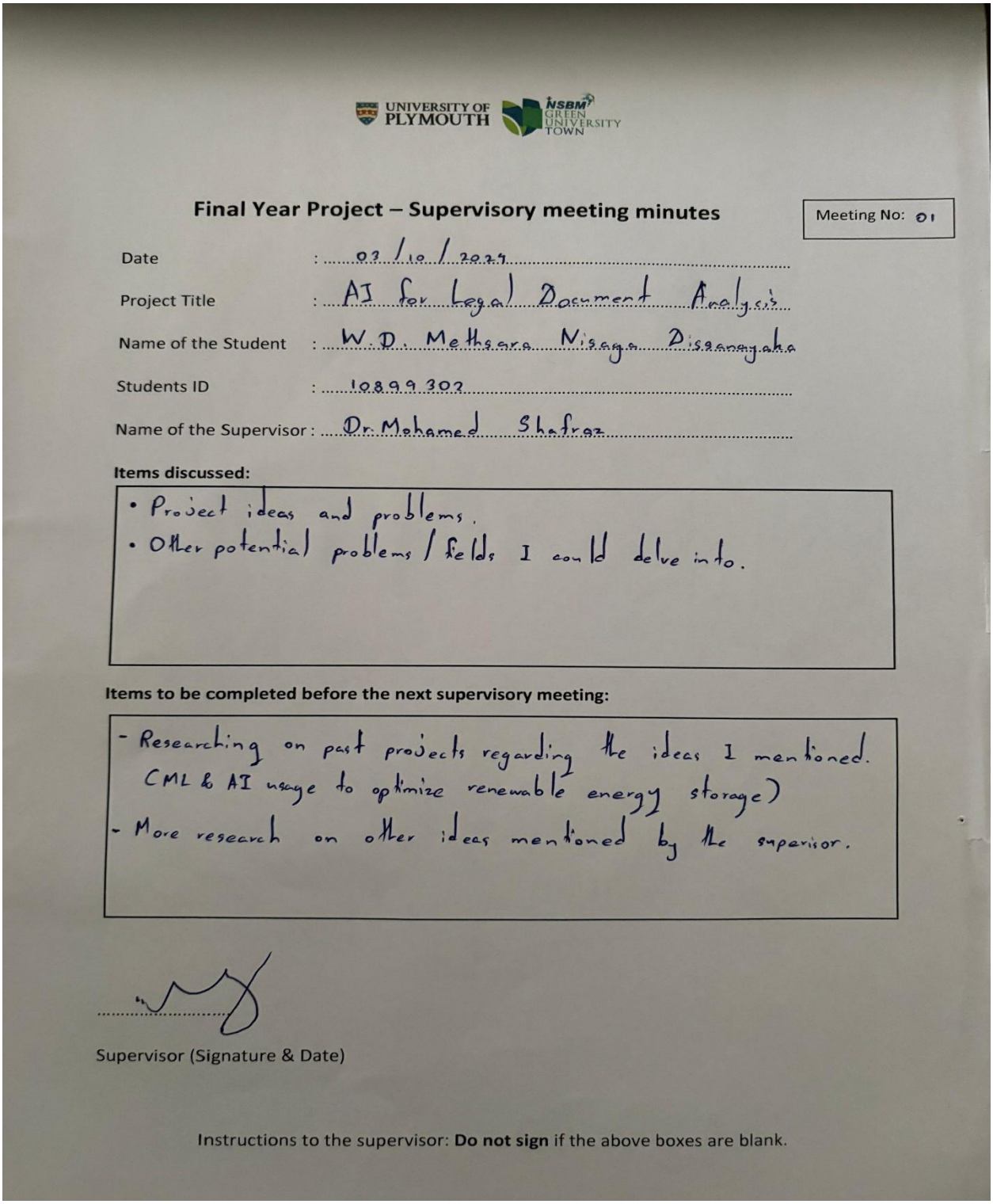
Zakir, M.H. *et al.* (2024) 'Artificial Intelligence and Machine Learning in Legal Research: A Comprehensive Analysis', *Qlantic Journal of Social Sciences*, 5(1), pp. 307–317. Available at: https://doi.org/10.55737/qjss.203679344.

Zheng, J. *et al.* (2024) 'Fine-tuning Large Language Models for Domain-specific Machine Translation'. Available at: http://arxiv.org/abs/2402.15061.

Zhong, H. *et al.* (2020) 'How Does NLP Benefit Legal System: A Summary of Legal Artificial Intelligence'. Available at: http://arxiv.org/abs/2004.12158.

THANK YOU!

11.4 Meeting Minutes



UNIVERSITY OF PLYMOUTH | NSBM GREEN UNIVERSITY TOWN

Final Year Project – Supervisory meeting minutes

Meeting No: 01

Date : 03 / 10 / 2024

Project Title : AI for Legal Document Analysis

Name of the Student : W.D. Methsara Nisaga Disanayaka

Students ID : 10899302

Name of the Supervisor : Dr. Mohamed Shafraz

Items discussed:

- Project ideas and problems.
- Other potential problems / fields I could delve into.

Items to be completed before the next supervisory meeting:

- Researching on past projects regarding the ideas I mentioned. (ML & AI usage to optimize renewable energy storage)
- More research on other ideas mentioned by the supervisor.

Supervisor (Signature & Date)

Instructions to the supervisor: **Do not sign** if the above boxes are blank.

UNIVERSITY OF PLYMOUTH | NSBM GREEN UNIVERSITY TOWN

Final Year Project – Supervisory meeting minutes

Meeting No: 02

Date : 10/10/2024

Project Title : AI for Legal Document Analysis

Name of the Student : W.D. Methsara Nisaya Dissanayaka

Students ID : 10899302

Name of the Supervisor : Dr. Mohamed Shafraz

Items discussed:

- Finalising project idea and scope.
- Deciding on the functions that would be needed.

Items to be completed before the next supervisory meeting:

⇒ Work on project proposal

⇒ Conduct literature review

10/10/24

Supervisor (Signature & Date)

Instructions to the supervisor: **Do not sign** if the above boxes are blank.

UNIVERSITY OF PLYMOUTH | NSBM GREEN UNIVERSITY TOWN

Final Year Project – Supervisory meeting minutes

Meeting No: 03

Date : 06/12/2024

Project Title : AI for Legal Document Analysis

Name of the Student : W.D. Methsara Nisaga Dissanayaka

Students ID : 10899302

Name of the Supervisor : Dr. Mohamed Shafraz

Items discussed:

- Addressing the current issues regarding knowledge graphs.
- Talked about the progress regarding the implementations current features.

Items to be completed before the next supervisory meeting:

→ Work on PID & complete it
→ start designing the solution

6/12/24

Supervisor (Signature & Date)

Instructions to the supervisor: **Do not sign** if the above boxes are blank.

UNIVERSITY OF PLYMOUTH | NSBM GREEN UNIVERSITY TOWN

Final Year Project – Supervisory meeting minutes

Meeting No: 04

Date : 01/04/2025

Project Title : AI - Powered Legal Document Assistant

Name of the Student : W.D. Methsara Nisaga Dissanayaka

Students ID : 10899202

Name of the Supervisor : Dr. Mohamed Shafraz

Items discussed:

- Managing multiple databases and backend structure.
- ~~finished the fro~~
- Approval for current UI and backend features.

Items to be completed before the next supervisory meeting:

→ ~~Final~~ Refer to the final report structure and complete the document.

→ Implement the system with all agreed features.

01/04/2025

Supervisor (Signature & Date)

Instructions to the supervisor: **Do not sign** if the above boxes are blank.

UNIVERSITY OF PLYMOUTH

NSBM GREEN UNIVERSITY TOWN

PUSL3190 Computing Individual Project
Student Progression Report
[Student Copy]

01. Student Name ..W.D. Methsara Nisaga Dissanayaka.....
02. Plymouth Index Number ..10899302.....
03. Degree Program ..BSc. (Hons) Data Science.....
04. Supervisor Name ..Dr. Mohamed Shafraz.....
05. Project Title ..AI for Legal Document Analysis.....

Meeting Number	Meeting 01	Meeting 02	Meeting 03	Meeting 04	Meeting 05	Meeting 06	Meeting 07
Date	03/10/24	10/10/24	06/12/24	01/04/25			
Student Signature	Methsara.	Methsara.	Methsara.	Methsara.			
Supervisor Signature	M.	M.	M.	M.			

Meeting Number	Meeting 08	Meeting 09	Meeting 10	Meeting 11	Meeting 12	Meeting 13	Meeting 14
Date							
Student Signature							
Supervisor Signature							

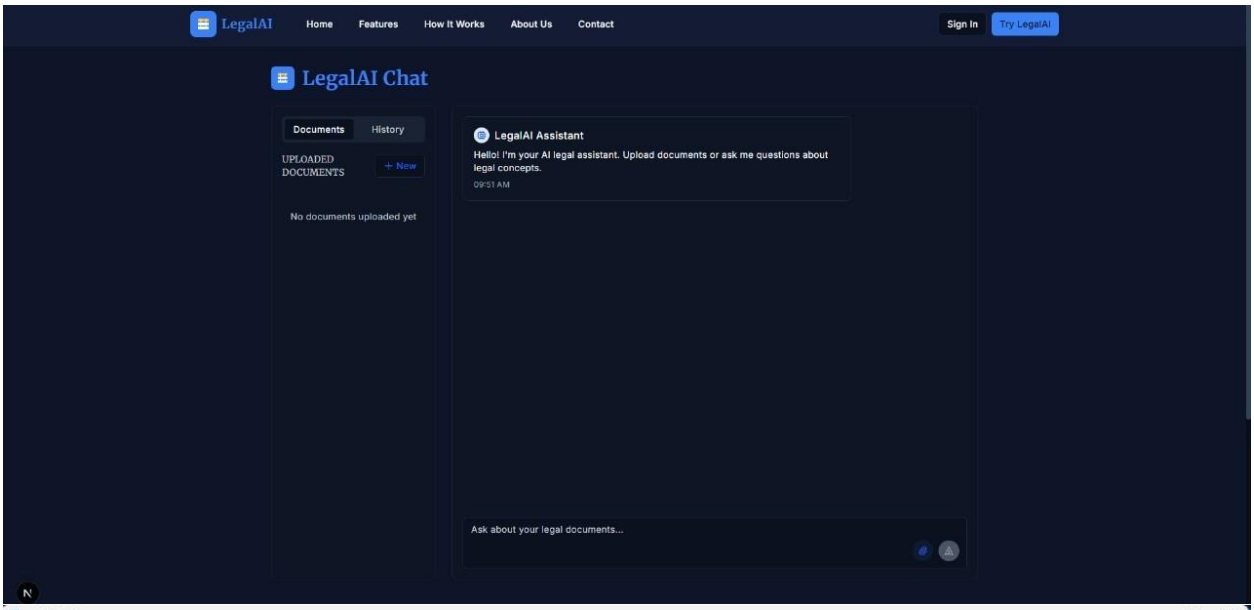
UNIVERSITY OF PLYMOUTH

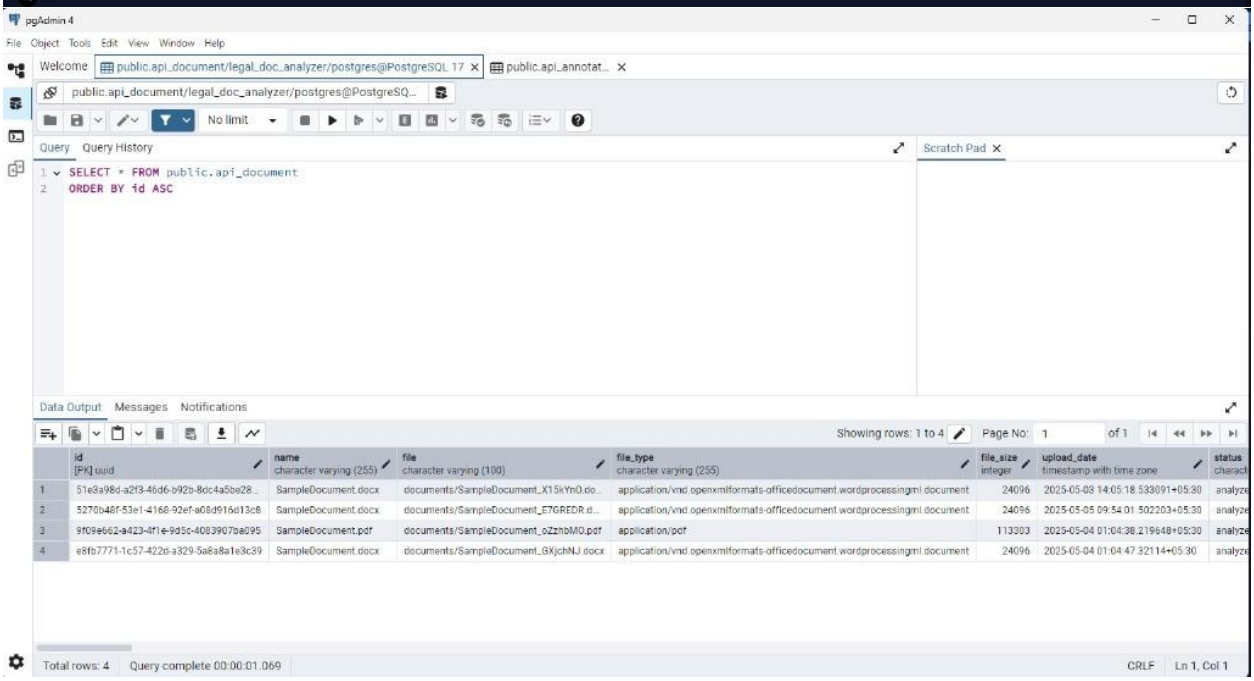
NSBM GREEN UNIVERSITY TOWN

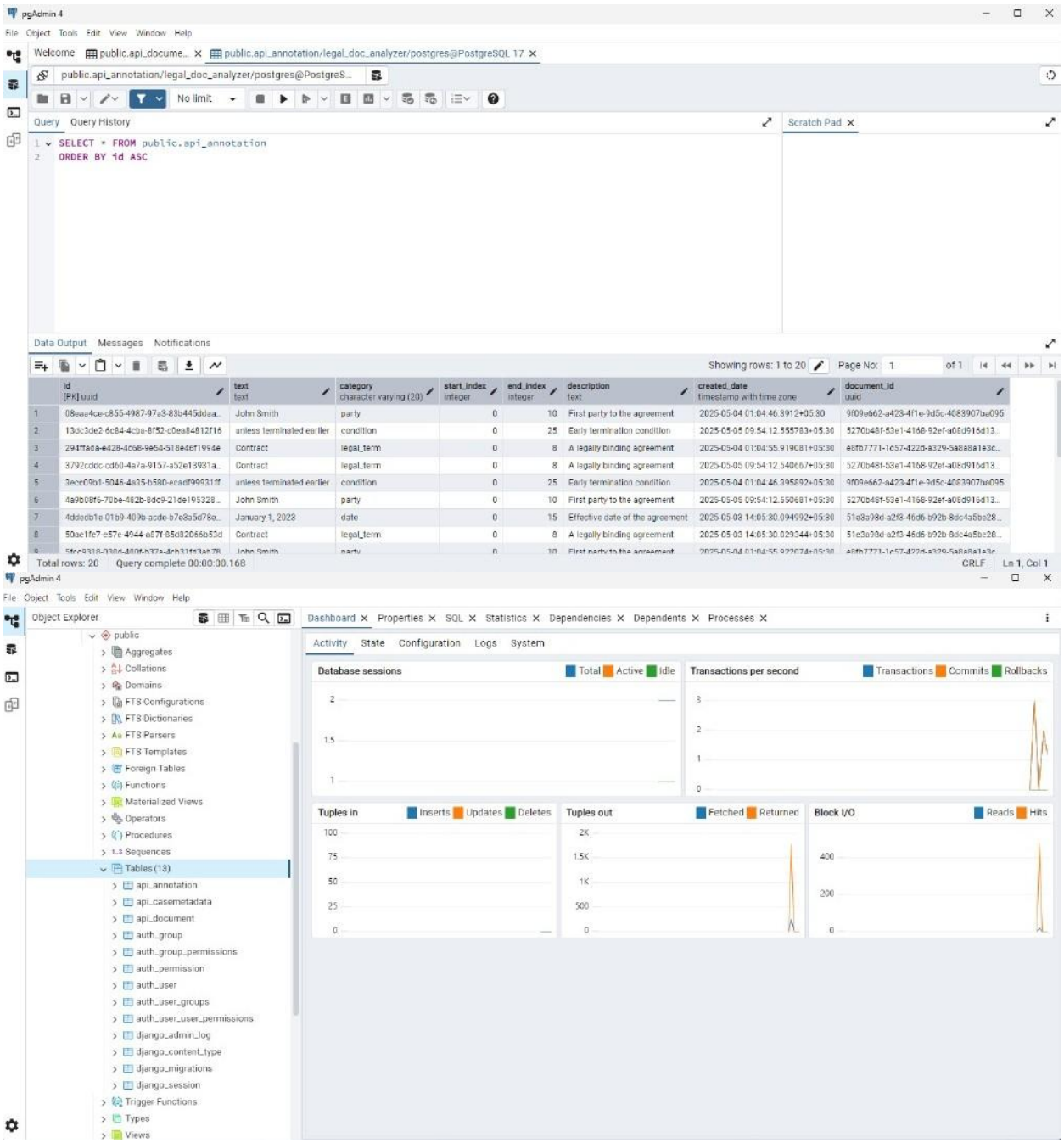
Documentations	Proposal	PID	Interim 01	Interim 02	Research Abstract	Final Submission
Date	13/12/24					
Approved (Yes / No)	Yes	Yes	Yes			
Student Signature						
Supervisor Signature						

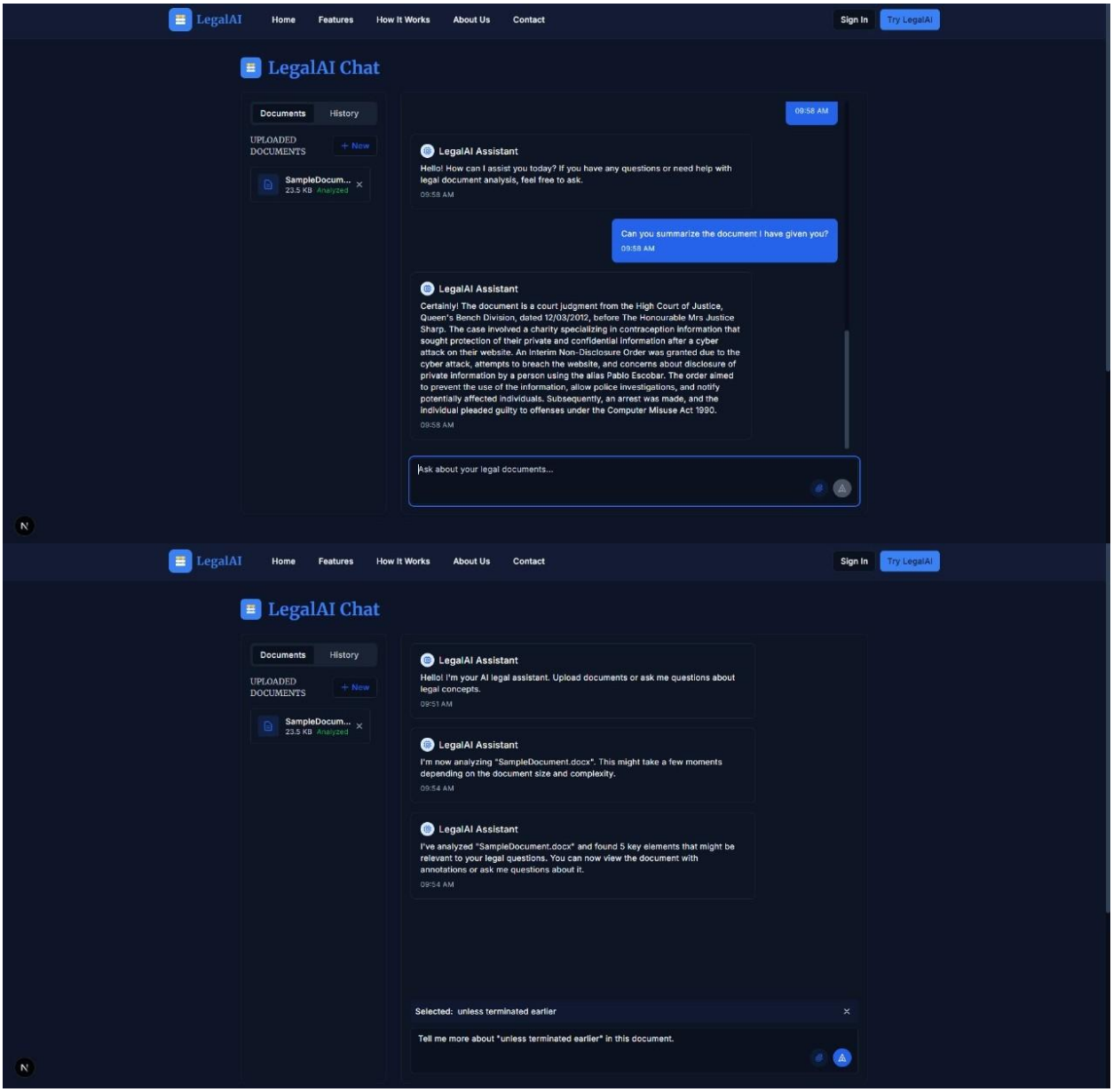
Other Comments (Supervisor Use Only)

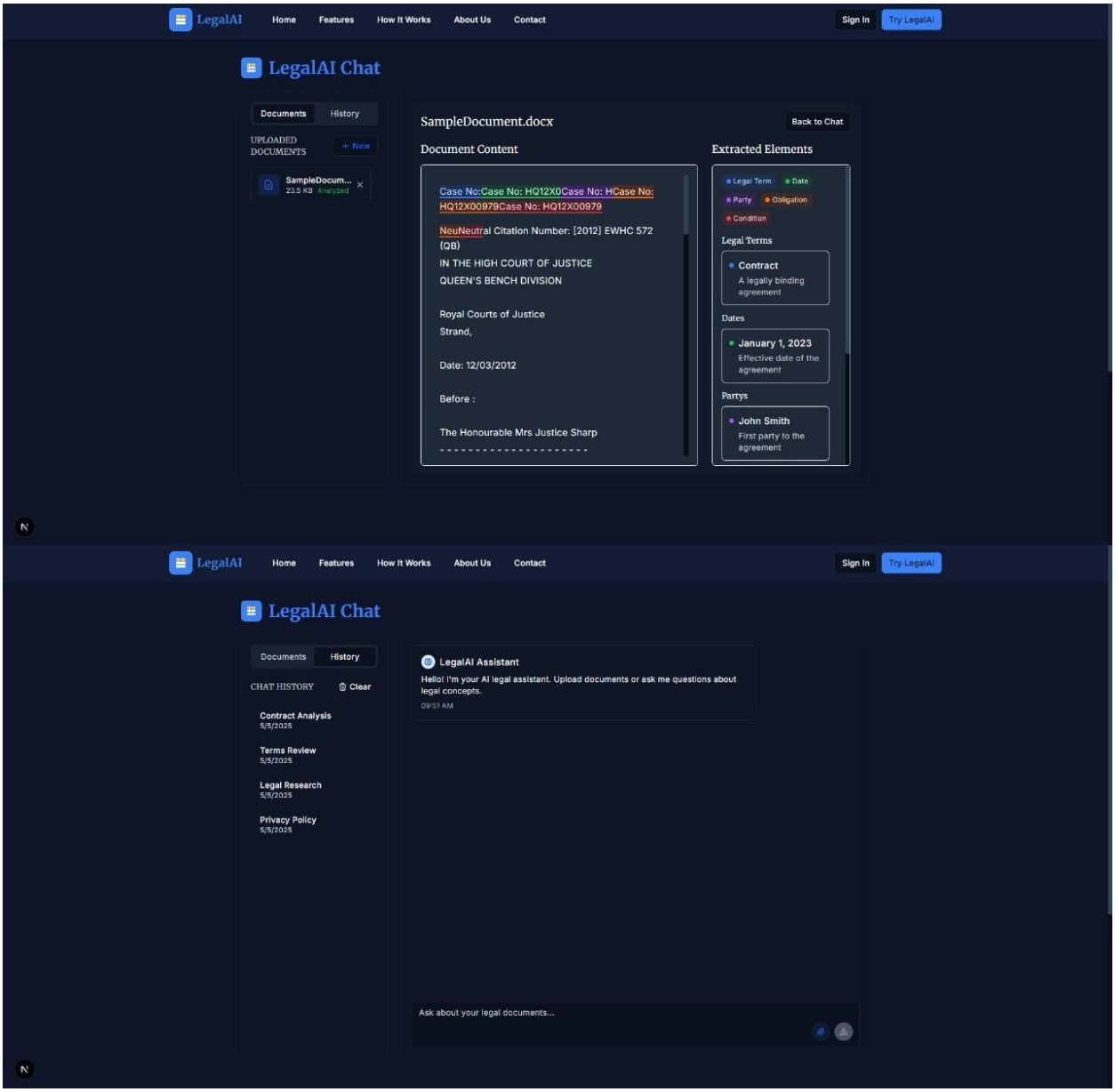
11.5 Other Materials



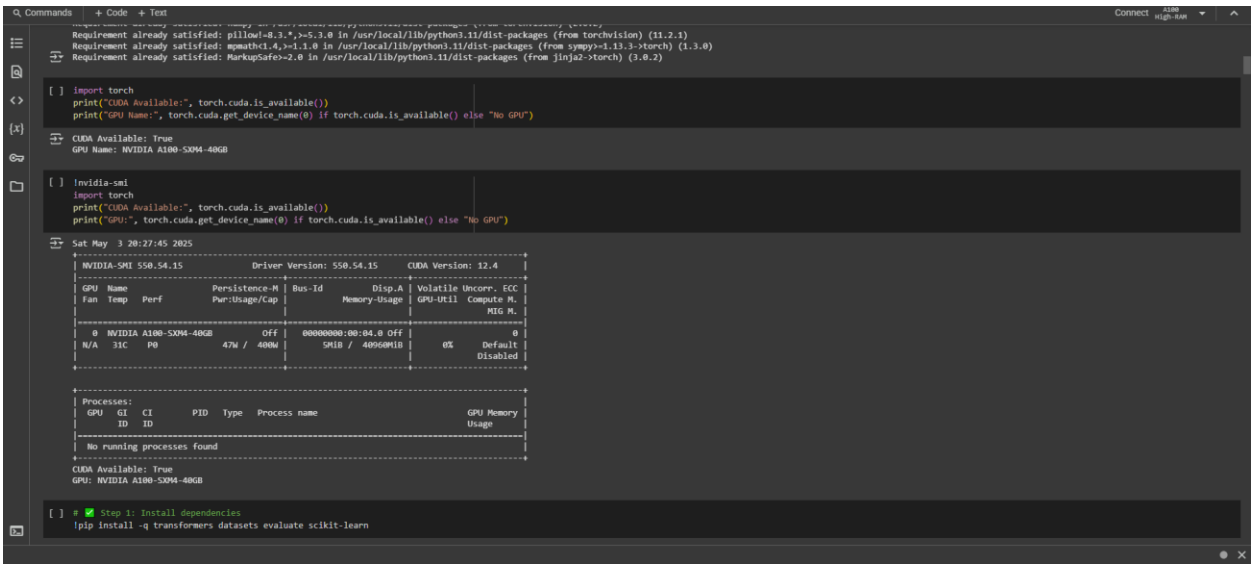


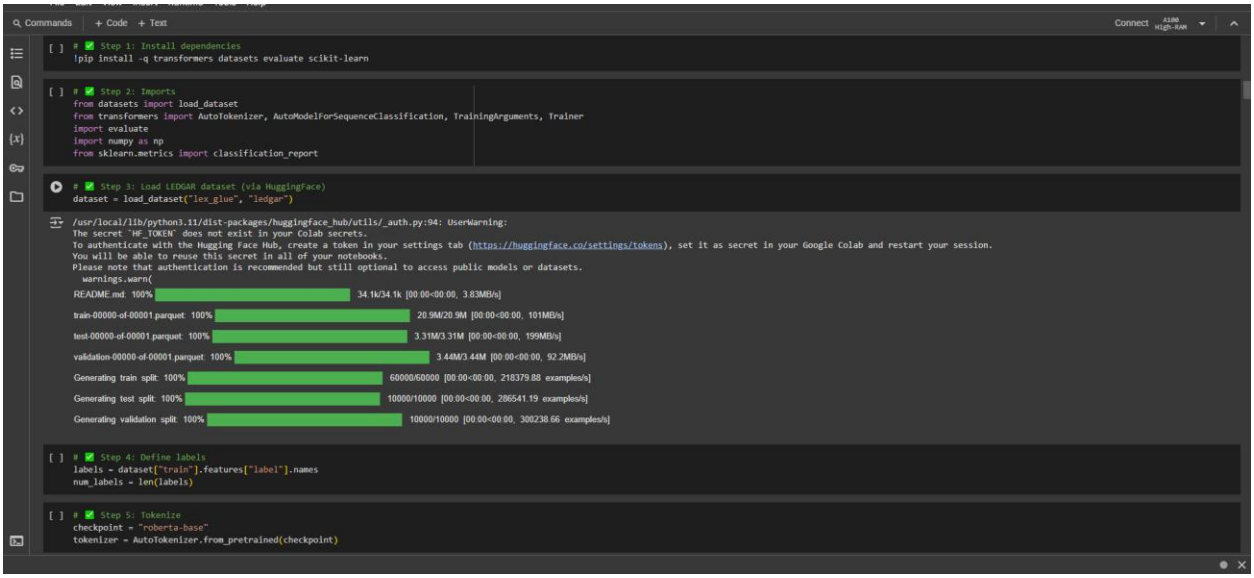

pgAdmin 4
File Object Tools Edit View Window Help
Welcome
public.api_annotation/legal_doc_analyzer/postgres@PostgreSQL 17
No limit
Query Query History
SELECT * FROM public.api_annotation
ORDER BY id ASC
Scratch Pad
Data Output Messages Notifications
Showing rows: 1 to 20
Page No: 1 of 1
id [PK] uuid
text text
category character varying (20)
start_index integer
end_index integer
description text
created_date timestamp with time zone
document_id uuid
Total rows: 20 Query complete 00:00:00.168
CRLF Ln 1, Col 1
Object Explorer
Dashboard Properties SQL Statistics Dependencies Dependents Processes
Activity State Configuration Logs System
Database sessions
Total Active Idle
Transactions per second
Transactions Commits Rollbacks
Tuples in
Inserts Updates Deletes
Tuples out
Fetched Returned
Block I/O
Reads Hits
public
Aggregates
Collations
Domains
FTS Configurations
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views
Operators
Procedures
Sequences
Tables (13)
api_annotation
api_casemetadata
api_document
auth_group
auth_group_permissions
auth_permission
auth_user
auth_user_groups
auth_user_user_permissions
django_admin_log
django_content_type
django_migrations
django_session
Trigger Functions
Types
Views

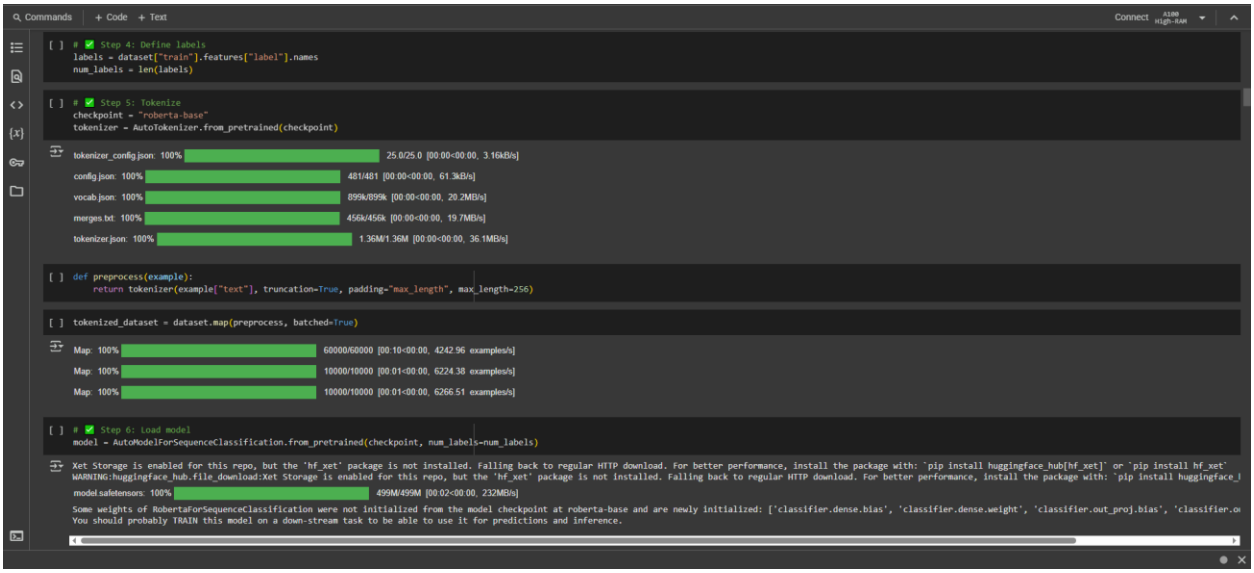

LegalAI
Home
Features
How It Works
About Us
Contact
Sign In
Try LegalAI
LegalAI Chat
Documents
History
UPLOADED DOCUMENTS
+ New
SampleDocum...
23.5 KB Analyzed
09:58 AM
LegalAI Assistant
Hello! How can I assist you today? If you have any questions or need help with legal document analysis, feel free to ask.
09:58 AM
Can you summarize the document I have given you?
09:58 AM
LegalAI Assistant
Certainly! The document is a court judgment from the High Court of Justice, Queen's Bench Division, dated 12/03/2012, before The Honourable Mrs Justice Sharp. The case involved a charity specializing in contraception information that sought protection of their private and confidential information after a cyber attack on their website. An Interim Non-Disclosure Order was granted due to the cyber attack, attempts to breach the website, and concerns about disclosure of private information by a person using the alias Pablo Escobar. The order aimed to prevent the use of the information, allow police investigations, and notify potentially affected individuals. Subsequently, an arrest was made, and the individual pleaded guilty to offenses under the Computer Misuse Act 1990.
09:58 AM
Ask about your legal documents...
LegalAI
Home
Features
How It Works
About Us
Contact
Sign In
Try LegalAI
LegalAI Chat
Documents
History
UPLOADED DOCUMENTS
+ New
SampleDocum...
23.5 KB Analyzed
LegalAI Assistant
Hello! I'm your AI legal assistant. Upload documents or ask me questions about legal concepts.
09:51 AM
LegalAI Assistant
I'm now analyzing "SampleDocument.docx". This might take a few moments depending on the document size and complexity.
09:54 AM
LegalAI Assistant
I've analyzed "SampleDocument.docx" and found 5 key elements that might be relevant to your legal questions. You can now view the document with annotations or ask me questions about it.
09:54 AM
Selected: unless terminated earlier
Tell me more about "unless terminated earlier" in this document.


LegalAI
Home
Features
How It Works
About Us
Contact
Sign In
Try LegalAI
LegalAI Chat
Documents
History
UPLOADED DOCUMENTS
+ New
SampleDocum...
23.5 KB Analyzed
SampleDocument.docx
Back to Chat
Document Content
Case No:Case No: HQ12X0Case No: HCase No: HQ12X00979Case No: HQ12X00979
NeuNeutral Citation Number: [2012] EWHC 572 (QB)
IN THE HIGH COURT OF JUSTICE
QUEEN'S BENCH DIVISION
Royal Courts of Justice
Strand,
Date: 12/03/2012
Before :
The Honourable Mrs Justice Sharp
Extracted Elements
Legal Term
Date
Party
Obligation
Condition
Legal Terms
Contract
A legally binding agreement
Dates
January 1, 2023
Effective date of the agreement
Partys
John Smith
First party to the agreement
LegalAI
Home
Features
How It Works
About Us
Contact
Sign In
Try LegalAI
LegalAI Chat
Documents
History
CHAT HISTORY
Clear
Contract Analysis
5/5/2025
Terms Review
5/5/2025
Legal Research
5/5/2025
Privacy Policy
5/5/2025
LegalAI Assistant
Hello! I'm your AI legal assistant. Upload documents or ask me questions about legal concepts.
09:51 AM
Ask about your legal documents...

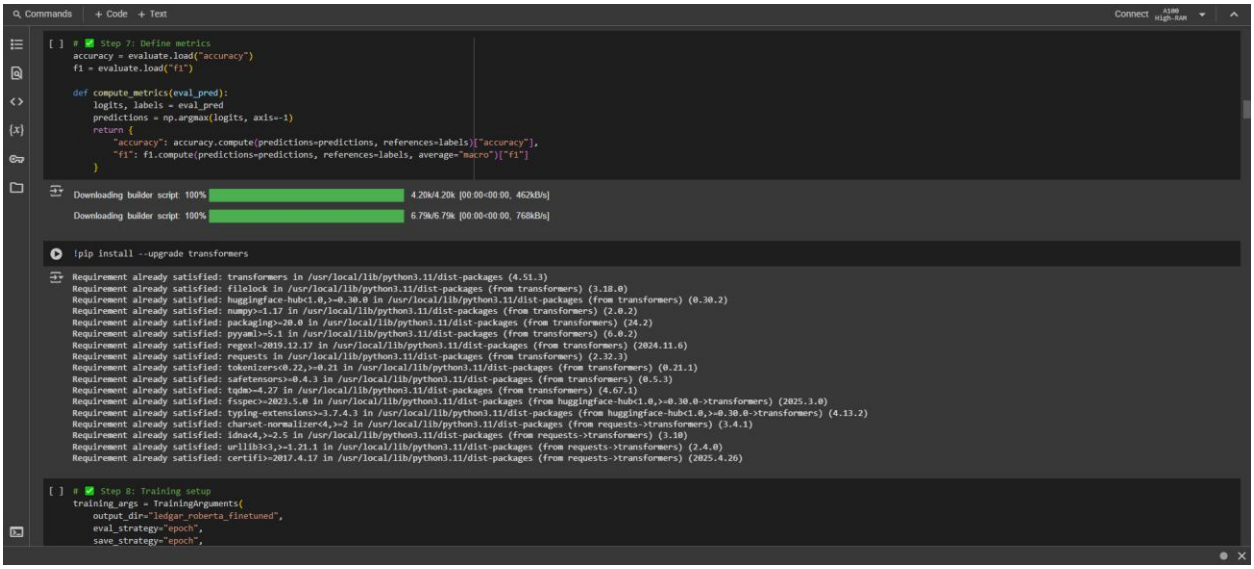
11.6 Test Results



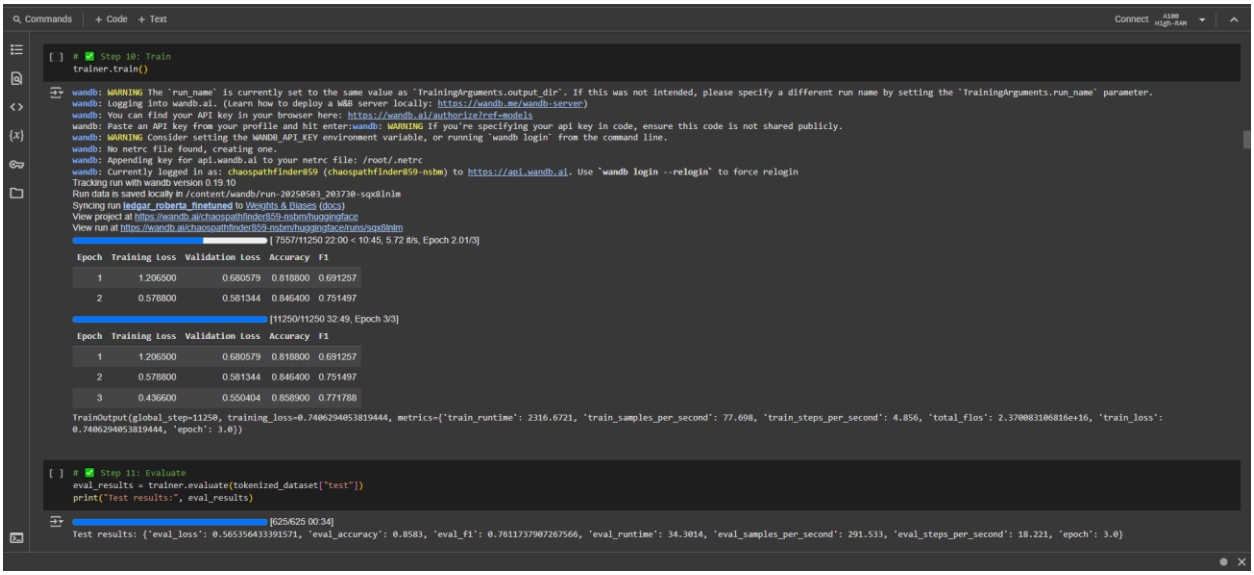




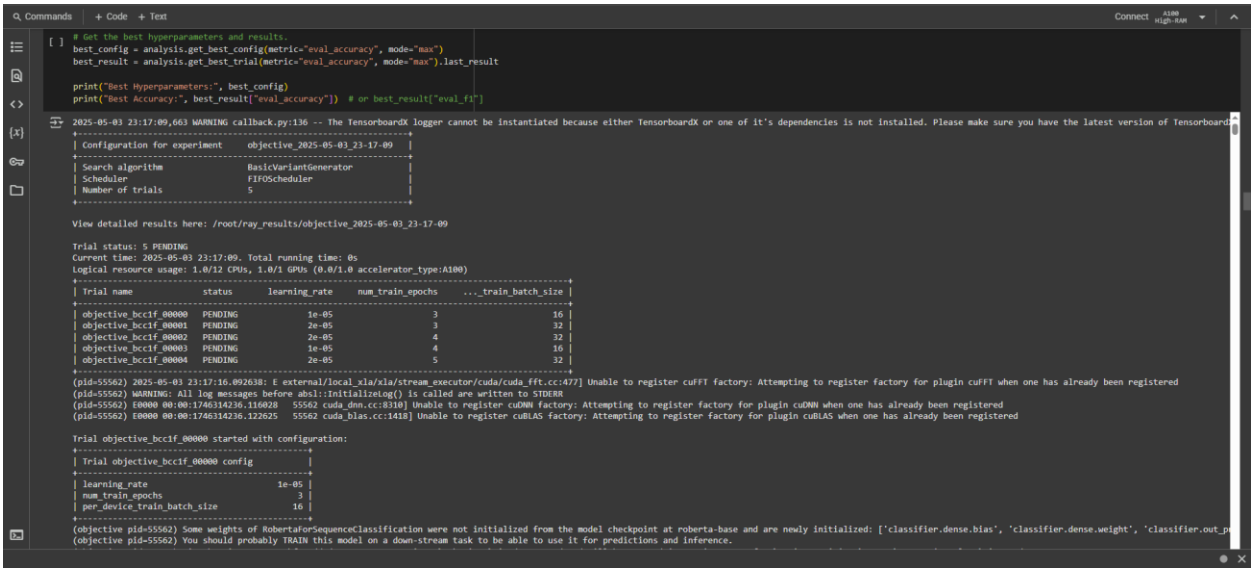
```
Commands + Code + Text                                                    Connect A100 High-RAM

[ ] # ✅ Step 4: Define labels
    labels = dataset["train"].features["label"].names
    num_labels = len(labels)

[ ] # ✅ Step 5: Tokenize
    checkpoint = "roberta-base"
    tokenizer = AutoTokenizer.from_pretrained(checkpoint)

    tokenizer_config.json: 100% 25.0/25.0 [00:00<00:00, 3.16kB/s]
    config.json: 100% 481/481 [00:00<00:00, 61.3kB/s]
    vocab.json: 100% 899k/899k [00:00<00:00, 20.2MB/s]
    merges.txt: 100% 456k/456k [00:00<00:00, 19.7MB/s]
    tokenizer.json: 100% 1.36M/1.36M [00:00<00:00, 36.1MB/s]

[ ] def preprocess(example):
        return tokenizer(example["text"], truncation=True, padding="max_length", max_length=256)

[ ] tokenized_dataset = dataset.map(preprocess, batched=True)

    Map: 100% 60000/60000 [00:10<00:00, 4242.96 examples/s]
    Map: 100% 10000/10000 [00:01<00:00, 6224.38 examples/s]
    Map: 100% 10000/10000 [00:01<00:00, 6266.51 examples/s]

[ ] # ✅ Step 6: Load model
    model = AutoModelForSequenceClassification.from_pretrained(checkpoint, num_labels=num_labels)

    Xet Storage is enabled for this repo, but the 'hf_xet' package is not installed. Falling back to regular HTTP download. For better performance, install the package with: `pip install huggingface_hub[hf_xet]` or `pip install hf_xet`
    WARNING:huggingface_hub.file_download:Xet Storage is enabled for this repo, but the 'hf_xet' package is not installed. Falling back to regular HTTP download. For better performance, install the package with: `pip install huggingface_
    model.safetensors: 100% 499M/499M [00:02<00:00, 232MB/s]
    Some weights of RobertaForSequenceClassification were not initialized from the model checkpoint at roberta-base and are newly initialized: ['classifier.dense.bias', 'classifier.dense.weight', 'classifier.out_proj.bias', 'classifier.o
    You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
```



```
Commands + Code + Text                                                    Connect A100 High-RAM

[ ] # ✅ Step 7: Define metrics
    accuracy = evaluate.load("accuracy")
    f1 = evaluate.load("f1")

    def compute_metrics(eval_pred):
        logits, labels = eval_pred
        predictions = np.argmax(logits, axis=-1)
        return {
            "accuracy": accuracy.compute(predictions=predictions, references=labels)["accuracy"],
            "f1": f1.compute(predictions=predictions, references=labels, average="macro")["f1"]
        }

    Downloading builder script: 100% 4.20k/4.20k [00:00<00:00, 462kB/s]
    Downloading builder script: 100% 6.79k/6.79k [00:00<00:00, 768kB/s]

    !pip install --upgrade transformers

    Requirement already satisfied: transformers in /usr/local/lib/python3.11/dist-packages (4.51.3)
    Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-packages (from transformers) (3.18.0)
    Requirement already satisfied: huggingface-hub<1.0,>=0.30.0 in /usr/local/lib/python3.11/dist-packages (from transformers) (0.30.2)
    Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.11/dist-packages (from transformers) (2.0.2)
    Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-packages (from transformers) (24.2)
    Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.11/dist-packages (from transformers) (6.0.2)
    Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.11/dist-packages (from transformers) (2024.11.6)
    Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (from transformers) (2.32.3)
    Requirement already satisfied: tokenizers<0.22,>=0.21 in /usr/local/lib/python3.11/dist-packages (from transformers) (0.21.1)
    Requirement already satisfied: safetensors>=0.4.3 in /usr/local/lib/python3.11/dist-packages (from transformers) (0.5.3)
    Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.11/dist-packages (from transformers) (4.67.1)
    Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.11/dist-packages (from huggingface-hub<1.0,>=0.30.0->transformers) (2025.3.0)
    Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.11/dist-packages (from huggingface-hub<1.0,>=0.30.0->transformers) (4.13.2)
    Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests->transformers) (3.4.1)
    Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests->transformers) (3.10)
    Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests->transformers) (2.4.0)
    Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests->transformers) (2025.4.26)

[ ] # ✅ Step 8: Training setup
    training_args = TrainingArguments(
        output_dir="ledgar_roberta_finetuned",
        eval_strategy="epoch",
        save_strategy="epoch",
```



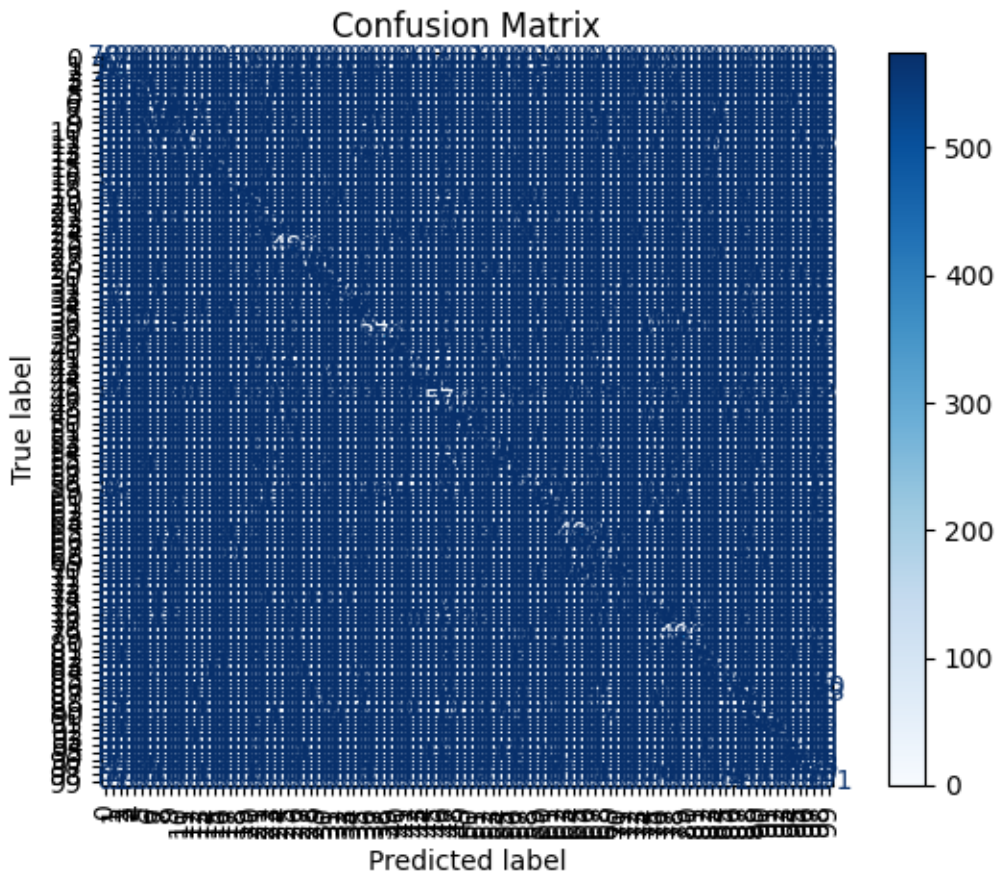
```
Commands + Code + Text                                                                                    Connect A100 High-RAM

[ ] # ✅ Step 10: Train
    trainer.train()

wandb: WARNING The `run_name` is currently set to the same value as `TrainingArguments.output_dir`. If this was not intended, please specify a different run name by setting the `TrainingArguments.run_name` parameter.
wandb: Logging into wandb.ai. (Learn how to deploy a W&B server locally: https://wandb.me/wandb-server)
wandb: You can find your API key in your browser here: https://wandb.ai/authorize?ref=models
wandb: Paste an API key from your profile and hit enter:wandb: WARNING If you're specifying your api key in code, ensure this code is not shared publicly.
wandb: WARNING Consider setting the WANDB_API_KEY environment variable, or running `wandb login` from the command line.
wandb: No netrc file found, creating one.
wandb: Appending key for api.wandb.ai to your netrc file: /root/.netrc
wandb: Currently logged in as: chaospathfinder859 (chaospathfinder859-nsbm) to https://api.wandb.ai. Use `wandb login --relogin` to force relogin
Tracking run with wandb version 0.19.10
Run data is saved locally in /content/wandb/run-20250503_203730-sqx8lnlm
Syncing run ledgar_roberta_finetuned to Weights & Biases (docs)
View project at https://wandb.ai/chaospathfinder859-nsbm/huggingface
View run at https://wandb.ai/chaospathfinder859-nsbm/huggingface/runs/sqx8lnlm
[ 7557/11250 22:00 < 10:45, 5.72 it/s, Epoch 2.01/3]

| Epoch | Training Loss | Validation Loss | Accuracy | F1 |
|---|---|---|---|---|
| 1 | 1.206500 | 0.680579 | 0.818800 | 0.691257 |
| 2 | 0.578800 | 0.581344 | 0.846400 | 0.751497 |

[11250/11250 32:49, Epoch 3/3]

| Epoch | Training Loss | Validation Loss | Accuracy | F1 |
|---|---|---|---|---|
| 1 | 1.206500 | 0.680579 | 0.818800 | 0.691257 |
| 2 | 0.578800 | 0.581344 | 0.846400 | 0.751497 |
| 3 | 0.436600 | 0.550404 | 0.858900 | 0.771788 |

TrainOutput(global_step=11250, training_loss=0.7406294053819444, metrics={'train_runtime': 2316.6721, 'train_samples_per_second': 77.698, 'train_steps_per_second': 4.856, 'total_flos': 2.370083106816e+16, 'train_loss':
0.7406294053819444, 'epoch': 3.0})

[ ] # ✅ Step 11: Evaluate
    eval_results = trainer.evaluate(tokenized_dataset["test"])
    print("Test results:", eval_results)

[625/625 00:34]
Test results: {'eval_loss': 0.565356433391571, 'eval_accuracy': 0.8583, 'eval_f1': 0.7611737907267566, 'eval_runtime': 34.3014, 'eval_samples_per_second': 291.533, 'eval_steps_per_second': 18.221, 'epoch': 3.0}
```

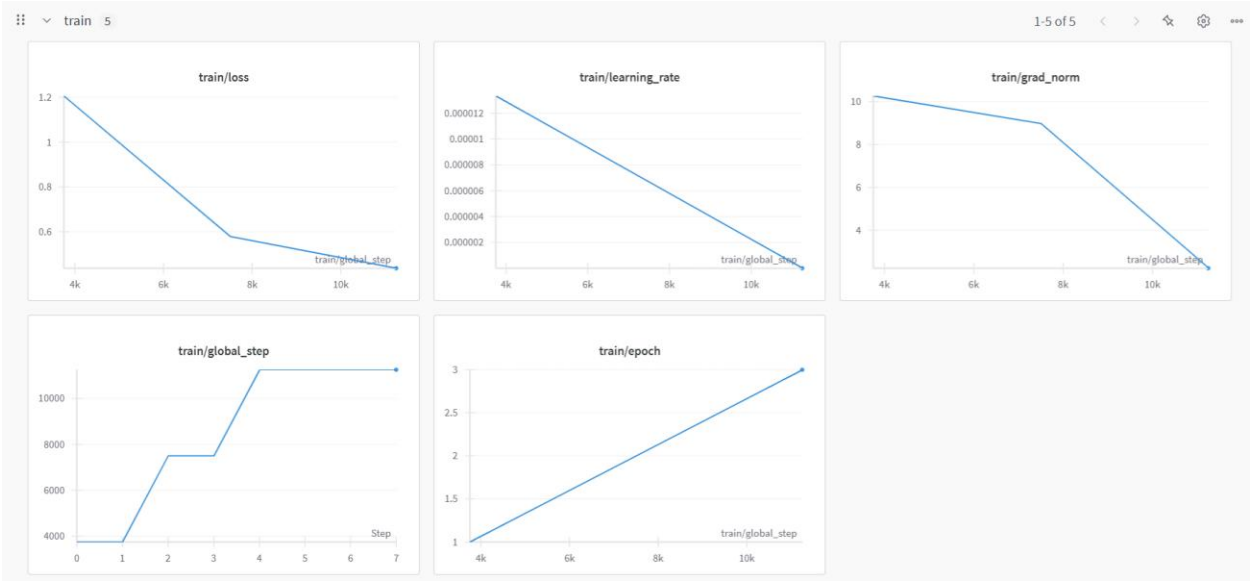


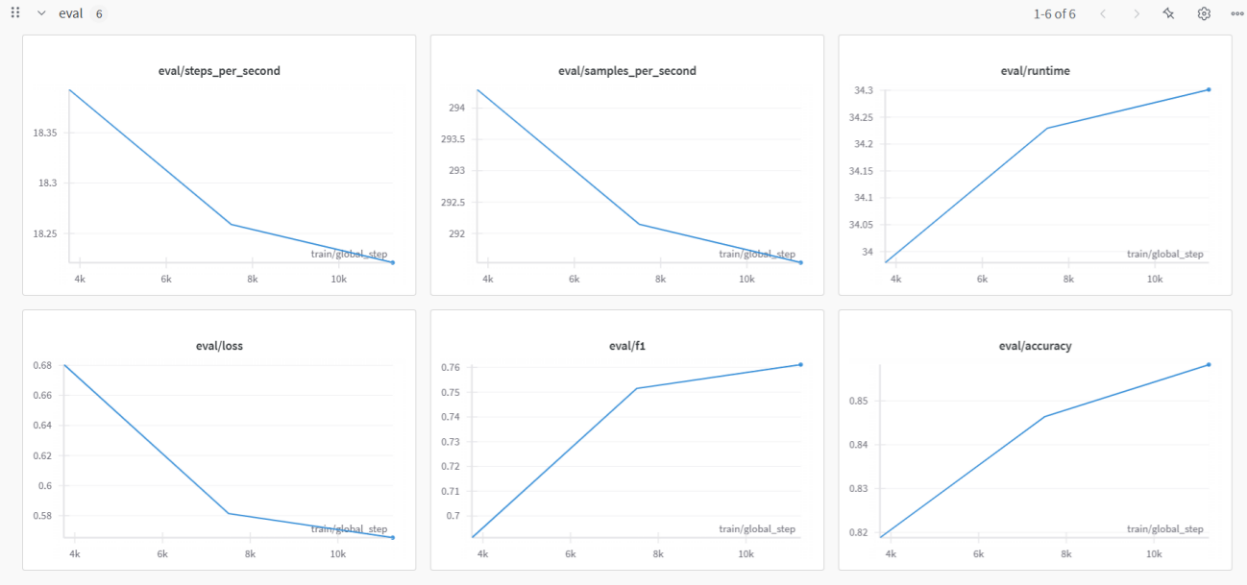
```
Commands + Code + Text                                                                                    Connect A100 High-RAM

[ ] # Get the best hyperparameters and results.
    best_config = analysis.get_best_config(metric="eval_accuracy", mode="max")
    best_result = analysis.get_best_trial(metric="eval_accuracy", mode="max").last_result

    print("Best Hyperparameters:", best_config)
    print("Best Accuracy:", best_result["eval_accuracy"])  # or best_result["eval_f1"]

2025-05-03 23:17:09,663 WARNING callback.py:136 -- The TensorboardX logger cannot be instantiated because either TensorboardX or one of it's dependencies is not installed. Please make sure you have the latest version of Tensorboard
+------------------------------------------------------------+
| Configuration for experiment     objective_2025-05-03_23-17-09   |
+------------------------------------------------------------+
| Search algorithm                 BasicVariantGenerator     |
| Scheduler                        FIFOScheduler             |
| Number of trials                 5                         |
+------------------------------------------------------------+

View detailed results here: /root/ray_results/objective_2025-05-03_23-17-09

Trial status: 5 PENDING
Current time: 2025-05-03 23:17:09. Total running time: 0s
Logical resource usage: 1.0/12 CPUs, 1.0/1 GPUs (0.0/1.0 accelerator_type:A100)
+-----------------------------------------------------------------------------------------------------------+
| Trial name              status       learning_rate     num_train_epochs     ..._train_batch_size |
+-----------------------------------------------------------------------------------------------------------+
| objective_bcc1f_00000   PENDING             1e-05                    3                       16 |
| objective_bcc1f_00001   PENDING             2e-05                    3                       32 |
| objective_bcc1f_00002   PENDING             2e-05                    4                       32 |
| objective_bcc1f_00003   PENDING             1e-05                    4                       16 |
| objective_bcc1f_00004   PENDING             2e-05                    5                       32 |
+-----------------------------------------------------------------------------------------------------------+
(pid=55562) 2025-05-03 23:17:16.092638: E external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:477] Unable to register cuFFT factory: Attempting to register factory for plugin cuFFT when one has already been registered
(pid=55562) WARNING: All log messages before absl::InitializeLog() is called are written to STDERR
(pid=55562) E0000 00:00:1746314236.116028   55562 cuda_dnn.cc:8310] Unable to register cuDNN factory: Attempting to register factory for plugin cuDNN when one has already been registered
(pid=55562) E0000 00:00:1746314236.122625   55562 cuda_blas.cc:1418] Unable to register cuBLAS factory: Attempting to register factory for plugin cuBLAS when one has already been registered

Trial objective_bcc1f_00000 started with configuration:
+-------------------------------------------------+
| Trial objective_bcc1f_00000 config              |
+-------------------------------------------------+
| learning_rate                              1e-05 |
| num_train_epochs                               3 |
| per_device_train_batch_size                   16 |
+-------------------------------------------------+
(objective pid=55562) Some weights of RobertaForSequenceClassification were not initialized from the model checkpoint at roberta-base and are newly initialized: ['classifier.dense.bias', 'classifier.dense.weight', 'classifier.out_p
(objective pid=55562) You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
```

```
Using device: cuda

=== Classification Report ===
              precision    recall  f1-score   support

           0     0.9176    0.8864    0.9017        88
           1     0.5484    0.3542    0.4304        48
           2     0.8120    0.9062    0.8565       224
           3     0.9130    0.9130    0.9130        23
           4     0.0000    0.0000    0.0000        53
           5     0.5455    0.4615    0.5000        26
           6     0.8519    0.9787    0.9109        47
           7     0.8657    0.8923    0.8788       195
           8     0.0000    0.0000    0.0000         4
           9     0.6290    0.6290    0.6290        62
          10     0.7263    0.7667    0.7459        90
          11     0.9821    0.9821    0.9821       112
          12     0.8841    0.7531    0.8133        81
          13     0.6479    0.7302    0.6866       126
          14     0.0000    0.0000    0.0000         2
          15     1.0000    1.0000    1.0000        70
          16     0.9672    0.9365    0.9516        63
          17     0.8646    0.9540    0.9071        87
          18     0.9846    1.0000    0.9922        64
          19     0.9213    0.9567    0.9387       208
          20     0.8950    0.9701    0.9310       167
          21     0.6190    0.7879    0.6933        33
          22     0.7253    0.8148    0.7674        81
          23     0.6087    0.6667    0.6364        63
          24     0.9200    0.8415    0.8790        82
          25     0.4500    0.6000    0.5143        15
          26     0.9819    0.9939    0.9878       490
          27     0.8824    0.9091    0.8955        66
          28     0.6786    0.3393    0.4524        56
          29     0.7192    0.8824    0.7925       119
          30     0.8235    0.8750    0.8485        48
          31     0.9231    0.8660    0.8936        97
          32     0.8068    0.8103    0.8468        58
          33     0.7901    0.8649    0.8258        74
          34     0.5926    0.5161    0.5517        31
          35     0.6792    0.7660    0.7200        47
          36     0.5946    0.4889    0.5366        45
          37     0.6364    0.1667    0.2642        42
          38     0.8674    0.9920    0.9256       376
          39     0.9661    0.9828    0.9744        58
```

```
          67     0.8961    0.9857    0.9388        70
          68     0.7152    0.8571    0.7798       126
          69     0.5526    0.6562    0.6000        32
          70     0.3333    0.4615    0.3871        13
          71     0.9750    0.9286    0.9512        42
          72     0.0000    0.0000    0.0000         5
          73     0.8611    1.0000    0.9254        31
          74     0.8525    0.8667    0.8595        60
          75     0.7626    0.8983    0.8249       118
          76     0.6923    0.7660    0.7273        94
          77     0.8537    0.7778    0.8140        45
          78     0.8125    0.9286    0.8667        14
          79     0.9507    0.9830    0.9666       412
          80     0.9848    0.9848    0.9848        66
          81     0.6939    0.7907    0.7391        43
          82     0.6071    0.5862    0.5965        29
          83     0.9538    0.9841    0.9688        63
          84     0.7949    0.6458    0.7126        96
          85     0.9167    0.9910    0.9524       222
          86     0.5714    0.1600    0.2500        75
          87     0.9048    0.9216    0.9612       185
          88     0.8376    0.8305    0.8340       118
          89     0.8875    0.8068    0.8452       176
          90     0.8125    0.5778    0.6753        45
          91     1.0000    0.9565    0.9778        46
          92     0.9833    0.9833    0.9833       120
          93     0.9808    0.9808    0.9808        52
          94     0.5000    0.2000    0.2857        20
          95     0.8632    0.9425    0.9011        87
          96     0.9732    0.9820    0.9776       111
          97     0.7870    0.8012    0.7940       166
          98     0.8387    0.5361    0.6541        97
          99     0.5401    0.9439    0.6871       107

    accuracy                         0.8589     10000
   macro avg     0.7789    0.7629    0.7622     10000
weighted avg     0.8518    0.8589    0.8496     10000


Accuracy: 0.8589

[MICRO] Precision: 0.8589 | Recall: 0.8589 | F1: 0.8589

[MACRO] Precision: 0.7789 | Recall: 0.7629 | F1: 0.7622

[WEIGHTED] Precision: 0.8518 | Recall: 0.8589 | F1: 0.8496
```


Confusion Matrix
True label
Predicted label
500
400
300
200
100
0


train 5
1-5 of 5
train/loss
train/learning_rate
train/grad_norm
train/global_step
train/epoch
Step


eval 6
1-6 of 6
eval/steps_per_second
18.35
18.3
18.25
train/global_step
4k
6k
8k
10k
eval/samples_per_second
294
293.5
293
292.5
292
train/global_step
eval/runtime
34.3
34.25
34.2
34.15
34.1
34.05
34
train/global_step
eval/loss
0.68
0.66
0.64
0.62
0.6
0.58
train/global_step
eval/f1
0.76
0.75
0.74
0.73
0.72
0.71
0.7
train/global_step
eval/accuracy
0.85
0.84
0.83
0.82
train/global_step

THANK YOU!