# Habib University

CS 435 - Generative AI

Project Idea - 5

**LexiSearch: A Real-Time Document Q&A System for Compliance and Legal Use Cases**

## Team Members


- Hammad Sajid (HU ID: hs07606)
- Iqra Azfar (HU ID: ia07614)
- Ali Muhammad Asad (HU ID: aa07190)


## Project Description

An AWS-powered AI assistant for instant legal document Q&A, reducing compliance review time.

## Abstract


Organizations struggle to extract timely insights from vast legal and compliance document repositories, leading to inefficiencies in audits, customer support, and regulatory checks. LexiSearch addresses this challenge by leveraging AWS services to build a scalable, real-time question-answering system. The solution uses Amazon Textract for text extraction, SageMaker to generate embeddings with domain-specific language models such as LEGAL-BERT and LEGAL-RoBERTa [2], and FAISS for efficient semantic search. A serverless API powered by Lambda and API Gateway enables low-latency responses to user queries, while retrieval-augmented generation (RAG) ensures answers are grounded in the latest legal documents. Additionally, LexiSearch integrates an intent-based summarization feature inspired by Mullick et al.'s framework [1], allowing users to extract legally significant phrases.


## Dataset & Citation

To ensure the system is trained and tested on real-world legal documents, LexiSearch will utilize the Legal Case Reports Dataset from the UCI Machine Learning Repository. This dataset includes 59 annotated legal documents categorized into intents such as Corruption, Murder, Land Dispute, and Robbery. These documents are particularly valuable as they provide a wide variety of legal contexts, making them ideal for training and evaluating the performance of our system in extracting relevant information from legal documents. Additionally, we may supplement this with publicly available terms-of-service documents from platforms like *Terms of Service; Didn't Read*, which provide another layer of compliance-related content.

## Demo Clarity

The LexiSearch demo will demonstrate its functionality through a user-friendly Streamlit interface and API integration via Postman. Users can upload PDF documents to an S3 bucket, input legal questions (e.g., "What are the termination clauses?"), and receive instant, contextually grounded answers. The process involves uploading a document, entering a query, and viewing results with relevant sections highlighted. For Postman-based testing, users send a POST request containing the document and query, and the system returns answers via a RESTful API powered by Lambda and API Gateway. Inspired by Mullick et al.'s "An Evaluation Framework for Legal Document Summarization," [1] the demo includes intent-based summarization, generating summaries focused on intent phrases (e.g., "preparation to kill" for murder cases). The system extracts intent phrases using JointBERT and evaluates summaries using the proposed Intent Metric. This methodology ensures clarity and relevance in handling real-time queries.

## Evaluation Metrics

LexiSearch's evaluation framework will rely on rigorous benchmarking using established metrics such as Mean Reciprocal Rank (MRR) with a target of $\geq 0.85$ and latency benchmarks ensuring responses within $\leq 1.5$ seconds. Accuracy will be validated against ground-truth datasets like CUAD. Additionally, we will incorporate the Intent Metric from Mullick et al.'s paper,[1] which evaluates summary relevance based on intent phrases annotated in legal documents. The metric calculates precision and recall as fractions of sentences forming "closePairs" with intent phrases and derives an F1 Score. This approach ensures contextual relevance and human satisfaction in generated outputs, aligning with our goal of producing accurate and meaningful results.

## LLM Choice

LexiSearch will use lower-parameter models like BART-base and Llama 1B instead of high-parameter models like Jurassic-2 or Falcon-40B, as they are cost-effective and easier to host on SageMaker endpoints while maintaining robust performance when fine-tuned for domain-specific tasks. For instance, BART-base can be fine-tuned on the CUAD dataset, and Llama 1B can generate concise, accurate answers. Inspired by Nasir et al.'s "A Comprehensive Framework for Reliable Legal AI," [2],we will also experiment with domain-specific models like LEGAL-BERT and LEGAL-RoBERTa, pre-trained on large-scale legal corpora, which excel in intent classification and document summarization.

## Rigor and Methodology

To ensure rigor, LexiSearch will undergo thorough testing and validation using multiple datasets, including the Legal Case Reports Dataset, to ensure generalizability. Cross-validation techniques will enhance robustness, and the system will be benchmarked against academic and industry standards, such as Google's Document AI, to demonstrate its competitive edge. Retrieval-augmented generation (RAG) ensures answers are grounded in the latest documents, improving

reliability. Inspired by Nasir et al.'s methodology[2], we will implement a sliding window attention mechanism to handle long legal documents exceeding transformer token limits, enabling processing of thousands of tokens without losing contextual information. This approach provides a strong foundation for handling complex legal texts effectively.

## Proposed Schedule and Milestones

### Week 5-7: Research & Planning

- Set up S3 bucket for document storage.
- Configure IAM roles for Lambda, Textract, and SageMaker.
- Build Terraform scripts for automated AWS resource provisioning.
- **Deliverable**: Functional S3 pipeline with Textract extraction for PDFs.

### Week 8-11: Embeddings & Vector Database

- Process text chunks using LangChain's **RecursiveCharacterTextSplitter**.
- Generate embeddings via SageMaker (e.g., all-MiniLM-L6-v2).
- Store embeddings in Pinecone/FAISS for low-latency retrieval.
- **Deliverable**: Vector database with indexed legal documents.

### Week 12-13: Q&A System Integration

- Deploy a SageMaker endpoint for LLM inference (e.g., Falcon-40B).
- Build a Lambda function with LangChain's **RetrievalQA** chain.
- Connect API Gateway to Lambda for RESTful query handling.
- **Deliverable**: Functional demo accepting user queries via Postman.

### Week 14-15: Optimization & Testing

- Benchmark latency (cold vs. warm Lambda starts).
- Fine-tune chunk size (e.g., 512 tokens) and overlap (20%) for accuracy.
- Validate answers against a ground-truth legal Q&A dataset.
- **Deliverable**: Performance report with corresponding scores.

### Week 16: Final Demo & Deployment

- Try building a simple Streamlit frontend for user interactions.
- **Deliverable**: GitHub repo with code, docs, and demo assets.

## References


[1] Ankan Mullick, Abhilash Nandy, Manav Nitin Kapadnis, Sohan Patnaik, R Raghav, and Roshni Kar. *An Evaluation Framework for Legal Document Summarization.* Indian Institute of Technology Kharagpur, L3S Research Center, Leibniz Universitat Hannover, 2024.

[2] Sidra Nasir, Qamar Abbas, Samita Bai, and Rizwan Ahmed Khan. *A Comprehensive Framework for Reliable Legal AI: Combining Specialized Expert Systems and Adaptive Refinement.* Faculty of Information Technology, Salim Habib University, Karachi, Pakistan, 2024.