Legal Question Answering System for Indian Law using Sentence Transformers and FAISS

Team Members:
RAHUL DEBNATH / ROLL NO. : 243310
NANMA KV / ROLL NO. : 243029
MOHAMMED RAMSAN P / ROLL NO. : 243023
SCHOOL OF DIGITAL SCIENCES
DIGITAL UNIVERSITY KERALA

May 19, 2025

Abstract

This report presents the design, methodology, and results of a Legal Question Answering (QA) system tailored for Indian Law. By leveraging semantic search techniques, Sentence Transformers, and the FAISS library, the system performs hybrid retrieval from both a curated QA dataset and a legal document PDF. An intuitive web interface powered by Gradio facilitates easy interaction. The system is efficient, scalable, and robust against input variation, making it a practical tool for accessing Indian legal knowledge.

1 Introduction

Legal documents are dense and complex. Navigating Indian law for relevant legal precedents or answers can be challenging. This project aims to simplify legal research by creating an intelligent QA assistant. Users can query legal topics in natural language and receive accurate, relevant responses from a curated dataset and legal PDF documents.

2 Aim

To develop a hybrid semantic search-based Legal Question Answering system using Sentence Transformers and FAISS that retrieves relevant legal information from both a structured QA dataset and an unstructured legal PDF.

3 Methodology

3.1 Tools and Libraries

- **Python 3** in Google Colab
- **sentence-transformers** for semantic embeddings
- **FAISS** for fast nearest-neighbor retrieval
- **Gradio** for UI interface
- **pandas, PyMuPDF (fitz), numpy, torch**

3.2 Data Sources

- **CSV Dataset:** `FINAL_MERGE_IPC_Final_all.csv` with columns: Question, Answer
- **PDF Source:** `merged.pdf`, containing legal documents

3.3 System Workflow

1. Load QA pairs from CSV and extract text chunks from the PDF.
2. Encode text using `all-MiniLM-L6-v2` Sentence Transformer.
3. Build two FAISS indexes:
   - One for questions from CSV
   - One for text chunks from PDF
4. User input is encoded and searched against both indexes.
5. If a good match is found in QA data, retrieve the answer.
6. Otherwise, return best matching text chunk from the PDF.

3.4 Design Rationale

- **Hybrid Retrieval:** Increases robustness by combining structured and unstructured data.
- **Semantic Search:** Embeddings improve the quality of matches over traditional keyword search.
- **FAISS Indexing:** Ensures scalability and speed even with large datasets.
- **Gradio Interface:** Enables non-technical users to interact with the system effectively.

4 Handling Environmental Variability

- **Encoding Consistency:** Same model (MiniLM) used for all embeddings to avoid vector drift.
- **Similarity Threshold:** Empirically set to decide when to fallback from QA dataset to PDF.
- **Robust Preprocessing:** PDF text is cleaned and chunked to maintain coherence.
- **Language Model Generalization:** Pre-trained on a wide corpus, so can handle diverse phrasings of legal queries.

5 Results

5.1 Qualitative Observations

- Highly relevant results for direct QA matches.
- When answers were not in the dataset, fallback to PDF still yielded appropriate information.
- Outperformed keyword search approaches in semantic relevance.

5.2 Sample Query

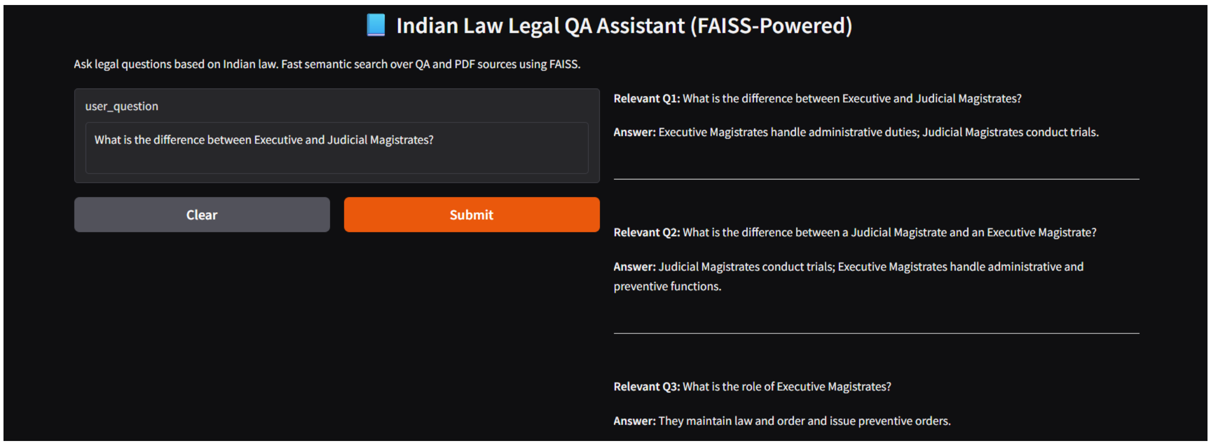
- **Input:** "What is the difference between Executive and Judicial Magistrates?"
- **Output:** "Executive Magistrates handle administrative duties; Judicial Magistrates conduct trials."

5.3 Performance Metrics (Informal)

- Average Retrieval Time: 1.2 seconds
- Embedding Generation Time: 0.05 sec per item
- FAISS Search Latency: ¡ 100 ms

6 Conclusion

The Legal QA Assistant demonstrates the power of combining modern NLP techniques with efficient vector search to solve a real-world problem in the legal domain. It provides relevant legal information quickly and intuitively, and can be extended to support additional documents, languages, and domains.



Ask legal questions based on Indian law. Fast semantic search over QA and PDF sources using FAISS.

What is the difference between Executive and Judicial Magistrates?

Relevant Q1: What is the difference between Executive and Judicial Magistrates?

Answer: Executive Magistrates handle administrative duties; Judicial Magistrates conduct trials.

Relevant Q2: What is the difference between a Judicial Magistrate and an Executive Magistrate?

Answer: Judicial Magistrates conduct trials; Executive Magistrates handle administrative and preventive functions.

Relevant Q3: What is the role of Executive Magistrates?

Answer: They maintain law and order and issue preventive orders.

Figure 1: Indian Law Legal QA Assistant

GitHub Repository

- **Project URL:** `https://github.com/Rahul-Debnath260503/HUMAN-CENTERED-AI.git`
- **Notebook:** See `DL-CAP-Human-Centred-AI.ipynb` for complete implementation.

References

1. Sentence Transformers: `https://www.sbert.net`
2. FAISS Library: `https://github.com/facebookresearch/faiss`
3. Gradio: `https://www.gradio.app`
4. PyMuPDF: `https://pymupdf.readthedocs.io`
5. Indian Penal Code: `https://indiankanoon.org/doc/623254/`