LAWLITE – INDIAN LEGAL SECTION SUMMARIES

Project number: 18

Group Members :
Sri Sampath-se23umcs055
Satyavardhan-se23umcs022
Sathvik reddy-se23umcs001
Neal teja-se23umcs004

Course / Instructor : Nidhi Goyal

Date : 27/11/25

MOTIVATION

- **Why is this project important?**
Legal language in India is often lengthy and difficult for the general public to understand. Most people struggle to interpret sections without legal expertise.

- **Real-world problem it addresses:**
Citizens, students, and professionals waste time and face confusion when trying to understand bare acts. There is a need for quick, reliable, plain-language summaries.

- **Gaps in current solutions:**
Existing online legal portals (e.g., Indian Kanoon, Bare Acts sites) provide raw text but lack simplified explanations or structured summaries.

PROBLEM STATEMENT

- **Problem Definition:**
Users find it difficult to comprehend complex legal jargon in Indian law sections.

- **Inputs & Expected Outputs:**

Input:
Section number or keyword (e.g., "Section 420 IPC")

Output:
- Official legal text with citations
- Concise, plain-language summary for easy understanding

- **Scope:**
- **Included:** Retrieval of official text, NLP-based summarization, disclaimer for legal accuracy
- **Excluded:** Providing professional legal advice or interpretations beyond summaries

DATASETS

We took datasets from kaggle :

- Constitution of india.CSV

- Index.CSV

- docum.pdf

- legaldoc.pdf

EXPECTED OUTCOMES

Results / Metrics:
- Accurate retrieval of relevant sections (>95% retrieval precision).
- Summaries evaluated with ROUGE/BLEU scores for quality.
- Readability improvements measured with Flesch-Kincaid index.

Applications / Use-Cases:
- Quick reference for law students, journalists, and citizens.
- Support for civic awareness programs and legal literacy.

Benefits:
- Saves time and reduces confusion when understanding Indian law.
- Provides a single platform for both official text and simplified summaries.

NLP CONCEPTS USED IN OUR LAW SUMMARISER PROJECT

We used 5 major NLP concepts:

1. Text Preprocessing

2. Tokenization

3. Stopword Removal

4. TF-IDF Vectorization

5. Cosine Similarity

NLP CONCEPTS USED IN OUR LAW SUMMARISER PROJECT

We used 5 major NLP concepts:

1. **Text Preprocessing** – cleaned dataset

2. **Tokenization** – splitting text into words

3. **Stopword Removal**– removing useless words

4. **TF-IDF Vectorization** – converting text into weighted numeric vectors

5. **Cosine Similarity** – matching user query with correct Article

6. **Rule-Based NLP** – exact Article match, number detection, category detection

METHODOLOGY & MODEL

- **First, we created a proper workflow called an NLP pipeline, which includes:**

Dataset Cleaning & Merging:

- We collected data from different sources and then:

- Removed duplicate Articles and definitions

- Fixed formatting problems (extra spaces, line breaks)

- Removed unnecessary headings

- Combined everything into one clean dataset

ANALYSIS & DISCUSSION

Strengths:

- Fast and accurate Article search

- Simple, easy-to-read summaries

- Works for both Constitution Articles & Legal Documents

Limitations:

- Summaries are basic

- Depends fully on dataset quality

- TF-IDF cannot understand deep legal meaning

Future Improvements:

- Use BERT for smarter summaries

- Add multilingual support

- Add spell correction & suggestions

OUR CONTRIBUTION

Member 1 – Dataset & Data Cleaning Lead

Member 2 – NLP & Model Development Lead

Member 3 – UI/Frontend & Application Developer

Member 4 – Documentation, Testing & Presentation Lead

PROPOSED PIPELINE (DETAILS)

Step 1: Data Collection & Preparation
- Gather open-source Indian legal texts (e.g., IPC, CrPC, Bare Acts).
- Clean and standardize formatting, remove duplicate sections.

Step 2: Information Retrieval
- Implement keyword/section search using Elasticsearch or Whoosh to fetch the relevant section text quickly.

Step 3: Preprocessing
- Tokenization, sentence segmentation, and stop-word removal tailored for legal language.

Step 4: Summarization Model
- Extractive: TextRank or BERT-based extractive summarizer.
- Abstractive (optional): Transformer model (e.g., fine-tuned BART/T5).

Step 5: Web Application
- Backend: Python Flask/Django.
- Frontend: React/HTML/CSS for user-friendly display of full text + summary + disclaimer.

Tools & Frameworks: Python, NLTK/Spacy, HuggingFace Transformers, Elasticsearch, Flask/Django, React.

Datasets: Publicly available Bare Acts/IPC datasets, Indian Kanoon or government legal text repositories (open data).

TIMELINE

Week	Task	
1–2	Literature review, requirements analysis	
3–4	Data collection & cleaning	
5–6	Build and test information-retrieval system	
7–8	Develop and fine-tune summarization model	
9–10	Integrate backend + frontend web app	
11	User testing & evaluation	
12	Documentation & final presentation	

DEMO

CONCLUSION

Our Law Summariser uses key NLP techniques like preprocessing, tokenization, stopword removal, TF-IDF, and cosine similarity to understand user queries and find the most relevant legal Article. By converting text into numerical vectors and matching similarity, the system delivers accurate results even without exact input. This makes legal information easier to search, faster to access, and more user-friendly.