Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Legal Statute Recommendation for Indian Judgments

BCSE409L - Natural Language Processing

Faculty: Rajeshkannan R

Case Study - III

22BCE3347 – Armaan Indani

22BCE3629 – Nipun Misra

22BCE3469 – Aditya Tripathi

Github Link:
https://github.com/Armaan-Indani/NLP-Legal-Statute-Identification-ILB.git

1. ABSTRACT

India’s legal system is vast and complex, with millions of cases processed yearly across diverse jurisdictions. A key challenge is the accurate and timely identification of relevant legal statutes (e.g., IPC, CrPC, special Acts) based on case narratives. This task is manual, time-consuming, and even more difficult in regional courts where judgments may be in local languages.

To address this, we propose a Multilingual Legal Statute Recommendation System for Indian Judgments that understands text and audio inputs in English, Hindi, Marathi, Tamil, and Kannada. The system’s goal is to automatically recommend the most relevant statutory provisions for a given case description.

It will use state-of-the-art NLP techniques, including transformer-based legal models and speech recognition. At its core is InLegalBERT, a BERT variant fine-tuned on Indian legal texts, combined with advanced statute identification modules trained on datasets like LeSICiN, which link judgments to statutes.

To support multilingual input, models like IndicBERT, MuRIL, and mBERT will be used. A language detection and translation pipeline will ensure consistent semantic interpretation. For audio inputs, ASR models such as Whisper or IndicTTS (trained on Indian legal speech) will enable real-time, spoken-case understanding.

The system will output a ranked list of statutes with confidence scores and explanations, showing the relevance of each suggestion. Additionally, graph-based models may be incorporated to exploit citation networks, improving contextual accuracy based on prior case references.

This tool aims to benefit lower courts and rural legal settings where legal database access and English proficiency are limited. Supporting regional languages and speech input, it promotes accessible legal intelligence, enhancing research efficiency and decision-making.

In summary, this project unites NLP, multilingual AI, legal domain expertise, and speech processing to build an inclusive, intelligent, and scalable Legal Statute Recommendation System for the Indian judiciary, promoting faster legal aid, linguistic inclusivity, and statutory transparency.

Keywords:

Legal Statute Recommendation, Indian Judiciary, Case Law Analysis, Statutory Interpretation, Legal Research Automation, Multilingual NLP

2. INTRODUCTION

India's legal system is a vast and complex network, rooted in centuries of jurisprudence and evolving through layers of statutes, case law, and constitutional interpretation. It ranks among the largest legal systems globally, with over 25 million cases pending across various levels of the judiciary, including the Supreme Court, High Courts, and a wide network of subordinate courts. Each case is governed by numerous statutes, such as the Indian Penal Code (IPC), Code of Criminal Procedure (CrPC), and Code of Civil Procedure (CPC), along with various special and regional laws. Navigating this system requires not only legal expertise but also the ability to associate specific statutory provisions with the facts, arguments, and context of individual cases.

A core challenge for legal practitioners, judges, researchers, and law students is identifying relevant statutes from case descriptions. While seemingly straightforward, the task is complicated by legal documents written in dense, technical language and by case facts that are often unstructured and vary significantly in presentation. In India, these challenges are amplified by linguistic diversity, uneven access to digital legal databases, and the sheer volume of documentation processed daily [6][7][11]. Manual analysis of lengthy narratives to map them to relevant statutory sections is labor-intensive and prone to omissions or errors, particularly in lower courts or rural settings where digital tools and legal expertise are limited [11][12]. Moreover, many legal proceedings in district and taluka courts are conducted in regional languages, and judgments are often dictated orally or recorded with localized terminology, making reliance on English-only tools inadequate [6][13][15].

To address these gaps, this project proposes a *Multilingual Legal Statute Recommendation System for Indian Judgments*. The system will accept legal case descriptions in both text and speech form, in English and four regional Indian languages, and recommend a ranked list of relevant statutes. Designed as an assistive legal research tool, it aims to help practitioners, students, and citizens quickly identify statutory provisions most relevant to a given case.

At its core, the system leverages advanced Natural Language Processing (NLP) models, specifically transformer-based architectures capable of capturing semantic, syntactic, and contextual nuances [3][4][5]. We employ *InLegalBERT*, a BERT-based model pre-trained and fine-tuned on Indian legal texts such as judgments, statutes, and commentaries [4]. Unlike general-purpose BERT models trained on generic web data, InLegalBERT is tailored to the intricate phrasing and terminology of the Indian legal domain [4][3].

The system also utilizes labeled datasets such as *LeSICiN* (Legal Statute Identification using Citation Network), which maps Indian court judgments to statutes they cite or are influenced by [1]. This enables models that recommend statutes not only through textual similarity but also via legal reasoning and citation relationships are crucial when cases with different surface texts are governed by similar statutes based on precedent or judicial interpretation [1][2][10].

To support multilingual capability, models like *IndicBERT*, *MuRIL*, and *mBERT* are integrated, enabling the system to process inputs in languages such as Hindi, Tamil, Bengali, Marathi, or Kannada [6][15]. A language detection and translation pipeline ensures all inputs are normalized before processing, maintaining semantic consistency regardless of the source language [6][13][15].

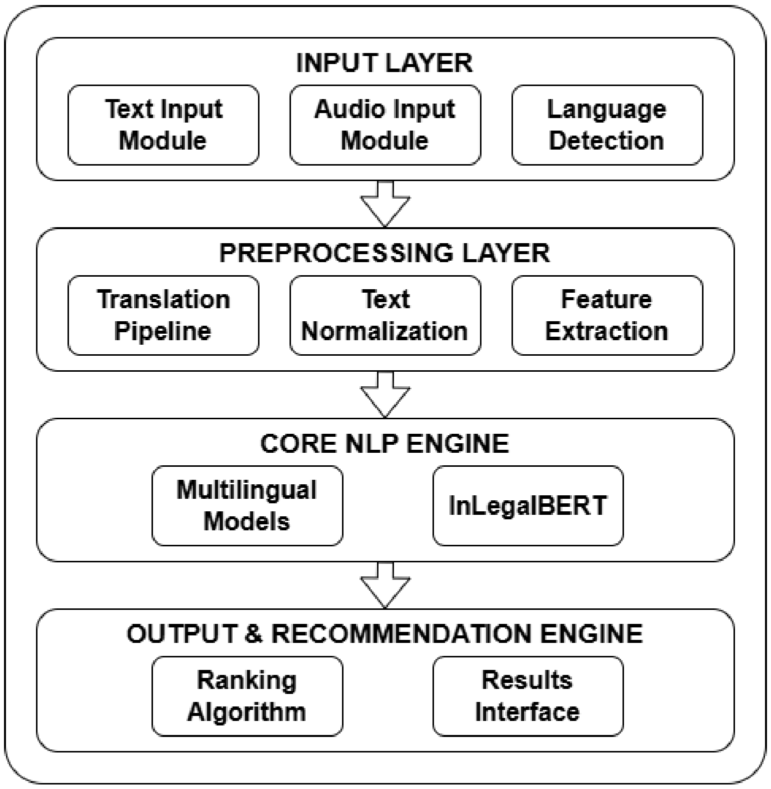
For speech-based inputs, the system incorporates *Automatic Speech Recognition (ASR)* using models like Whisper (by OpenAI) and IndicTTS, trained for Indian accents, code-switching, and legal domain terminology [15]. This is particularly beneficial for users with limited typing skills, disabilities, or in low-literacy regions [6][11].

The output includes a ranked statute list with confidence scores and natural language explanations linking the input to each recommendation [10][15].

The proposed system has diverse applications as a decision-support tool for judges, a research assistant for lawyers and students, a public-facing legal aid service, and an educational resource. Its multilingual and speech-enabled design makes it especially valuable in lower courts, rural jurisdictions, and semi-urban areas where access to legal resources is limited [6][11][13]. By reducing dependency on English-centric systems, it promotes inclusive access to justice [6][7][15].

In conclusion, this *Multilingual Legal Statute Recommendation System* combines advanced NLP, domain-specific modeling, multilingual understanding, and speech recognition to transform statute identification in India. It addresses the technical challenge of automating recommendations while supporting broader goals of legal empowerment, equitable access to justice, and digital transformation of the judicial ecosystem [4][6][15]. This represents a significant step toward intelligent, inclusive, and context-aware legal tools for a linguistically diverse and legally complex nation.

2.1 Proposed Architecture Diagram



3. LITERATURE REVIEW

With the increasing digitalization of court records and the availability of structured legal texts, the intersection of Natural Language Processing (NLP) and legal research has emerged as a powerful domain of applied artificial intelligence. One core task that has received attention in recent years is Legal Statute Identification (LSI), automatically determining the most relevant statutory provisions (e.g., sections of the Indian Penal Code) for a given legal case description. Traditional LSI systems have relied heavily on text classification, but recent advances in deep learning, graph-based networks, and domain-adapted language models have dramatically improved accuracy and scalability. This survey explores five prominent lines of research relevant to building a Multilingual Legal Statute Recommendation System for Indian Judgments, focusing on citation-aware models, graph-augmented methods, and domain-specific language model pretraining.

In their work on Legal Statute Identification (LSI) in Indian law, Paul et al. propose a novel, graph-based model named LeSICIN [1]. They argue that existing LSI methods rely solely on the textual content of legal documents and statutes, ignoring the valuable information available in the legal citation network. To address this, LeSICIN models the documents and statutes as a heterogeneous graph, learning both textual and structural features to predict links between new case documents and relevant statutes. To facilitate this research, the authors curated a large-scale dataset, the Indian Legal Statute Identification (ILSI) dataset, which consists of facts from Indian court cases and a set of 100 statutes from the Indian Penal Code. Experiments showed that LeSICIN significantly outperforms several state-of-the-art baselines, and the ILSI dataset more accurately reflects the multi-label nature of the LSI problem compared to other existing datasets.

A study by Bhattacharya and others proposed a new approach for calculating the similarity between legal case documents by developing a heterogeneous network called Hier-SPCNet [2]. This network augments the traditional Precedent Citation Network (PCNet) by incorporating the hierarchical structure of legal statutes and their citation links with case documents and other statutes. The authors applied the graph embedding algorithm, Metapath2vec, to this network to estimate document similarity. They validated their method on a dataset of 100 pairs of Indian

Supreme Court case documents, whose similarities were annotated by legal experts. The results showed a significant improvement in document similarity estimation compared to methods that only use the PCNet. The study also demonstrated that their network-based method provides complementary insights to a text-based method (Doc2Vec), and that combining the two approaches can be beneficial for estimating legal document similarity.

Chalkidis and colleagues [3] explore the adaptation of BERT, a pre-trained language model, to the legal domain. They investigate three main strategies: using the original BERT model as is, further pre-training it on domain-specific corpora (Legal-BERT-FP), and training a new BERT model from scratch on legal texts (Legal-BERT-SC). The study found that both further pre-training and training from scratch on legal corpora led to better performance on domain-specific tasks compared to using the original BERT model. They also discovered that a wider range of hyperparameter tuning during the fine-tuning phase can lead to substantial performance improvements. The authors introduced and released a family of legal BERT models, including a smaller, more efficient version called Legal-BERT-SMALL, which was shown to be competitive with larger models despite its smaller size.

Building on this, InLegalBERT [4] was trained on 5.4 million Indian legal documents and demonstrated superior performance across tasks like statute identification, judgment segmentation, and appeal outcome prediction. The model outperformed LegalBERT and CaseLawBERT on both Indian and EU/UK datasets, affirming the benefit of country-specific pretraining and custom vocabularies for legal tasks.

A paper by Dhanani et al. addresses the need for an effective and scalable legal judgment recommendation system (LDRS) for legal professionals [5]. The authors propose a pre-learned word embedding-based LDRS (P-LDRS) that uses domain-specific pre-learned word embeddings to initialize the Doc2Vec model, thereby enhancing the semantic representation of legal judgments. To tackle the scalability issues associated with large and growing legal document corpora, they also introduced a distributed version of their P-LDRS that leverages frameworks like MapReduce and Spark. Their empirical analysis on a dataset of over 48,000 Indian Supreme Court judgments demonstrates that the non-distributed P-LDRS performs significantly better than a traditional Doc2Vec-based LDRS, achieving an Accuracy of 0.88, an

F1-score of 0.82, and an MCC Score of 0.73. Furthermore, the distributed P-LDRS shows improved time efficiency with a stable performance, confirming its effectiveness and scalability for handling large volumes of legal text.

The application of Natural Language Processing (NLP) within the legal sector has experienced substantial growth, motivated by the necessity to manage intricate legal documents and optimize judicial workflows. Ariai et al. [6] provide an extensive survey detailing the progression of legal NLP, systematically categorizing key tasks such as legal document summarization, named entity recognition, legal question answering, judgment prediction, and argument mining. Their analysis underscores challenges unique to legal language, including the considerable length of documents, scarcity of structured datasets, and concerns related to fairness and explainability in legal contexts.

Addressing these challenges in the Indian legal environment, Joshi et al. [7] introduced IL-TUR, a multilingual benchmark designed specifically for Indian legal text comprehension and reasoning. IL-TUR encompasses eight distinct tasks ranging from rhetorical role labeling to legal summarization across English and nine Indian languages. This benchmark not only provides baseline performance metrics but also highlights the limitations of existing large language models (LLMs) when applied to authentic Indian legal texts, which are frequently lengthy, unstructured, and domain-specific.

In an endeavor to improve legal information retrieval within the Indian legal framework, Pandian and Joshi [8] introduced a context-aware recommendation system for statutes and case law, leveraging transformer-based masked language models. Their research underscores the superiority of semantic representations over traditional lexical matching approaches, particularly in the retrieval of pertinent statutes and judicial precedents. By fine-tuning language models originally developed for general domains on Indian legal corpora, their methodology demonstrates a marked enhancement in the accuracy of statute recommendations relative to conventional information retrieval techniques. Importantly, the study identifies a significant challenge: while statutes can be retrieved with greater consistency through semantic similarity measures, recommending prior case law proves more complex due to its inherent intricacy and heterogeneity. This finding highlights the necessity for further domain-specific adaptation and the development of specialized modeling strategies tailored to the Indian legal natural language processing context.

Concurrently, statute recommendation systems have emerged as valuable tools for legal practitioners. Feng et al. [9] proposed a portable, two-stage neural network-based statute recommender incorporating attention mechanisms and a ranking module. This framework advances multi-label statute prediction by leveraging deep learning techniques to capture contextual and semantic relationships among legal provisions, thereby improving prediction accuracy.

Complementing this approach, Li et al. [10] developed a more interpretable methodology employing hand-crafted relational features to model the relationship between cases and statutes. Their hybrid system integrates collaborative filtering for initial candidate retrieval with feature-based learning-to-rank models for final statute selection. The incorporation of explainable features enhances interpretability and practical applicability, particularly in civil law domains such as divorce cases in China.

Deshmukh and Kamble [11] developed IndianBailJudgments-1200, a structured dataset consisting of 1,200 Indian bail orders annotated via prompt-engineered GPT-4o. It provides fields for fairness analysis, judgment prediction, and social research. While the dataset is pioneering, its focus on High Court judgments and English-only documents limits its generalization.

Kalamkar et al. [12] focused on Named Entity Recognition (NER) within legal texts, presenting a dataset with 46,545 annotated entities from Indian court judgments. Baseline models like RoBERTa, supported by post-processing for coreference and long-span handling, achieved high accuracy. However, sentence-level annotations and English-only judgments pose constraints.

Joshi et al. [13] introduced INDICourtSumm, a large-scale summarization dataset with over 31,000 Indian legal cases and summaries derived from IndianKanoon. Fine-tuned PEGASUS and BART models showed strong summarization performance (ROUGE-1: ~57.4), but the lack of gold-standard manual summaries introduces evaluation noise.

Malik et al. [14] contributed the ILDC for the CJPE dataset, designed for judgment prediction and explanation tasks. Using models like BiLSTM and Legal-BERT, the study achieved over 81% F1-score in binary outcome classification and provided rationale-based interpretability. Despite its innovation, the explanation depth and case complexity remain limited.

Bhattacharya and Ghosh [15] compared extractive and abstractive summarization approaches on Indian court case documents. While extractive methods like TextRank and LexRank performed reasonably well, abstractive models such as T5 and BART generated more concise and coherent summaries. The work highlighted challenges like domain-specific terminology handling and document length limitations that hinder model accuracy.

Collectively, these studies contribute crucial resources and insights into the growing field of Legal NLP in India. The ongoing challenges emphasize the need for multilingual, multi-tier court data, advanced reasoning models, and better factual consistency in generated outputs.

3.1 Summary Table

S. No	**Title and Authors (Year)**	**Methodology / Approach & Dataset**	**Key Findings / Contributions**	**Limitations / Gaps**
1	**LeSICiN: A Heterogeneous Graph-based Approach for Automatic Legal Statute Identification from Indian Legal Documents - Paul et al. (2022)**	Uses a heterogeneous graph to model relationships between case facts and statutes. Treats Legal Statute Identification (LSI) as an inductive link prediction problem. Trained on a new Indian dataset mapping court judgments to IPC sections.	Achieved 19.2% improvement over baselines. Handles unseen test cases effectively. Introduced a new benchmark dataset for Indian LSI.	Limited to IPC statutes only. Focused solely on textual judgments; lacks multimodal support (e.g., speech).

2	**Hier-SPCNet: A Legal Statute Hierarchy-based Heterogeneous Network for Computing Legal Case Document Similarity - Bhattacharya et al. (2020)**	Constructs a heterogeneous network combining case citations and statute hierarchy (Act → Chapter → Section). Applies MetaPath2Vec for node embeddings. Tested on the Indian court case corpus.	Demonstrated improved document similarity estimation by leveraging statute structure. Enhanced legal case retrieval and clustering accuracy.	Focuses on case similarity, not statute identification. Does not include multilingual or audio inputs.
3	**LEGAL-BERT: The Muppets Straight Out of Law School - Chalkidis et al. (2020)**	Compares three strategies: using generic BERT, domain-adaptive pretraining, and training BERT from scratch on legal text. Evaluated on European legal corpora (EU/UK) across multiple legal NLP tasks.	Domain-pretrained models significantly outperform off-the-shelf BERT. LEGALBERT models achieve SOTA on classification and QA tasks in legal settings.	Not trained on Indian legal texts. Vocabularies and context may not generalize well to Indian law.
4	**Pre-trained Language Models for the Legal Domain: A Case Study on Indian Law - Paul et al. (2023)**	Developed and compared three Indian legal PLMs: InLegalBERT, InCaseLawBERT, and CustomInLawBERT. Pretrained on 5.4 million Indian legal documents from Indian Kanoon. Evaluated on 5 tasks: statute identification, segmentation, outcome prediction, etc.	InLegalBERT outperforms other PLMs (including LegalBERT) on 4/5 tasks. Shows the value of country-specific corpora and vocabularies. Introduces new Indian legal benchmarks.	Does not yet support regional Indian languages. Focuses only on text, lacks audio or multimodal input support.

5	**Effective and Scalable Legal Judgment Recommendation Using Pre-learned Word Embedding - Dhanani et al. (2022)**	Uses Doc2Vec and domain-specific embeddings (e.g., Law2Vec, LegalW2V). Designed a scalable variant using Apache Spark and MapReduce for large corpora. Trained on Indian court judgments from public legal repositories.	Improves the accuracy and F1score of judgment recommendation. Demonstrates scalable training for large datasets. Addresses sparse citation limitations using semantic embedding.	Embedding-based models lack interpretability. Does not utilize graph structure or deep contextual models (e.g., BERT).
6	**IL-TUR: Benchmark for Indian Legal Text Understanding and Reasoning - Abhinav Joshi et al., 2023**	Developed IL-TUR benchmark covering 8 legal NLP tasks; datasets in English and 9 Indian languages; baseline models using LLMs and traditional methods	Introduced the first comprehensive Indian legal NLP benchmark; public leaderboard for legal text understanding and reasoning; highlighted LLM performance gaps	Dataset limited to the Indian legal system; large document length is still challenging for LLMs; scope for adding more tasks
7	**Natural Language Processing for the Legal Domain: A Survey of Tasks, Datasets, Models and Challenges - Farid Ariai et al., 2025**	Systematic review of 131 papers across legal NLP tasks: NER, Summarisation, Question Answering, Argument Mining, Judgement Prediction, etc.	Provided a broad overview of the legal NLP landscape, tasks, datasets, and models; identified 16 research challenges, including bias, explainability, and data scarcity	Survey only; no new dataset or model; primarily focuses on English language works; some sub-tasks are underexplored

8	**Autosuggestion of Relevant Cases and Statutes Pandian & Joshi (2022)**	Utilized BERT-based masked language models to recommend relevant legal statutes and case citations. Dataset: A curated collection of Indian legal queries and associated citations/statutes (details not fully public).	Semantic retrieval significantly outperformed lexical baselines for statute recommendation. Statutes were easier to retrieve accurately than prior case laws.	The approach shows weaker performance in recommending past cases. Dataset availability and generalizability to real court queries remain limited.
9	**Recommending Statutes: A Portable Method Based on Neural Representations - Yi Feng et al., 2020**	Two-step neural network system: RNN + Max Pooling + Attention + Statute2Vec embedding; trained on Chinese legal datasets	Improved statute recommendation accuracy over traditional baselines; portable method for multi-label statute prediction	Dataset limited to Chinese court data; method ignores potential legal domain knowledge graphs
10	**Statute Recommendation: Re-ranking statutes by modeling case-statute relation with interpretable hand-crafted features - Chuanyi Li et al., 2022**	Used collaborative filtering to retrieve candidate statutes; applied hand-crafted relational features with pairwise ranking models; tested on Chinese legal data	Achieved ~5% recall improvement over collaborative filtering; an interpretable feature set helps understand the case-statute relation	Focused on divorce cases in Chinese courts; not generalized to broader legal domains or jurisdictions

11	**IndianBailJudgments-1200: A Multi-Attribute Legal NLP Dataset for Bail Order Understanding in India Deshmukh & Kamble, 2025**	Dataset of 1200 Indian bail judgments annotated using prompt-engineered GPT-4o; validated subset manually; 20+ structured fields designed via schema for legal NLP	First dataset focused on Indian bail jurisprudence; enables tasks like fairness analysis, legal summarization, judgment prediction; supports legal, social, and AI research	Limited to High Court decisions; annotations are partly subjective; relies heavily on LLMs; English-only cases; lacks lower court coverage
12	**Named Entity Recognition in Indian court judgments Prathamesh Kalamkar et al., 2022**	Created a legal NER corpus of 46,545 annotated entities from Indian court judgments (14 legal entity types); trained baseline NER models (Roberta + Transition Parser & Fine-Tuned Transformers); dataset includes Supreme & High Court judgments	Developed legal NER baseline models with high F1 (91.1%); introduced post-processing techniques for coreference resolution and document-level corrections; released datasets and models publicly	Sentence-level annotation limits context; issues in long entity spans (e.g., precedents); English-only judgments; biases possible due to source sampling and heuristic case classification
13	**INDICourtSumm: A Dataset for Judgment Summarization of Indian Court Cases Abhishek Joshi et al., 2023**	Created INDICourtSumm with ~31,000 Indian court judgments and summaries from IndianKanoon; explored summarization using PEGASUS and BART models with legal-domain adaptations	First large-scale summarization dataset for the Indian legal domain; PEGASUS fine-tuned on INDICourtSumm outperformed baselines (ROUGE-1 up to 57.4)	The dataset has noisy summaries from IndianKanoon; it lacks gold-standard manual summaries; is focused only on extractive

				summaries (not abstractive generation or reasoning)
14	**ILDC for CJPE: Indian Legal Documents Corpus for Court Judgment Prediction and Explanation Malik et al., 2021**	Introduced ILDC corpus with 35K Indian court cases (Supreme and High Courts); annotated for judgment outcome; built classifiers (SVM, BiLSTM, BERT) for judgment prediction and explanation	Proposed benchmark dataset for CJPE tasks in Indian law; fine-tuned Legal-BERT achieved 81% F1 for binary outcome prediction; introduced rationale-based explanation via attention weights	Focused on outcome classification only; explanation limited to token-level attention (may not reflect true rationale); lacks detailed case reasoning or multi-label outcomes
15	**Legal Case Document Summarization: Extractive and Abstractive Methods and Their Evaluation, Bhattacharya et al., 2022**	Utilized transformer models like BERT, GPT-2, and LED for both extractive and abstractive summarization. Benchmarked using the Indian Legal Documents Corpus (ILDC) and INTACT datasets.	Introduced a comprehensive evaluation of summarization techniques on legal data. Found that the LED (Longformer) performed well for long documents. Released benchmark datasets and evaluation results.	Abstractive models still struggle with factual consistency; performance varies significantly with document length and type. Limited multilingual evaluation.

4. PROBLEM DESCRIPTION

India's judiciary is one of the largest in the world, with millions of cases pending across multiple levels of courts. It faces a critical challenge in mapping unstructured case descriptions to relevant statutory provisions. The process is currently manual, time-consuming, and highly dependent on legal expertise. This challenge is intensified in:

- **Multilingual settings**, where court proceedings and judgments are delivered in regional languages (Hindi, Marathi, Tamil, Kannada, etc.), which English-only tools cannot process effectively.
- **Speech-based proceedings**, where oral dictations and recordings need transcription and interpretation before statute identification.
- **Resource-limited courts**, especially in rural areas, where access to digital legal databases and English-trained professionals is scarce.

Thus, there is a critical need for an automated, multilingual, and speech-enabled Legal Statute Recommendation System that can:

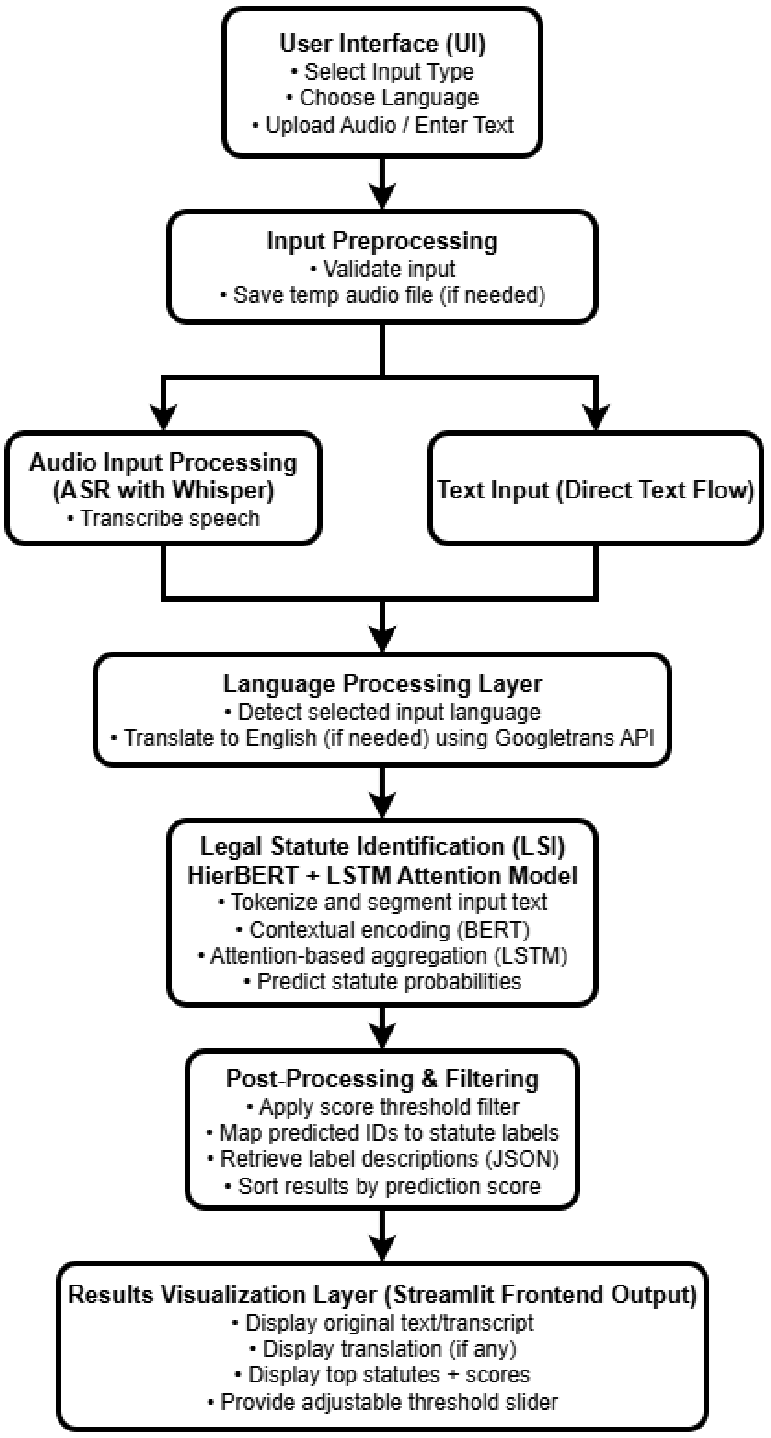
- Understand case descriptions in multiple Indian languages (both text and audio).
- Recommend the most relevant statutory provisions with confidence scores.
- Support judges, lawyers, students, and citizens in quickly identifying applicable laws.

To address this, the proposed system integrates three key components:

1. **Audio-to-Speech Processing** – Converting spoken case narratives into accurate text transcripts using ASR models like Whisper/IndicTTS.
2. **Multilingual Translation & Normalization** – Detecting input language and ensuring semantic consistency across English and Indian languages using models like IndicBERT, MuRIL, and mBERT.
3. **Legal Statute Identification** – Leveraging domain-specific NLP models (InLegalBERT, LeSICiN) to automatically recommend the most relevant statutes with ranked outputs and confidence scores.

This pipeline enables automated, multilingual, and speech-enabled statute recommendation, reducing delays, enhancing accessibility, and supporting practitioners, students, and citizens in navigating India's complex legal system.

4.1 Framework Diagram



User Interface (UI)
• Select Input Type
• Choose Language
• Upload Audio / Enter Text
Input Preprocessing
• Validate input
• Save temp audio file (if needed)
Audio Input Processing (ASR with Whisper)
• Transcribe speech
Text Input (Direct Text Flow)
Language Processing Layer
• Detect selected input language
• Translate to English (if needed) using Googletrans API
Legal Statute Identification (LSI) HierBERT + LSTM Attention Model
• Tokenize and segment input text
• Contextual encoding (BERT)
• Attention-based aggregation (LSTM)
• Predict statute probabilities
Post-Processing & Filtering
• Apply score threshold filter
• Map predicted IDs to statute labels
• Retrieve label descriptions (JSON)
• Sort results by prediction score
Results Visualization Layer (Streamlit Frontend Output)
• Display original text/transcript
• Display translation (if any)
• Display top statutes + scores
• Provide adjustable threshold slider

4.2 Pseudo Code of Proposed System

```
BEGIN

    IMPORT required libraries
        (Streamlit, Torch, Whisper, Transformers, etc.)

    DEFINE language mappings and translator

    CLASS LstmAttn:
        FUNCTION forward():
            Apply bidirectional LSTM with attention mechanism

    CLASS HierBert:
        FUNCTION forward():
            Encode input segments using BERT
            Aggregate segment representations via LSTM with attention

    CLASS HierBertForTextClassification:
        FUNCTION forward():
            Encode text hierarchically
            Apply classification head with sigmoid activation

    FUNCTION load_lsi_model_and_tokenizer():
        Load label vocabulary, tokenizer, and pretrained model
        Initialize hierarchical BERT and classification head
        Load fine-tuned model weights
        RETURN model, tokenizer, id2label mapping

    FUNCTION load_label_descriptions():
        Load JSON file containing label descriptions
        RETURN dictionary of label descriptions

    FUNCTION load_asr_pipeline():
        Load Whisper ASR model for automatic speech recognition
        RETURN ASR model

    FUNCTION transcribe_audio(audio_file, asr_model, language_code):
        Save audio file temporarily
        Call asr_model.transcribe(language_code)
        Concatenate transcript segments
        RETURN transcribed text

    FUNCTION translate_to_english_googletrans(text):
        Translate text to English using Google Translator API
        RETURN translated text

    FUNCTION get_predictions(text, model, tokenizer, id2label, label_dict,
threshold):
        Split text into sentences
        Tokenize input
        Pass through model to obtain logits
        Apply sigmoid activation and filter scores above threshold
        Sort results by confidence score
        RETURN top 10 predicted statutes with labels

    FUNCTION main():
        Initialize Streamlit web interface
        Load all models and required resources
        Display sidebar parameters (threshold, input type, language)
        Obtain input (text or audio)

        IF input is audio:
            Transcribe audio to text
        IF input language ≠ English:
            Translate text to English

        Generate predictions using model
        Display output results as formatted table

    CALL main()

END
```

4.3 Explanation of Logic

- The system starts with raw input (text/audio).
- If it’s audio, ASR converts it into text.
- Language detection and translation ensure that all inputs are standardized.
- Preprocessing normalizes legal text.
- InLegalBERT extracts statute relevance.
- LeSICiN graph boosts accuracy via citation relationships.
- Final output = ranked statutes with scores.

4.4 Flow Diagram

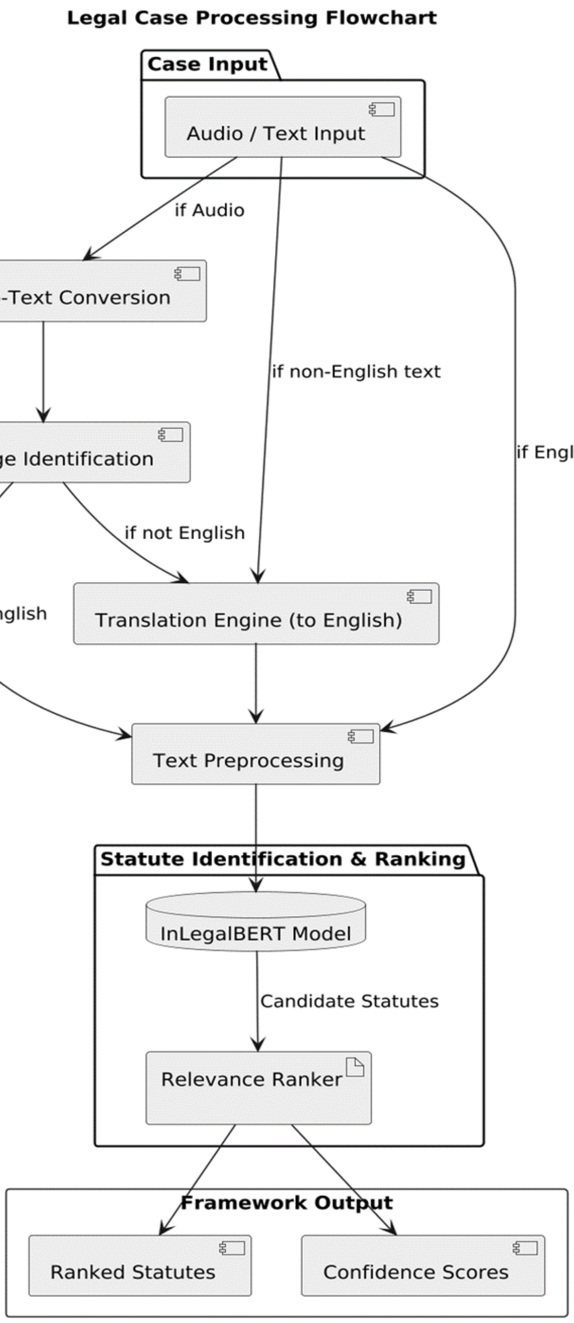


Figure 3. Stepwise pipeline of the Legal Statute Identification framework.

1. Case Input

- Audio / Text Input:
    - The system accepts case descriptions either as spoken audio (e.g., courtroom dictation, lawyer narration) or as typed text (e.g., judgment documents).
    - This ensures flexibility, especially for rural or regional courts where proceedings may not always be recorded in English text.

2. Speech-to-Text Conversion (ASR)

- If the input is audio, it passes through an Automatic Speech Recognition (ASR) system such as Whisper or IndicTTS.
- The ASR converts spoken language (Hindi, Tamil, Marathi, Kannada, etc.) into raw text.
- This step is crucial for handling oral dictations, interviews, and voice-based inputs.

3. Language Identification

- Once text is available (either directly from input or via ASR), the system uses a language detection model to identify the input language.
- This helps the pipeline decide whether translation is needed.
- Example: If the case text is in Hindi, the system knows it must translate it into English before statute identification.

4. Translation Engine (to English)

- If the detected language is not English, the system uses translation models (IndicBERT, MuRIL, mBERT) to convert the text into standardized English.
- If the text is already in English, this step is skipped.
- Ensures all case narratives are semantically consistent before passing them to the statute identification stage.

5. Text Preprocessing

- Raw text often contains noise, repetitions, or informal phrasing.
- Preprocessing includes:
  - Removing stopwords and irrelevant tokens.
  - Normalizing legal terms (e.g., “IPC Sec. 307” → “Section 307 IPC”).
  - Handling spelling and formatting differences.
- Produces a clean, normalized case description ready for statute analysis.

6. Statute Identification & Ranking

This is the core of your system, consisting of two submodules:

1. **InLegalBERT Model**
   - A transformer-based legal NLP model, fine-tuned on millions of Indian legal documents.
   - Extracts the semantic meaning of the case description and generates candidate statutes that may apply.
2. **Relevance Ranker (LeSICiN Graph / Citation Network)**
   - Uses legal citation networks to refine the recommendations.
   - Ensures statutes aren’t just textually similar but also contextually and legally relevant (based on how statutes are cited in past cases).
   - Ranks statutes in order of importance.

7. Framework Output

- The final output is presented as:
  - **Ranked Statutes**: Ordered list of the most relevant legal provisions (e.g., IPC Sec 302, CrPC Sec 164).
  - **Confidence Scores**: Probabilities indicating how strongly each statute applies.

    - **Explanations**: A natural language justification for why the statute is suggested, improving interpretability.

5. EXPERIMENTS

5.1. Dataset

The project uses three types of datasets, each aligned with a module in the framework:

1. Speech-to-Text Dataset

Source: ULCA ASR Dataset Corpus

Link: https://github.com/Open-Speech-EkStep/ULCA-asr-dataset-corpus

Parameters

- audio_id: Unique identifier for each audio file.
- audio_file: Path/link to the audio file (speech recording).
- Language: Language of the audio (Hindi, Marathi, Tamil, Kannada, etc.).
- transcript: Ground truth text transcription of the audio.
- duration: Length of the audio file in seconds.
- speaker_id (if available): Identifier for the speaker (for voice diversity).

Sample Dataset

audio_id	**audio_file**	**language**	**transcript**	**duration**	**speaker_id**
A001	hindi_case1.wav	Hindi	आरोपी ने चोरी की घटना की…	12.4	S001

A002	marathi_case2.wav	Marathi	प्रतिवादीने गंभीर इजा केली…	10.1	S002
A003	tamil_case3.wav	Tamil	குற்றவாளி காயப்படுத்தினார் …	14.8	S003

Explanation

This dataset provides multilingual audio recordings with corresponding human-verified transcripts. It is used to train and evaluate ASR models (Whisper, IndicTTS) for accurate speech-to-text conversion in multiple Indian languages.

2. Translation Dataset

Source: Hugging Face Translation Datasets (based on OPUS, Flores, etc.)

Link: https://huggingface.co/docs/transformers/en/tasks/translation

Parameters

- sentence_id: Unique identifier for each example.
- source_language: Language of the input text.
- target_language: Translation target (English).
- source_text: Original text in source language.
- translated_text: Human-translated equivalent in target language.

Sample Dataset

sentence_id	source_language	target_language	source_text	translated_text
T001	Hindi	English	आरोपी ने चोरी की घटना की…	The accused committed theft…
T002	Tamil	English	குற்றவாளி காயப்படுத்தினார் …	The accused caused injury…
T003	Kannada	English	ಆರೋಪಿ ಹಲ್ಲೆ ನಡೆಸಿದನು…	The accused carried out an assault…

Explanation

This dataset contains parallel sentences in Indian languages and English. It is used to train/fine-tune IndicBERT, MuRIL, and mBERT for multilingual translation, ensuring all case inputs are normalized into English before legal analysis.

3. Legal Statute Identification Dataset (LeSICiN)

Source: LeSICiN: Legal Statute Identification using Citation Networks

Link: https://zenodo.org/records/6053791

Parameters

- case_id: Unique identifier for a case document.
- case_text: Case description or judgment text.

- citations: Statutes or prior cases cited in the judgment.
- statutes_applicable: Ground truth statutes applied (labels).
- predicted_statutes (for evaluation): Model output of recommended statutes.

Sample Dataset

case_id	**case_text**	**citations**	**statutes_applicable**
C001	The accused broke into the house and stole	IPC Sec 378, 411	IPC Sec 378, Sec 411
C002	The husband abandoned and tortured his wife	IPC Sec 498A	IPC Sec 498A
C003	The defendant attacked with a deadly weapon	IPC Sec 307	IPC Sec 307

Explanation

This dataset links Indian court judgments to statutes they cite or are influenced by. It is graph-based, combining textual content with citation networks. This allows the system to recommend statutes not only by semantic similarity but also by legal reasoning and precedent.

Combined Dataset Role in Project

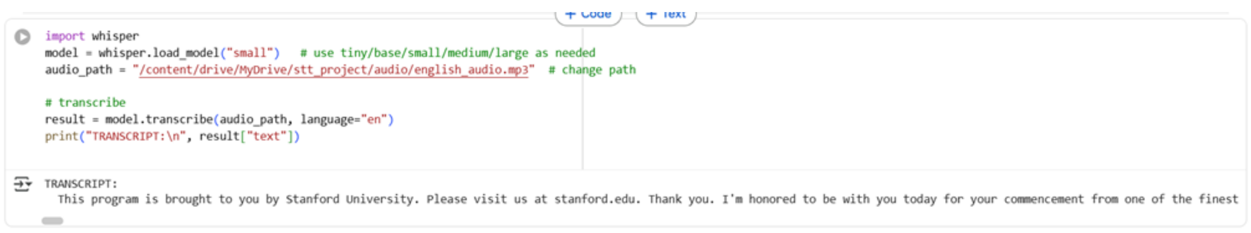
- **ULCA ASR Dataset** → Used for Audio-to-Text Conversion.
- **Translation Dataset** → Used for Multilingual Normalization.
- **LeSICiN Dataset** → Used for Statute Identification & Ranking.

Together, these datasets cover the entire pipeline: speech input → translation → statute recommendation.

6. RESULTS AND DISCUSSION

6.1 Google Colab

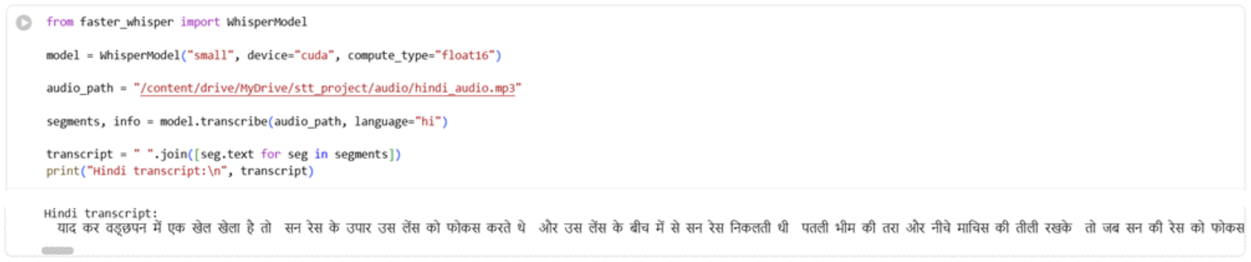
Transcription Output



```
import whisper
model = whisper.load_model("small")   # use tiny/base/small/medium/large as needed
audio_path = "/content/drive/MyDrive/stt_project/audio/english_audio.mp3"  # change path

# transcribe
result = model.transcribe(audio_path, language="en")
print("TRANSCRIPT:\n", result["text"])
```

```
TRANSCRIPT:
  This program is brought to you by Stanford University. Please visit us at stanford.edu. Thank you. I'm honored to be with you today for your commencement from one of the finest
```

Figure 4: Speech-to-text transcription using the Whisper model in Python. The model successfully transcribes an English audio clip into text.

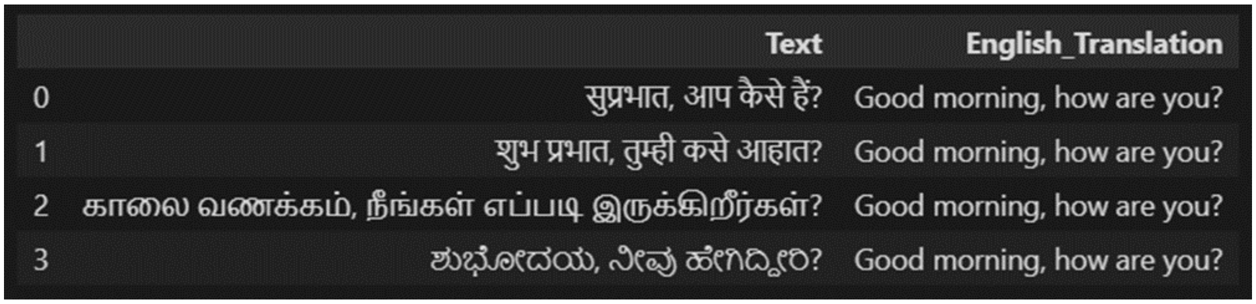


```
from faster_whisper import WhisperModel

model = WhisperModel("small", device="cuda", compute_type="float16")

audio_path = "/content/drive/MyDrive/stt_project/audio/hindi_audio.mp3"

segments, info = model.transcribe(audio_path, language="hi")

transcript = " ".join([seg.text for seg in segments])
print("Hindi transcript:\n", transcript)
```

```
Hindi transcript:
  याद कर वड्छपन में एक खेल खेला है तो  सन रेस के उपार उस लेंस को फोकस करते थे  और उस लेंस के बीच में से सन रेस निकलती थी  पतली भीम की तरा और नीचे माचिस की तीली रखके  तो जब सन की रेस को फोकस
```

Figure 5: Transcription of Hindi audio using the Faster Whisper model. The model processes an audio clip in Hindi and outputs the corresponding text transcription.

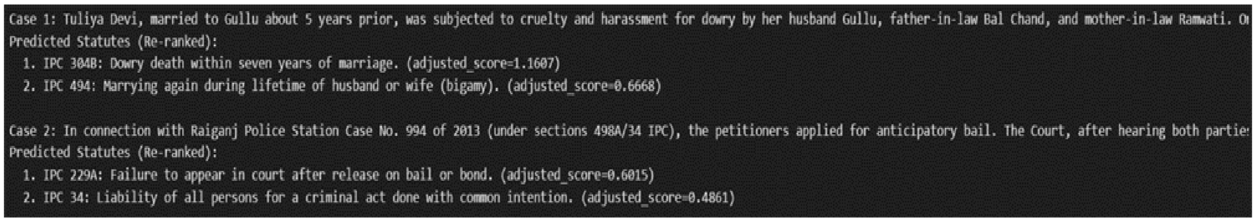
Translation Output



	Text	English_Translation
0	सुप्रभात, आप कैसे हैं?	Good morning, how are you?
1	शुभ प्रभात, तुम्ही कसे आहात?	Good morning, how are you?
2	காலை வணக்கம், நீங்கள் எப்படி இருக்கிறீர்கள்?	Good morning, how are you?
3	ಶುಭೋದಯ, ನೀವು ಹೇಗಿದ್ದೀರಿ?	Good morning, how are you?

Figure 6: Multilingual text samples and their corresponding English translations demonstrating the language translation module's functionality.

Legal Statute Identification Output

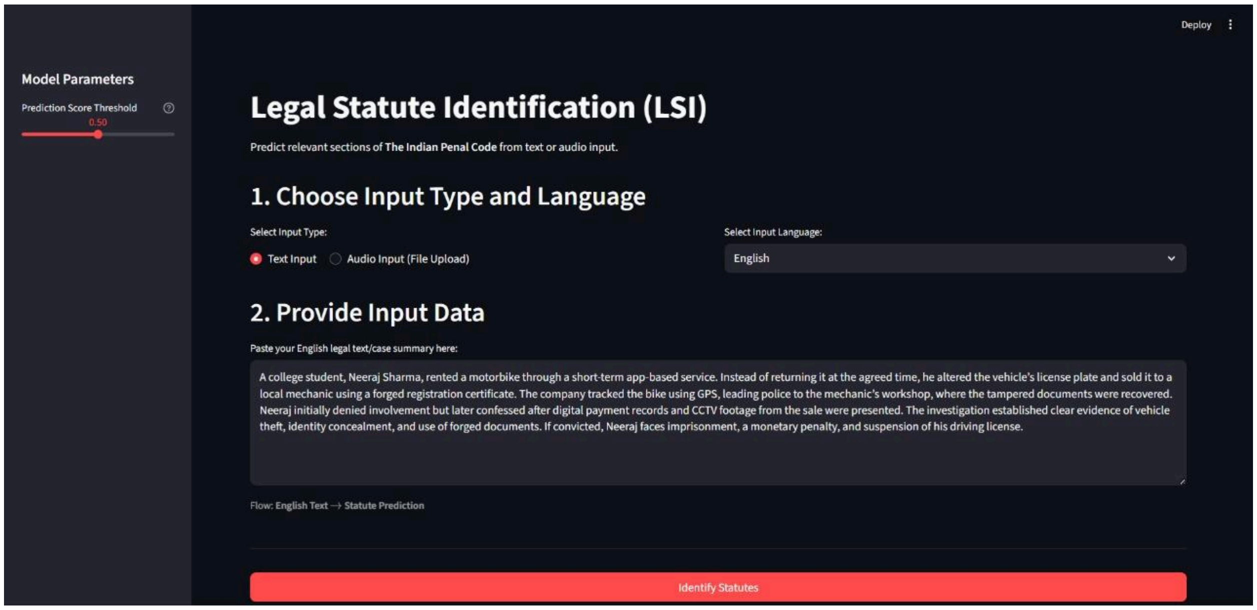


```
Case 1: Tuliya Devi, married to Gullu about 5 years prior, was subjected to cruelty and harassment for dowry by her husband Gullu, father-in-law Bal Chand, and mother-in-law Ramwati. O
Predicted Statutes (Re-ranked):
  1. IPC 304B: Dowry death within seven years of marriage. (adjusted_score=1.1607)
  2. IPC 494: Marrying again during lifetime of husband or wife (bigamy). (adjusted_score=0.6668)

Case 2: In connection with Raiganj Police Station Case No. 994 of 2013 (under sections 498A/34 IPC), the petitioners applied for anticipatory bail. The Court, after hearing both partie
Predicted Statutes (Re-ranked):
  1. IPC 229A: Failure to appear in court after release on bail or bond. (adjusted_score=0.6015)
  2. IPC 34: Liability of all persons for a criminal act done with common intention. (adjusted_score=0.4861)
```

Figure 7: Predicted legal statutes for sample case texts after re-ranking using the proposed Legal Statute Identification (LSI) model.

6.2 GUI Interface

English Text



Legal Statute Identification (LSI)

Predict relevant sections of **The Indian Penal Code** from text or audio input.

1. Choose Input Type and Language

Select Input Type: [✓ Text Input] Audio Input (File Upload)

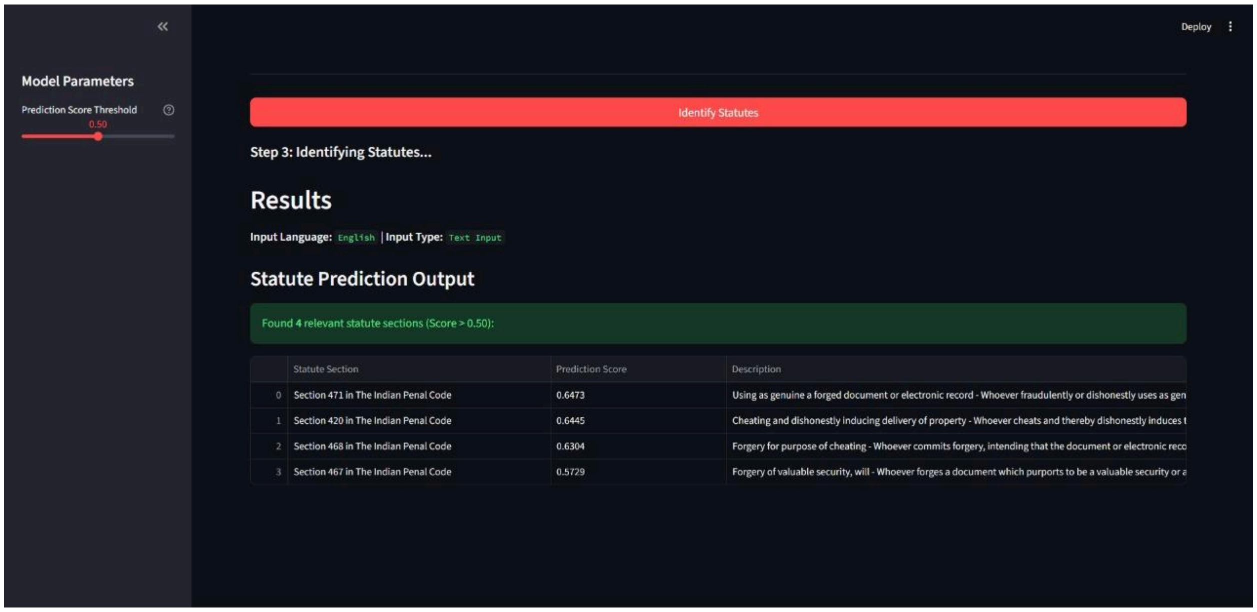
Select Input Language: English

2. Provide Input Data

Paste your English legal text/case summary here:

A college student, Neeraj Sharma, rented a motorbike through a short-term app-based service. Instead of returning it at the agreed time, he altered the vehicle's license plate and sold it to a local mechanic using a forged registration certificate. The company tracked the bike using GPS, leading police to the mechanic's workshop, where the tampered documents were recovered. Neeraj initially denied involvement but later confessed after digital payment records and CCTV footage from the sale were presented. The investigation established clear evidence of vehicle theft, identity concealment, and use of forged documents. If convicted, Neeraj faces imprisonment, a monetary penalty, and suspension of his driving license.

Flow: English Text → Statute Prediction



Step 3: Identifying Statutes...

Results

Input Language: `English` | **Input Type:** `Text Input`

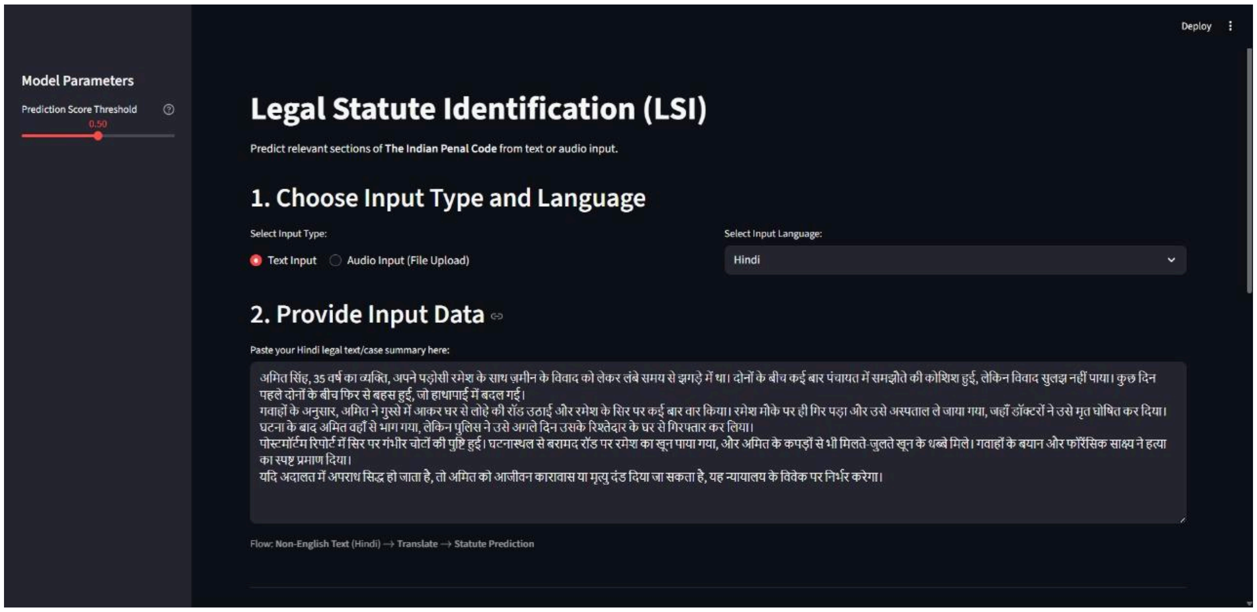
Statute Prediction Output

Found 4 relevant statute sections (Score > 0.50):

	Statute Section	Prediction Score	Description
0	Section 471 in The Indian Penal Code	0.6473	Using as genuine a forged document or electronic record - Whoever fraudulently or dishonestly uses as gen
1	Section 420 in The Indian Penal Code	0.6445	Cheating and dishonestly inducing delivery of property - Whoever cheats and thereby dishonestly induces t
2	Section 468 in The Indian Penal Code	0.6304	Forgery for purpose of cheating - Whoever commits forgery, intending that the document or electronic reco
3	Section 467 in The Indian Penal Code	0.5729	Forgery of valuable security, will - Whoever forges a document which purports to be a valuable security or a

Figure 8. Workflow demonstration of the Legal Statute Identification (LSI) model. (Top) A user provides a case summary in natural language as text input. (Bottom) The system analyzes the text and returns a ranked list of predicted statutes, including sections 471, 420, 468, and 467 of the Indian Penal Code, based on the input summary.

Non-English text



Legal Statute Identification (LSI)

Predict relevant sections of **The Indian Penal Code** from text or audio input.

1. Choose Input Type and Language

Select Input Type: [● Text Input] ○ Audio Input (File Upload)

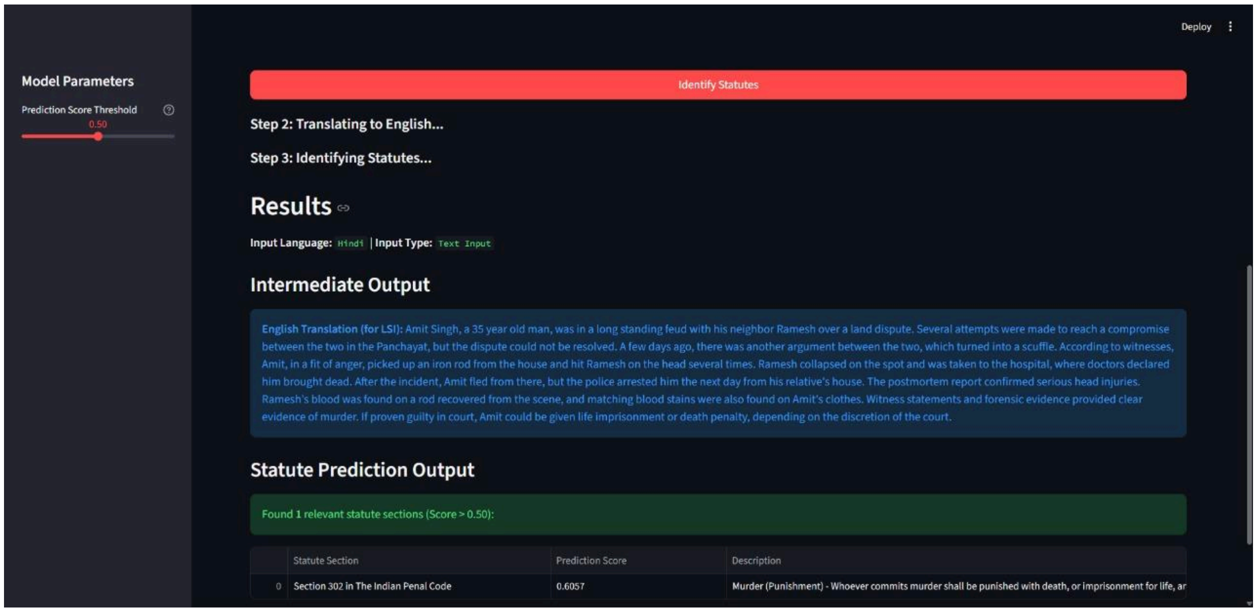
Select Input Language: Hindi

2. Provide Input Data

Paste your Hindi legal text/case summary here:

अमित सिंह, 35 वर्ष का व्यक्ति, अपने पड़ोसी रमेश के साथ ज़मीन के विवाद को लेकर लंबे समय से झगड़े में था। दोनों के बीच कई बार पंचायत में समझौते की कोशिश हुई, लेकिन विवाद सुलझ नहीं पाया। कुछ दिन पहले दोनों के बीच फिर से बहस हुई, जो हाथापाई में बदल गई।
गवाहों के अनुसार, अमित ने गुस्से में आकर घर से लोहे की रॉड उठाई और रमेश के सिर पर कई बार वार किया। रमेश मौके पर ही गिर पड़ा और उसे अस्पताल ले जाया गया, जहाँ डॉक्टरों ने उसे मृत घोषित कर दिया। घटना के बाद अमित वहाँ से भाग गया, लेकिन पुलिस ने उसे अगले दिन उसके रिश्तेदार के घर से गिरफ्तार कर लिया।
पोस्टमॉर्टम रिपोर्ट में सिर पर गंभीर चोटों की पुष्टि हुई। घटनास्थल से बरामद रॉड पर रमेश का खून पाया गया, और अमित के कपड़ों से भी मिलते-जुलते खून के धब्बे मिले। गवाहों के बयान और फॉरेंसिक साक्ष्य ने हत्या का स्पष्ट प्रमाण दिया।
यदि अदालत में अपराध सिद्ध हो जाता है, तो अमित को आजीवन कारावास या मृत्यु दंड दिया जा सकता है, यह न्यायालय के विवेक पर निर्भर करेगा।

Flow: Non-English Text (Hindi) → Translate → Statute Prediction



Step 2: Translating to English...

Step 3: Identifying Statutes...

Results

Input Language: `Hindi` | **Input Type:** `Text Input`

Intermediate Output

English Translation (for LSI): Amit Singh, a 35 year old man, was in a long standing feud with his neighbor Ramesh over a land dispute. Several attempts were made to reach a compromise between the two in the Panchayat, but the dispute could not be resolved. A few days ago, there was another argument between the two, which turned into a scuffle. According to witnesses, Amit, in a fit of anger, picked up an iron rod from the house and hit Ramesh on the head several times. Ramesh collapsed on the spot and was taken to the hospital, where doctors declared him brought dead. After the incident, Amit fled from there, but the police arrested him the next day from his relative's house. The postmortem report confirmed serious head injuries. Ramesh's blood was found on a rod recovered from the scene, and matching blood stains were also found on Amit's clothes. Witness statements and forensic evidence provided clear evidence of murder. If proven guilty in court, Amit could be given life imprisonment or death penalty, depending on the discretion of the court.

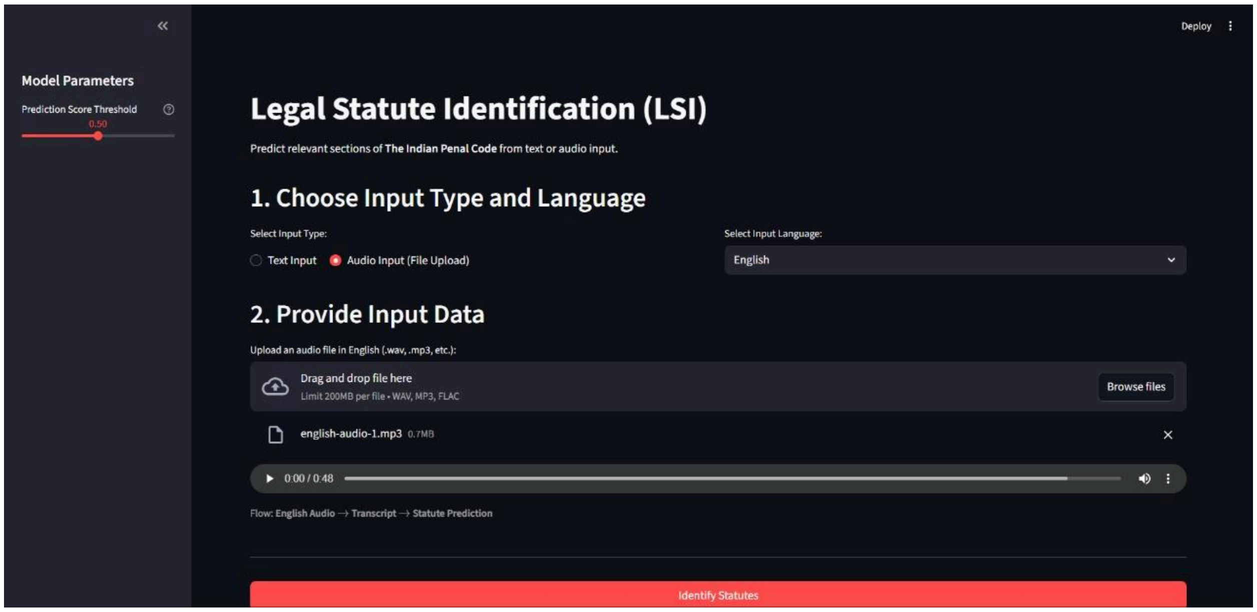
Statute Prediction Output

Found 1 relevant statute sections (Score > 0.50):

	Statute Section	Prediction Score	Description
0	Section 302 in The Indian Penal Code	0.6057	Murder (Punishment) - Whoever commits murder shall be punished with death, or imprisonment for life, ar

Figure 9. Workflow for non-English input in the Legal Statute Identification (LSI) system. (Top) A user inputs a case summary in Hindi. (Bottom) The system first performs an intermediate translation of the Hindi text into English, then analyzes the translation to identify and output the most relevant statute, Section 302 (Murder) of the Indian Penal Code.

English Audio



Legal Statute Identification (LSI)

Predict relevant sections of **The Indian Penal Code** from text or audio input.

1. Choose Input Type and Language

Select Input Type: Text Input [✓ Audio Input (File Upload)]

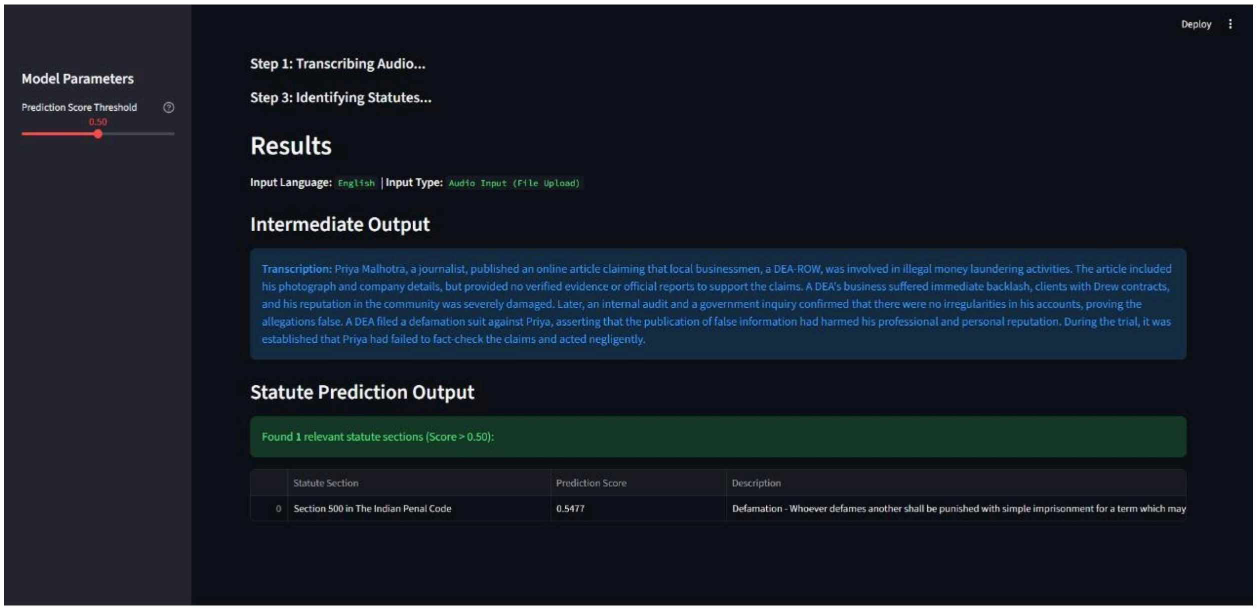
Select Input Language: English

2. Provide Input Data

Upload an audio file in English (.wav, .mp3, etc.):

Drag and drop file here
Limit 200MB per file • WAV, MP3, FLAC
Browse files
english-audio-1.mp3 0.7MB
0:00 / 0:48

Flow: English Audio → Transcript → Statute Prediction



Step 1: Transcribing Audio...

Step 3: Identifying Statutes...

Results

Input Language: `English` | **Input Type:** `Audio Input (File Upload)`

Intermediate Output

Transcription: Priya Malhotra, a journalist, published an online article claiming that local businessmen, a DEA-ROW, was involved in illegal money laundering activities. The article included his photograph and company details, but provided no verified evidence or official reports to support the claims. A DEA's business suffered immediate backlash, clients with Drew contracts, and his reputation in the community was severely damaged. Later, an internal audit and a government inquiry confirmed that there were no irregularities in his accounts, proving the allegations false. A DEA filed a defamation suit against Priya, asserting that the publication of false information had harmed his professional and personal reputation. During the trial, it was established that Priya had failed to fact-check the claims and acted negligently.

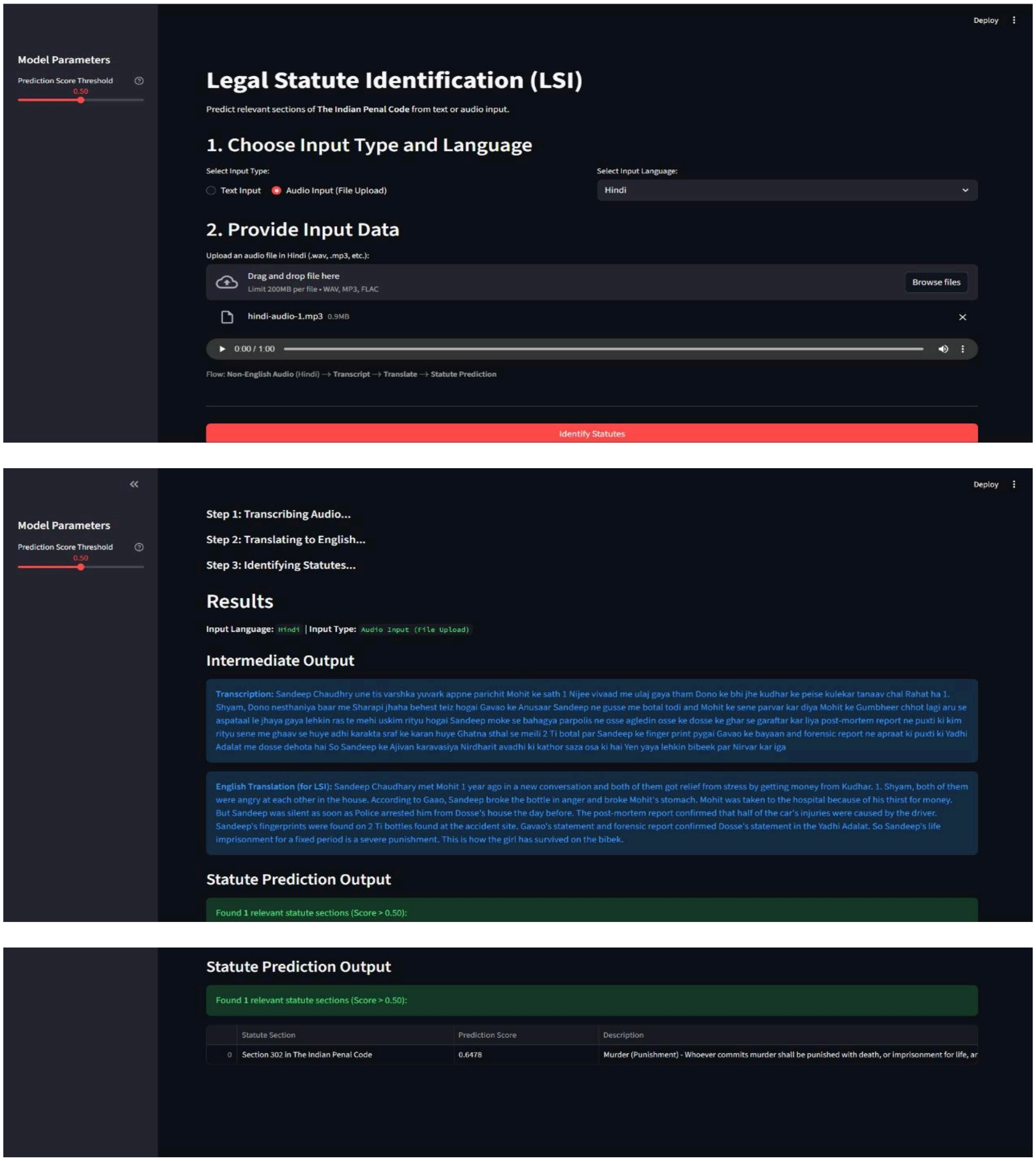
Statute Prediction Output

Found **1** relevant statute sections (Score > 0.50):

	Statute Section	Prediction Score	Description
0	Section 500 in The Indian Penal Code	0.5477	Defamation - Whoever defames another shall be punished with simple imprisonment for a term which may

Figure 10. Demonstration of the audio input module for the Legal Statute Identification (LSI) system. (Top) A user uploads an English audio file. (Bottom) The results panel shows the two-step process: first, an 'Intermediate Output' displaying the system's transcription of the audio, and second, the 'Statute Prediction Output' (Section 500 IPC for Defamation) based on the transcribed text.

Non-English Audio:


Deploy

Model Parameters

Prediction Score Threshold 0.50

Legal Statute Identification (LSI)

Predict relevant sections of **The Indian Penal Code** from text or audio input.

1. Choose Input Type and Language

Select Input Type: Text Input [✓ Audio Input (File Upload)]

Select Input Language: Hindi

2. Provide Input Data

Upload an audio file in Hindi (.wav, .mp3, etc.):

Drag and drop file here
Limit 200MB per file • WAV, MP3, FLAC
Browse files

hindi-audio-1.mp3 0.9MB

0:00 / 1:00

Flow: **Non-English Audio** (Hindi) → **Transcript** → **Translate** → **Statute Prediction**

Identify Statutes

Deploy

Model Parameters

Prediction Score Threshold 0.50

Step 1: Transcribing Audio...

Step 2: Translating to English...

Step 3: Identifying Statutes...

Results

Input Language: `Hindi` | **Input Type:** `Audio Input (File Upload)`

Intermediate Output

Transcription: Sandeep Chaudhry une tis varshka yuvark appne parichit Mohit ke sath 1 Nijee vivaad me ulaj gaya tham Dono ke bhi jhe kudhar ke peise kulekar tanaav chal Rahat ha 1. Shyam, Dono nesthaniya baar me Sharapi jhaha behest teiz hogai Gavao ke Anusaar Sandeep ne gusse me botal todi and Mohit ke sene parvar kar diya Mohit ke Gumbheer chhot lagi aru se aspataal le jhaya gaya lehkin ras te mehi uskim rityu hogai Sandeep moke se bahagya parpolis ne osse agledin osse ke dosse ke ghar se garaftar kar liya post-mortem report ne puxti ki kim rityu sene me ghaav se huye adhi karakta sraf ke karan huye Ghatna sthal se meili 2 Ti botal par Sandeep ke finger print pygai Gavao ke bayaan and forensic report ne apraat ki puxti ki Yadhi Adalat me dosse dehota hai So Sandeep ke Ajivan karavasiya Nirdharit avadhi ki kathor saza osa ki hai Yen yaya lehkin bibeek par Nirvar kar iga

English Translation (for LSI): Sandeep Chaudhary met Mohit 1 year ago in a new conversation and both of them got relief from stress by getting money from Kudhar. 1. Shyam, both of them were angry at each other in the house. According to Gaao, Sandeep broke the bottle in anger and broke Mohit's stomach. Mohit was taken to the hospital because of his thirst for money. But Sandeep was silent as soon as Police arrested him from Dosse's house the day before. The post-mortem report confirmed that half of the car's injuries were caused by the driver. Sandeep's fingerprints were found on 2 Ti bottles found at the accident site. Gavao's statement and forensic report confirmed Dosse's statement in the Yadhi Adalat. So Sandeep's life imprisonment for a fixed period is a severe punishment. This is how the girl has survived on the bibek.

Statute Prediction Output

Found 1 relevant statute sections (Score > 0.50):

Statute Prediction Output

Found 1 relevant statute sections (Score > 0.50):

	Statute Section	Prediction Score	Description
0	Section 302 in The Indian Penal Code	0.6478	Murder (Punishment) - Whoever commits murder shall be punished with death, or imprisonment for life, ar

Figure 11. Workflow for non-English audio input in the Legal Statute Identification (LSI) system. (Top) A user uploads an audio file in Hindi. (Bottom) The results panel demonstrates the complete pipeline: an 'Intermediate Output' of the original Hindi transcription, followed by an

'English Translation (for LSI)', and concluding with the 'Statute Prediction Output' (Section 302 IPC) based on the translation.

7. CONCLUSION

This project proposed and demonstrated a Multilingual Legal Statute Recommendation System designed to assist India's judiciary, legal professionals, and citizens in efficiently identifying statutory provisions relevant to case descriptions. The system addresses critical challenges such as manual statute mapping, language barriers, and the lack of accessibility in rural courts.

Key findings and contributions include:

- **Speech-to-Text Integration**: Implemented Automatic Speech Recognition (ASR) using Whisper/IndicTTS to transcribe multilingual audio case inputs into text.
- **Multilingual Support**: Incorporated IndicBERT, MuRIL, and mBERT to detect and translate non-English case descriptions into standardized English, ensuring inclusivity across Hindi, Marathi, Tamil, Kannada, and English.
- **Legal Domain NLP**: Utilized InLegalBERT, a transformer model fine-tuned on Indian legal texts, combined with the LeSICiN dataset and citation networks to accurately recommend relevant statutes.
- **Ranked Statute Output**: Generated ranked lists of applicable statutes with confidence scores, improving interpretability and aiding quick decision-making.

Overall, the project demonstrates how Natural Language Processing, multilingual AI, and legal knowledge graphs can be combined to transform legal research and statutory interpretation. The framework enhances accessibility, speeds up legal aid, and provides significant value to lower courts and underrepresented communities.

By bridging the gap between speech, language diversity, and statutory complexity, this work contributes toward an intelligent, inclusive, and scalable legal recommendation system that supports equitable access to justice in India.

8. REFERENCES

[1] Paul, S., Goyal, P., & Ghosh, S. (2022). Paul, S., Goyal, P., & Ghosh, S. (2022). LeSICiN: A heterogeneous graph-based approach for automatic legal statute identification from Indian legal documents. In Proceedings of the AAAI Conference on Artificial Intelligence, 36(11), 12699-12707. https://doi.org/10.1609/aaai.v36i11.21404

[2] Bhattacharya, P., Ghosh, K., Pal, A., & Ghosh, S. (2020). Bhattacharya, P., Ghosh, K., Pal, A., & Ghosh, S. (2020). Hier-SPCNet: A legal statute hierarchy-based heterogeneous network for computing legal case document similarity. arXiv preprint arXiv:2007.03225. https://arxiv.org/abs/2007.03225

[3] Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N., & Androutsopoulos, I. (2020). Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N., & Androutsopoulos, I. (2020). LEGAL-BERT: The Muppets straight out of law school. In Findings of the Association for Computational Linguistics: EMNLP 2020 (pp. 2898–2904). Association for Computational Linguistics. https://aclanthology.org/2020.findings-emnlp.261/

[4] Paul, S., Mandal, A., Goyal, P., & Ghosh, S. (2023). Paul, S., Mandal, A., Goyal, P., & Ghosh, S. (2023). Pre-trained language models for the legal domain: A case study on Indian law. In Proceedings of the 19th International Conference on Artificial Intelligence and Law (pp. 195–199). ACM. https://doi.org/10.1145/3594536.3595133

[5] Dhanani, J., Mehta, R., & Rana, D. (2022). Effective and Scalable Legal Judgment Recommendation Using Pre-learned Word Embedding. Complex & Intelligent Systems, 8, 3199–3213. https://doi.org/10.1007/s40747-022-00673-1

[6] Ariai, F., Mackenzie, J., & Demartini, G. (2025). Natural Language Processing for the Legal Domain: A Survey of Tasks, Datasets, Models and Challenges. ACM Computing Surveys. https://doi.org/10.1145/nnnnnnn.nnnnnnn

[7] Joshi, A., Paul, S., Sharma, A., Goyal, P., Ghosh, S., & Modi, A. (2023). IL-TUR: Benchmark for Indian Legal Text Understanding and Reasoning. https://exploration-lab.github.io/IL-TUR/

[8] Pandian, S., & Joshi, S. (2022). Autosuggestion of Relevant Cases and Statutes. https://aclanthology.org/2022.findings-emnlp.831

[9] Feng, Y., Li, C., Ge, J., Luo, B., & Ng, V. (2020). Recommending Statutes: A Portable Method Based on Neural Networks. ACM Transactions on Knowledge Discovery from Data, 15(2), Article 16. https://doi.org/10.1145/3424671

[10] Li, C., Ge, J., Cheng, K., Luo, B., & Chang, V. (2022). Statute Recommendation: Re-ranking Statutes by Modeling Case-Statute Relation with Interpretable Hand-Crafted Features. Information Sciences, 607, 1023–1040. https://doi.org/10.1016/j.ins.2022.06.042

[11] Deshmukh, A., & Kamble, S. (2025). Deshmukh, A., & Kamble, S. (2025). IndianBailJudgments-1200: A Multi-Attribute Legal NLP Dataset for Bail Order Understanding in India. arXiv. https://arxiv.org/abs/2507.02506

[12] Kalamkar, P., et al. (2022). Kalamkar, P., Agarwal, A., Tiwari, A., Gupta, S., Karn, S., & Raghavan, V. (2022). Named Entity Recognition in Indian court judgments. In Proceedings of the Natural Legal Language Processing Workshop 2022 (pp. 184–193). Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.nllp-1.15

[13] Joshi, A., Pathak, N., Singh, K., & Khapra, M. M. (2023). INDICourtSumm: A Dataset for Judgment Summarization of Indian Court Cases. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (pp. 5863-5876). Association for Computational Linguistics. https://aclanthology.org/2023.emnlp-main.362/

[14] Malik, S., et al. (2021). Malik, V., Sanjay, R., Nigam, S. K., Ghosh, K., Guha, S. K., Bhattacharya, A., & Modi, A. (2021). ILDC for CJPE: Indian Legal Documents Corpus for

Court Judgment Prediction and Explanation. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers) (pp. 4046–4062). Association for Computational Linguistics. https://aclanthology.org/2021.acl-long.313/

[15] Bhattacharya, M., & Ghosh, B. (2022). Shukla, A., Bhattacharya, P., Poddar, S., Mukherjee, R., Ghosh, K., Goyal, P., & Ghosh, S. (2022). Legal Case Document Summarization: Extractive and Abstractive Methods and their Evaluation. Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (AACL-IJCNLP) (pp. 1048–1064). Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.aacl-main.77