

Criminal Code Summarization and Outcome Prediction Using InLegalBERT and Bharatiya Nyaya Sanhita Mapping

Submitted in partial fulfillment of the requirements
for the degree of

B.E. Computer Engineering

By

Samit Fernandes 03 212031

Jaden franco 04 212032

Ralph Pereira 19 212085

Guide

K. Priya Karunakaran

Assistant Professor



Department of Computer Engineering
St. Francis Institute of Technology
(Engineering College)

An Autonomous Institute, Affiliated to University of Mumbai

2024-2025


CERTIFICATE

This is to certify that the project entitled “ **Criminal Code Summarization and Outcome Prediction Using InLegalBERT and Bharatiya Nyaya Sanhita Mapping**” is a bonafide work of “**Samit Fernandes(03), Jaden Franco(04) and Ralph Pereira(19)**” submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of B.E. in Computer Engineering.



(K. Priya Karunakaran)

Guide



(Dr. Kavita Sonawane)

Head Of Department



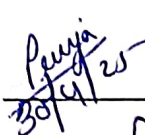
(Dr. Sincy George)

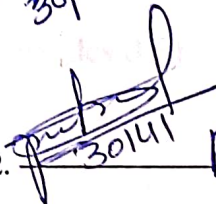
Principal

Project Approval Report for B.E.

This project report entitled "*Criminal Code Summarization and Outcome Prediction Using InLegalBERT and Bharatiya Nyaya Sanhita Mapping*" by *Samit Fernandes(03)*, *Jaden Franco(04)* and *Ralph Pereira(19)* is approved for the degree of *B.E. in Computer Engineering*.

Examiners

1.  Dr. Priyak
30/4/25

2.  Dr. Suresh P.
30/4/25

Date:

Place: Mumbai

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.



Samit Fernandes (03)



Jaden Franco (04)



Ralph Pereira (19)

Date: 30/4/25

Abstract

The Bharatiya Nyaya Sanhita (BNS) Act introduces a transformative change to India's legal system, replacing the Indian Penal Code (IPC) with a streamlined structure, which poses challenges for legal professionals reliant on IPC-based analysis tools that lack BNS-specific capabilities for document classification, summarization, and case outcome prediction. To bridge this gap, this project presents a BNS-compatible legal analysis system that includes a rule based mapping between IPC and BNS sections, supported by a synthetic dataset to model BNS cases. Document summarization is achieved by fine-tuning InLegalBERT, a language model tailored for legal contexts, enabling precise extraction of critical information, while a binary classification model predicts case outcomes, enhancing decision-making for BNS cases. Combining InLegalBERT-based summarization, structured IPC-to-BNS mapping, and outcome prediction, this solution supports legal professionals transitioning to the BNS framework and lays the groundwork for advancements in legal technology aligned with modern standards.

Keywords: *extractive summarization, abstractive summarization, hybrid summarization, IPC, BNS, InLegalBERT, BART, Indian Kanoon*

Contents

1	Introduction	1
1.1	Introduction	1
1.2	Background study Terminologies/ Definitions of new terms . . .	2
1.3	Fundamental study points of the selected topic and the domain . .	3
1.4	Identification of challenges in the selected topic	4
1.5	Problem Statement and Proposed Solution	4
1.6	Scope of the system	5
2	Review of Literature	6
2.1	Survey of Existing Systems	6
2.2	Limitations of Existing Systems	7
2.3	Motivation	8
3	Proposed System: Analysis	9
3.1	Detailed explanation of Proposed system	9
3.1.1	Working Principle	9
3.1.2	Phase/ Module - wise explanation	11
3.2	System Analysis	12
3.2.1	Functional Requirements	12
3.2.2	Non- Functional Requirements	13
3.2.3	Software and Hardware requirements	13
3.2.4	Use Case Modeling	14
3.3	Proposed System :Analysis, Modelling and Design	17
3.3.1	Class Diagram	17

3.3.2	Sequence Diagram	18
3.3.3	DFD	19
3.3.4	Architectural View	22
3.3.5	Algorithms / Methodology	25
3.3.6	UI/UX design	29
4	Implementation Plan and Experimental Set up of the Proposed system	34
4.1	Experimental Set up	34
4.1.1	Details discussion of input/Dataset	34
4.1.2	Performance Evaluation Parameters	35
4.2	Code	36
5	Proposed System: Analysis	40
5.1	Presentation and validation of the results for the proposed system .	40
5.1.1	Quantitative and Qualitative results	40
5.1.2	Document-wise Performance Evaluation	42
5.1.3	Conclusion	44
5.2	Comparative Analysis with existing systems	45
6	Conclusion	47
	Appendix-I	48
	References	49
	Acknowledgements	53

List of Figures

3.1	Work Flow Diagram For System	11
3.2	Use Case Diagram For System	14
3.3	Class Diagram of System	17
3.4	Sequence Diagram for Summarization & Case Analysis	18
3.5	Data Flow Diagram Level 0	19
3.6	Data Flow Diagram Level 1	20
3.7	Data Flow Diagram Level 2	21
3.8	BERT-Based Extractive Summarization Architecture	22
3.9	BART-Based Extractive Summarization Architecture	23
3.10	Homepage 1 of Summarization System	29
3.11	Features page of Summarization System	30
3.12	Upload the legal document that you want to summarize and analyze	31
3.13	Preview the uploaded document and select the summary options desired	31
3.14	Displays the desired summary	32
3.15	IPC Sections are extracted and its equivalent BNS details are dis- played	32
3.16	Analysis of the legal document is displayed to the user	33
3.17	All the generated summary history is stored for quick access . . .	33

List of Tables

3.1	Use case template 1 of Summarization System	15
3.2	Use case template 2 of Summarization System	16
5.1	Summary Statistics for Concise and Precise Methods	44
5.2	Evaluation Metrics for Summarization Methods (First Document)	45
5.3	Evaluation Metrics for Summarization Methods (Second Document)	45
5.4	Comparative analysis of different models for Indian legal document summarization.	46

Chapter 1

Introduction

1.1 Introduction

The introduction of the Bharatiya Nyaya Sanhita (BNS) Act marks a significant evolution in India's legal system, replacing the long-standing Indian Penal Code (IPC) with a modernized framework. While the IPC has provided a comprehensive structure for legal interpretation for over a century, its outdated language and complex categorizations have presented challenges in adapting to contemporary legal needs. The BNS Act simplifies these classifications and introduces a more streamlined approach to criminal law, aiming to enhance accessibility, efficiency, and relevancy in legal processes.

This transition, however, poses challenges for legal professionals who are accustomed to IPC based tools and methods. Existing systems and databases are structured around IPC classifications and cannot directly interpret or analyze cases under the new BNS framework. Therefore, there is a need for tools and datasets specifically tailored to the BNS Act to aid in tasks such as legal document summarization, section mapping, and case analysis. This project aims to bridge this gap by creating a rule-based mapping strategy to align IPC sections with their BNS counterparts, facilitating accurate legal document processing and analysis under the new legal framework.

1.2 Background study Terminologies/ Definitions of new terms

The shift from IPC to BNS represents a monumental update in India's criminal law, aiming to address the complexities and ambiguities present in the older legal code. While the IPC's layered classifications and terminology were designed to capture the nuances of various criminal acts, these intricacies often hinder quick and clear interpretation, especially in a digitalized, fast-paced world. The BNS Act simplifies legal codes to make them more accessible and comprehensible, particularly by removing outdated language and reorganizing sections into clearer, more straightforward categories.

One of the main challenges associated with this transition is the absence of BNS-specific tools and databases. Existing legal analysis tools rely on IPC-based structures and lack the adaptability required for the updated BNS framework. Additionally, without a dedicated dataset for BNS, there is no reliable resource for legal professionals to analyze cases under the new act. This project addresses these issues by creating a synthetic dataset that aligns IPC sections with their BNS equivalents and by developing tools tailored for BNS-based legal document summarization, analysis, and interpretation.

Terminologies:

- **Bharatiya Nyaya Sanhita (BNS) Act:** A recently introduced legal framework that replaces the Indian Penal Code in India. The BNS Act modernizes criminal law by simplifying language, streamlining sections, and addressing ambiguities to create a more accessible and efficient legal structure.
- **Extractive Summarization:** A technique in document summarization that generates concise summaries by selecting key sentences or phrases directly from the original text, preserving essential legal language and details.

- **Rule-based Mapping Strategy:** A systematic method to associate specific IPC sections with corresponding sections in the BNS Act. This ensures accurate and consistent interpretation of cases under the BNS framework.

1.3 Fundamental study points of the selected topic and the domain

- **Transition in Legal Frameworks:** This project explores the significant changes brought by the BNS Act and the implications of replacing the IPC. It aims to provide continuity in legal analysis while supporting the new standards and classifications introduced by BNS.
- **Document Summarization for Legal Analysis:** Summarization of lengthy legal documents is essential for efficient processing and decision-making. This study focuses on extractive summarization techniques that reduce document length while retaining key information, facilitating quick interpretation by legal professionals.
- **Case Analysis in the BNS Context:** This project incorporates a comprehensive analysis of legal cases, where the IPC sections are mapped to their BNS equivalents. By extracting critical case details and analyzing them in the context of the new legal framework, this approach helps legal professionals navigate and interpret cases more effectively under BNS.
- **Structured Mapping from IPC to BNS:** The structured mapping of IPC to BNS provides an approach to ensure that case details and legal interpretations remain accurate during the transition. This mapping serves as a foundational tool for professionals using BNS to understand historical cases and their updated classifications.

1.4 Identification of challenges in the selected topic

The shift from the Indian Penal Code (IPC) to the Bharatiya Nyaya Sanhita (BNS) framework presents multiple challenges, including the lack of datasets compatible with BNS, which hampers legal professionals reliant on IPC-based analysis tools. Existing systems lack systematic mapping strategies for translating IPC sections into BNS equivalents, creating difficulties in interpreting historical legal data under the new framework. Additionally, legal document summarization tools designed for IPC cases fail to align with the simplified classifications introduced by BNS, while predictive models lack configurations for case outcomes under BNS, limiting their applicability. To address these issues, tailored tools, datasets, and methodologies specifically aligned with BNS standards are imperative.

1.5 Problem Statement and Proposed Solution

The transition from the IPC to the BNS framework presents significant challenges, necessitating the development of effective legal document analysis tools. One major hurdle is the absence of a dedicated BNS dataset, which complicates accurate legal interpretation and analysis. Existing tools also struggle to adapt to the new BNS framework, impacting the efficiency of document summarization and case analysis.

To address these issues, our project aims to bridge the gap by mapping IPC sections to their BNS equivalents, thereby enabling automated legal analysis tailored to the BNS framework. In the absence of a dedicated BNS dataset, we utilized existing IPC legal cases and historical records for fine-tuning BART, an advanced language model for abstractive summarization.

BART was fine-tuned for legal document summarization, enabling the generation of concise and meaningful summaries of complex legal texts. Additionally, InLegalBERT was incorporated into the system as part of the extractive

summarization pipeline, designed to accurately extract key information from legal documents. The combination of BART for abstractive summarization and InLegalBERT for extractive summarization ensures that the system efficiently handles both detailed information extraction and concise summary generation, tailored to the BNS framework.

This approach, supported by structured IPC-to-BNS mapping, enhances the overall efficiency and accuracy of legal document processing and case analysis, helping legal professionals navigate the transition to the BNS system.

1.6 Scope of the system

The system focuses on enhancing legal document analysis in the context of the transition from the Indian Penal Code (IPC) to the Bharatiya Nyaya Sanhita (BNS). It provides automated tools to assist legal professionals by enabling the extraction and summarization of legal documents and mapping IPC sections to their corresponding BNS sections. The system fine-tunes a fine-tuned BART model for abstractive summarization of legal texts, while also utilizing IPC-to-BNS mapping to align extracted case details with the new legal framework. By doing so, it improves the efficiency, consistency, and relevance of legal document handling. The scope is limited to document summarization, IPC-BNS section mapping, and preliminary legal analysis support, ensuring the system remains practical and targeted for current legal needs.

Chapter 2

Review of Literature

2.1 Survey of Existing Systems

The field of text summarization has witnessed significant advancements, particularly with the emergence of pre-trained language models tailored for specific domains, including the legal sector. Paul et al. [1] re-train two popular legal PLMs, LegalBERT and CaseLawBERT, on Indian legal data. InLegalBERT improves significantly over LegalBERT, while the gains are much smaller for InCaseLawBERT over CaseLawBERT for Indian dataset. They also highlight the effectiveness of pre-trained language models in enhancing legal document summarization, showcasing their ability to understand legal terminologies and context.

In comparative analyses of extractive summarization techniques, Rani and Bidhan [2] examined traditional methods such as TextRank, TF-IDF, and LDA. Their findings revealed the strengths and weaknesses of each method, emphasizing that while TF-IDF is efficient for term extraction, TextRank provides a more nuanced approach to capturing contextual relationships within text. This study serves as a foundation for understanding the performance of different summarization algorithms and their applicability to various domains, including legal texts.

Recent advancements in latent semantic analysis also contribute to the summa-

rization landscape, as explored by Onah et al. [5], who apply LDA topic modeling to automatic text summarization. This approach enhances the understanding of contextual relationships within the text, further refining the summarization process. Additionally, Ramadhan et al. [6] investigate the implementation of the TextRank algorithm in summarizing product reviews, which demonstrates the versatility of this method across different types of content.

Jewani et al. [7] provide a brief overview of various extractive summarization methods, offering insights into emerging trends and methodologies that enhance summarization capabilities. Gupta [8] and Jain [9] discuss the application of TextRank and TF-IDF, respectively, in their practical implementations, contributing to a broader understanding of how these algorithms can be employed for effective text summarization. Collectively, these studies illustrate the evolving landscape of text summarization techniques, highlighting the necessity of combining traditional methods with advanced models to address the specific challenges of summarizing legal documents effectively.

2.2 Limitations of Existing Systems

While there is extensive research and many tools available for IPC-based legal analysis, few studies address the requirements of the BNS Act, leading to notable limitations in existing systems:

Lack of BNS-Compatible Datasets: Existing tools and databases are designed exclusively for IPC, lacking the structural adjustments needed to accommodate BNS categories. Without a dedicated BNS dataset, it is challenging to perform reliable analysis under the new legal framework.

Inadequate Mapping of IPC to BNS: Existing tools lack systematic mapping strategies for translating IPC sections into their BNS equivalents, making it difficult to interpret or apply historical data within the new BNS structure.

Limitations in Legal Document Summarization: Current summarization models are often designed for IPC-based cases, which may not align with the simplified classifications of BNS. Without tailored summarization techniques, these tools are inadequate for creating relevant summaries under the BNS Act.

Lack of Comprehensive Case Analysis for BNS Cases: Existing tools are not equipped to perform detailed case analysis specific to the BNS structure. Legal professionals face challenges in interpreting and analyzing case details under the BNS framework without dedicated analysis tools designed for these cases.

These gaps emphasize the need for a comprehensive system that provides structured BNS analysis, enabling accurate mapping, document summarization, and predictive modeling specifically for the new framework.

2.3 Motivation

The gaps and limitations in current legal systems underscore the need for a BNS-compatible solution. Key motivators include the urgent requirement for tools that support BNS standards, such as document summarization, section mapping, and outcome prediction, to aid legal professionals transitioning from IPC. The project also seeks to improve the accuracy of BNS analysis by creating a synthetic dataset and a rule-based mapping strategy to enhance precision in legal document interpretation. By providing tools for efficient document summarization and classification under BNS, this project boosts processing efficiency and supports professionals through the learning curve of adapting to BNS. Furthermore, as the BNS Act gains traction, this project establishes a foundation for future advancements in legal technology, meeting immediate professional needs while setting the stage for innovative BNS-aligned tools.

Chapter 3

Proposed System: Analysis

3.1 Detailed explanation of Proposed system

3.1.1 Working Principle

Summarization Pipeline Algorithm

Step 1: Preprocessing and Sentence Extraction

- Extract raw text from PDF (if needed) using `fitz` (PyMuPDF).
- Preprocess the text:
 - Remove unwanted headers/footers (like "Indian Kanoon").
 - Convert text to lowercase.
 - Remove special characters (only keep alphanumerics and punctuation).
 - Collapse multiple spaces.
- Tokenize the preprocessed text into individual sentences using `nltk.sent_tokenize`.

Step 2: Sentence Embedding with InLegalBERT

- Use the pretrained InLegalBERT model to obtain dense embeddings:
 - Tokenize the cleaned sentences using `bert_tokenizer`.

- Pass the tokenized sentences through `bert_model`.
- Extract the output: take the mean of token embeddings (`last_hidden_state.mean(dim=1)`) to get a single 768-dimensional embedding for each sentence.

Step 3: Sentence Scoring and Selection (Extractive Summarization)

- Compute a cosine similarity matrix between sentence embeddings.
- Use similarity scores to find the most representative sentences.
- Select top sentences based on similarity or clustering.

Step 4: Feeding Extracted Sentences to Fine-tuned BART

- Concatenate the top-ranked sentences into a single input text.
- Tokenize this extractive summary using `bart_tokenizer`.
- Pass the tokenized text into the fine-tuned BART model (`fine_tuned_bart_legal`) to perform abstractive summarization.
- Generate the final summary output.

The proposed system for criminal case summarization and outcome prediction is based on a hybrid summarization and analysis pipeline that leverages both extractive and abstractive summarization techniques, along with legal-domain-specific embeddings and classification models. The major algorithms and methodology steps are explained below.

3.1.2 Phase/ Module - wise explanation

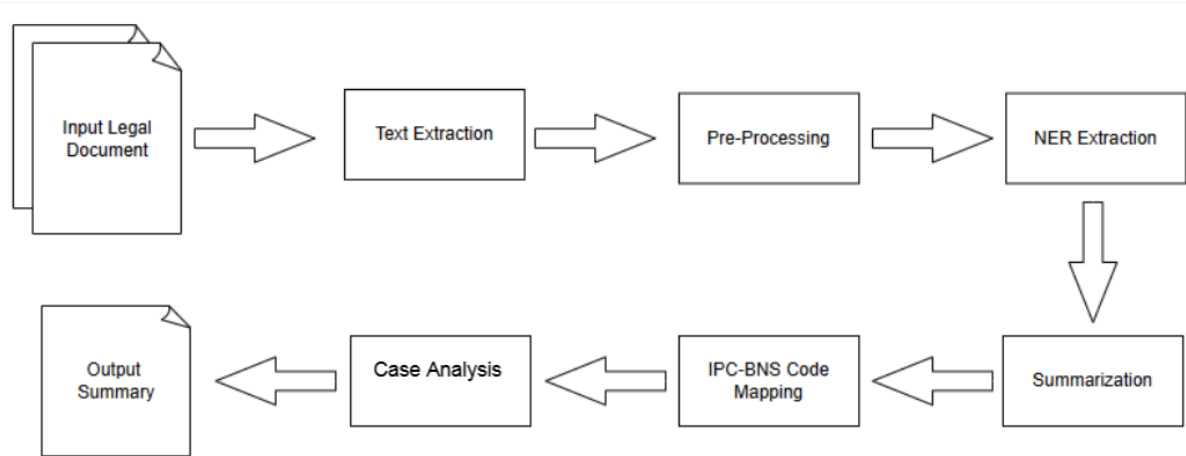


Figure 3.1: Work Flow Diagram For System

1. **Input Document:** The process begins with the input of criminal case documents into the system. These documents can include various legal texts such as case files, court rulings, and legal briefs. The input stage is crucial as it sets the foundation for the subsequent analysis and processing steps.
2. **Text Extraction:** Once the documents are inputted, the next step is text extraction. This involves extracting the relevant text from the documents, which may include removing any non-textual elements like images or tables. The goal is to obtain clean, raw text data that can be further processed.
3. **Preprocessing:** After text extraction, the preprocessing stage involves cleaning and organizing the extracted text. This can include tasks such as removing stop words, normalizing text, and handling any inconsistencies. Preprocessing ensures that the text data is in a suitable format for analysis, improving the accuracy and efficiency of subsequent steps.
4. **IPC-BNS Code Mapping:** In this step, the preprocessed text is mapped to relevant IPC (Indian Penal Code) and BNS (Bharatiya Nyaya Sanhita) codes. This mapping helps in categorizing the text according to legal

codes, which is essential for understanding the legal context and for further analysis.

5. **Summarization:** The summarization process condenses the preprocessed text into a concise summary. This step uses the InLegal BERT model to identify and extract the most important information from the text, making it easier to understand the key points of the document without reading the entire text.
6. **NER Extraction:** Named Entity Recognition (NER) extraction runs parallel to summarization. This process identifies and extracts specific entities such as names, dates, locations, and legal terms from the text. NER is important for understanding the context and specifics of the case.
7. **Case Analysis:** Using the summarized content and extracted legal entities, the system performs a detailed case analysis, highlighting important aspects like the case background, referenced legal provisions, evidence review, and case outcome.
8. **Final Output:** The final output is a comprehensive summary of the criminal case along with the case analysis. This output can be used by legal professionals to quickly understand the case details and its outcomes, aiding in decision-making and legal research.

3.2 System Analysis

3.2.1 Functional Requirements

- **Legal Document Summarization:** The system must generate concise and coherent summaries of lengthy legal documents using extractive and abstractive summarization techniques.
- **Case Classification:** The system should accurately classify legal cases by fine-tuning InLegalBERT with IPC legal cases and historical records.

- **IPC to BNS Mapping:** Utilize a structured strategy to map IPC sections to their BNS equivalents, ensuring accurate legal analysis.
- **Automated Legal Analysis:** Provide tools for automated legal analysis tailored to the BNS framework, enhancing efficiency and accuracy.

3.2.2 Non- Functional Requirements

- **Performance:** The system should process and analyze legal documents efficiently, providing quick and accurate results.
- **Scalability:** The system must be scalable to handle a large volume of legal documents and cases.
- **Reliability:** Ensure the system is reliable and provides consistent results across different legal documents and cases.
- **Usability:** The system should be user-friendly, with an intuitive interface for legal professionals.
- **Security:** Protect sensitive legal data and ensure that all data processing complies with relevant legal and ethical standards.
- **Maintainability:** The system should be easy to maintain and update, allowing for continuous improvements and adaptations to new legal frameworks.

3.2.3 Software and Hardware requirements

Hardware Requirements:-

- Windows 10 with x-64 based processor.
- Intel i3-9th generation
- Ryzen 3200h x64 based processor.
- 8 GB RAM

Software Requirements:-

- Pycharm
- OS-Windows 10
- Github
- Google Collab

3.2.4 Use Case Modeling



Figure 3.2: Use Case Diagram For System

As shown in Figure 3.2, the primary actor in the system is the User, who uploads a legal document and interacts with the system for document summarization and case analysis. The system extracts text from the uploaded document, summarizes it using InLegalBERT and BART and displays the summary to the user. Additionally, the system identifies IPC sections in the document and provides its equivalent BNS sections along with its details. The user can download the summary and also view the generated summary history. Finally, the case analysis is displayed to the user for review, supporting their legal decision-making.

Use case Template 1:

Table 3.1: Use case template 1 of Summarization System

ID	UC-001
Title	Generate Summary of Input Legal Document
Description	This use case allows a user to input a legal document to get a desired summary of the legal document.
Primary Actor	User
Secondary Actor	Summarization generator system
Preconditions	<ul style="list-style-type: none"> - The user has access to the summary generation system. - The system is operationable and capable of generating precise summaries. - Presence of a proper Legal document.
Postconditions	<ul style="list-style-type: none"> - The summary is successfully generated which can be used by the user to gain <u>meaning</u> insights of the document.
Trigger	The user uploads the legal document.
Main Success Scenario	<ul style="list-style-type: none"> - User provides a valid Legal document. - The system processes the legal document and extracts important and meaningful insights. - The summary is generated and displayed.
Status	Completed
Owner	Samit Fernandes
Priority	High

As shown in Table 3.1, this use case enables a user to upload a legal document and receive a precise summary generated by the system. The user interacts with a summarization generator, assuming the system is operational and the document is properly formatted. Upon uploading, the system processes the document, extracts key insights, and provides a summarized output. The use case has been completed and is considered high priority, owned by Samit Fernandes.

Use case Template 2:

Table 3.2: Use case template 2 of Summarization System

ID	UC-002
Title	Analyze Outcome of the Legal Document
Description	This use case allows a user to input a legal document to obtain analysis of the Legal document
Primary Actor	User
Secondary Actor	Case Analyzer system
Preconditions	<ul style="list-style-type: none"> - The user has access to the Case Analyzer system. - The system is operationable and capable of analyzing Outcome of the document. - Presence of a proper Legal document.
Postconditions	<ul style="list-style-type: none"> - The Outcome of the document is highlighted which will give the sections applicable to the convicted.
Trigger	The user uploads the legal document.
Main Success Scenario	<ul style="list-style-type: none"> - User provides a valid Legal document. - The system processes the legal document and lists all the sections applicable if possible. - The Case Outcome displayed.
Status	Completed
Owner	Jaden Franco
Priority	High

As shown in Table 3.2, this use case allows a user to upload a legal document and receive an outcome analysis through the Case Analyzer system. With a valid document and operational system, it identifies and highlights applicable sections relevant to the convicted. After processing, the system presents the case outcome. The use case is marked as completed, with high priority, and is owned by Jaden Franco.

3.3 Proposed System :Analysis, Modelling and Design

3.3.1 Class Diagram

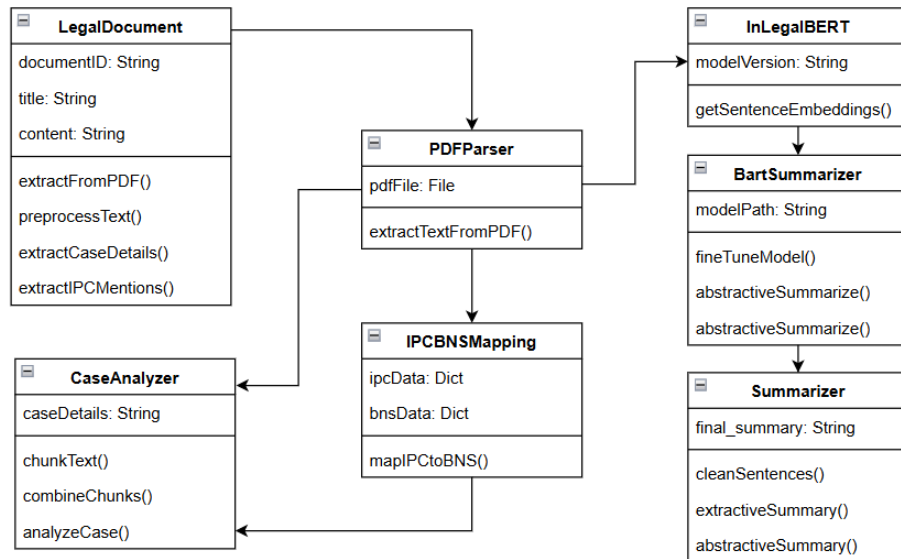


Figure 3.3: Class Diagram of System

Figure 3.3 shows that the `LegalDocument` class handles legal documents, including text extraction from PDFs, preprocessing, and extracting case details and IPC mentions. It interacts with the `PDFParser` to extract text, and `InLegalBERT` for generating sentence embeddings. The `BartSummarizer` fine-tunes the BART model and produces abstractive summaries from text, while the `Summarizer` generates both extractive and abstractive summaries. `IPCBNSMapping` maps IPC sections to BNS sections for accurate legal classification. `CaseAnalyzer` processes text in chunks and provides Case details. These classes work together to process, summarize, and analyze legal documents efficiently.

3.3.2 Sequence Diagram

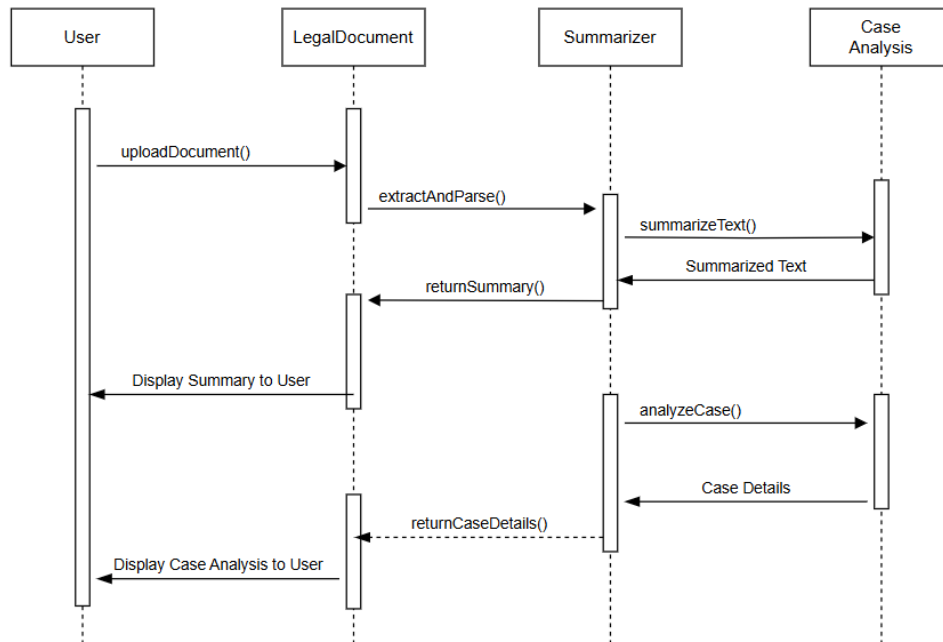


Figure 3.4: Sequence Diagram for Summarization & Case Analysis

Figure 3.4 illustrates the sequence diagram interaction flow between the User, LegalDocument, and CaseAnalyser within a system designed for legal document summarization and case details analysis. Initially, the User uploads a legal document, prompting the LegalDocument to extract and summarize the text using InLegalBERT and BART. Once the document is processed, the User can view the summary. Simultaneously, the LegalDocument triggers the CaseAnalyser to gather a relevant dataset, which is then used to analyze the legal case details. The system evaluates then displays the predicted results to the User. The diagram emphasizes the active role of the User throughout the process, interacting with the system at each step.

3.3.3 DFD

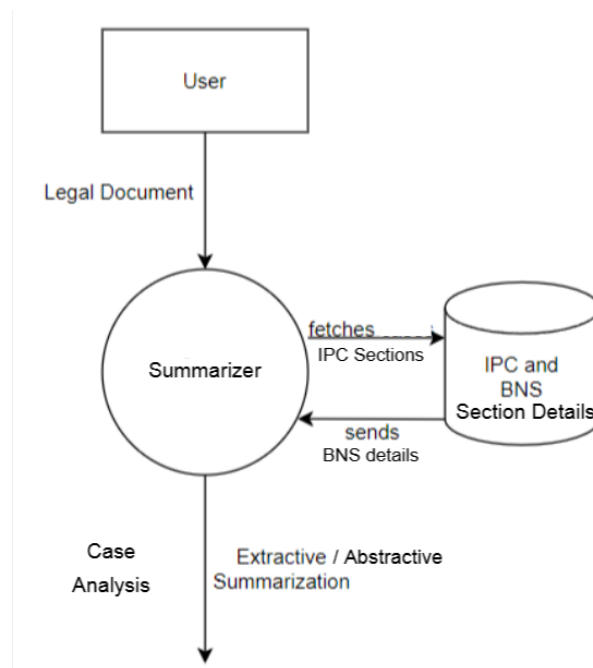


Figure 3.5: Data Flow Diagram Level 0

Figure 3.5 illustrates a high-level overview. The user submits a legal document to the InLegalBERT system, which serves as a summarizer and case analyzer. The system communicates with a database of IPC and BNS Sections, fetching relevant BNS equivalent sections for case analysis. InLegalBERT processes the document to produce an extractive summary, while fine-tuned BART provides an abstractive summary, which is returned to the user.

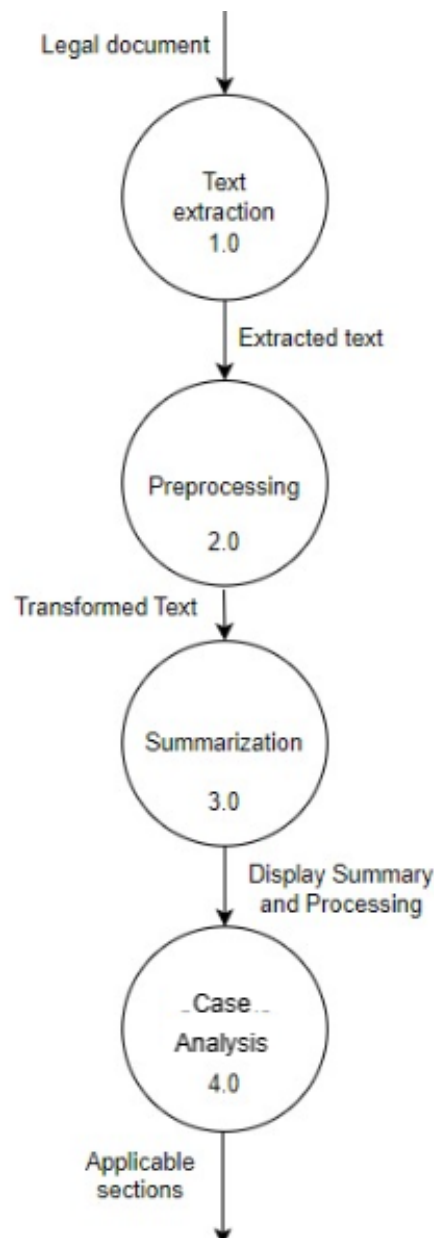


Figure 3.6: Data Flow Diagram Level 1

Figure 3.6 breaks down the InLegalBERT processing steps. First, the system extracts text from the legal document, then preprocesses it to standardize the format. Following preprocessing, the system generates a summary of the document and performs case analysis, which includes identifying applicable legal sections and case outcome. The summarized content and analysis results are displayed to the user.

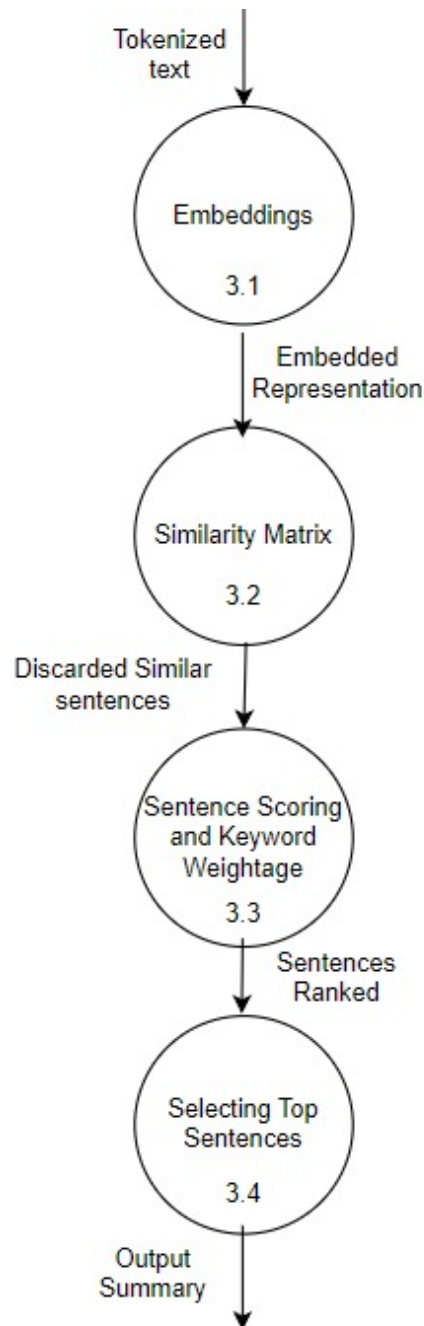


Figure 3.7: Data Flow Diagram Level 2

Figure 3.7 provides further detail on the summarization step. It starts with tokenizing the text and generating embeddings to represent each sentence. A similarity matrix is then created to compare sentences, allowing the system to discard similar sentences. Sentence scoring and keyword weighting are applied to rank sentences, and the top-ranked sentences are selected to form the final output summary.

3.3.4 Architectural View

Extractive Summarization:

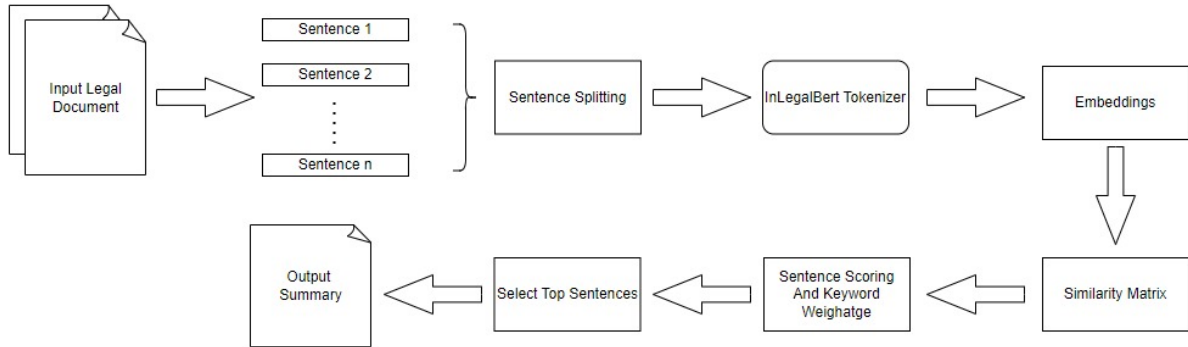


Figure 3.8: BERT-Based Extractive Summarization Architecture

Figure 3.8 illustrates the step-by-step process involved in generating an extractive summary of a legal document using a BERT-based model, specifically tailored for legal text, such as InLegal-BERT.

- **Input Text:** The process begins with the input of a legal document into the system. This document serves as the foundation for the subsequent steps in the summarization process.
- **Sentence Splitting:** The legal document is split into individual sentences. This step ensures that the text is broken down into manageable components for further analysis.
- **Tokenization:** Each sentence is tokenized using the InLegal-BERT tokenizer. Tokenization converts sentences into smaller units, making it easier for the BERT model to process them.
- **Generate Embeddings:** The InLegal-BERT model generates embeddings for each sentence. These embeddings represent the semantic meaning of the sentences in vector form.
- **Similarity Matrix Calculation:** The inner product of the sentence embeddings is computed to form a similarity matrix. This matrix captures the relationships between sentences based on their content.

- **Sentence Scoring:** Sentences are scored based on their position in the similarity matrix and their length. The scoring process ranks sentences according to their importance and relevance.
- **Keyword Weighting:** Legal terms within each sentence are weighted. This ensures that key legal concepts have a greater influence on the final summary.
- **Sentence Selection:** The top sentences are selected based on their combined score. This step identifies the most important sentences that will form the final summary.
- **Output Summary:** A fixed-length summary is generated by selecting the most relevant sentences. The summary presents the core information from the document in a concise manner.

Figure 3.8 effectively demonstrates how BERT-based extractive summarization efficiently processes legal documents by tokenizing text, calculating sentence similarities, weighting keywords, and selecting the most important sentences to generate a concise summary that captures the core legal information.

Abstractive Summarization:

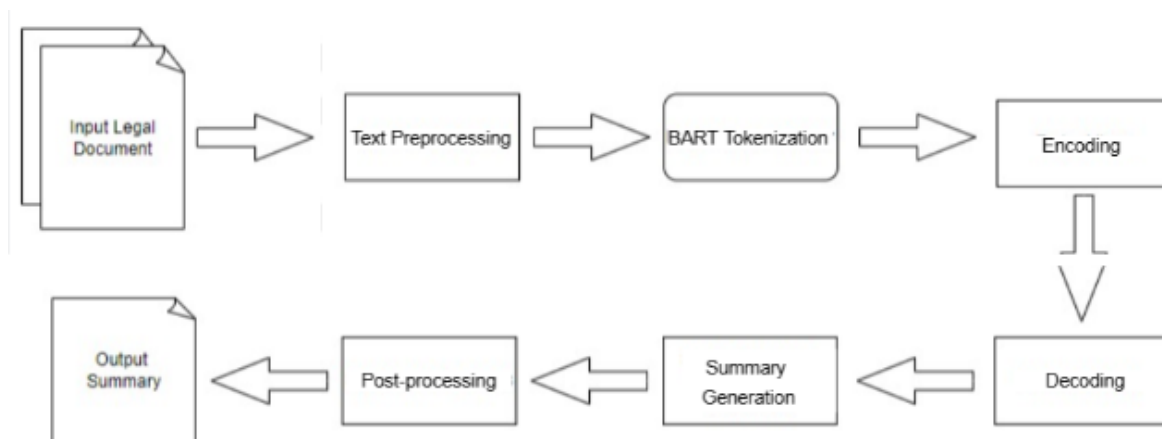


Figure 3.9: BART-Based Extractive Summarization Architecture

Figure 3.9 illustrates the step-by-step process involved in generating an abstractive summary of a legal document using a BART-based model, specifically

facebook/bart-large, fine-tuned for legal texts.

- **Input Text:** The process begins with the input of a legal document into the system. This document serves as the foundation for the subsequent summarization process.
- **Text Preprocessing:** The input text is preprocessed by cleaning, normalizing, and formatting it into structured paragraphs suitable for tokenization. This step ensures the text is in an optimal state for encoding.
- **Tokenization:** The preprocessed text is tokenized using the BART tokenizer. Tokenization converts the input into smaller subword units, enabling the model to handle complex legal terminology efficiently.
- **Encoding:** The tokenized input is passed through the BART encoder, which generates contextual embeddings that capture the meaning, sequence, and dependencies within the legal text.
- **Decoding:** The BART decoder processes the encoder's embeddings to generate a new sequence of tokens. Unlike extractive models, the decoder constructs entirely new sentences, ensuring paraphrasing and abstraction.
- **Summary Generation:** The output tokens are compiled to form a coherent and fluent summary. This summary captures the essential legal information, key arguments, and rulings from the original document.
- **Post-processing:** Minor corrections such as punctuation adjustment, sentence boundary refinement, and de-tokenization are applied to ensure the summary is grammatically correct and readable.
- **Output Summary:** The final abstractive summary presents the case's core elements in a condensed and rephrased manner, offering users a quick yet comprehensive understanding without replicating the original text.

Figure 3.9 showcases how BART-based abstractive summarization effectively processes legal documents by encoding and re-generating information, resulting

in a more natural, human-like summary that not only shortens but also rephrases the content to improve clarity and conciseness.

3.3.5 Algorithms / Methodology

(A) Preprocessing

The input legal documents are first preprocessed using text cleaning techniques:

- Removing unwanted characters, headers, footers, and special symbols.
- Splitting into sentences using a sentence tokenizer.
- Normalizing spacing and punctuation.

This ensures the document is clean, structured, and ready for further processing.

(B) Extractive Summarization using InLegalBERT

InLegalBERT is a variant of the BERT language model, fine-tuned specifically on Indian legal corpora including judgments, statutes, and case laws. It is designed to better understand legal terminologies, hierarchical case structures, and statutory references that a generic BERT model may miss.

InLegalBERT generates dense embeddings (vector representations) of sentences which capture both the semantic meaning and legal context. These embeddings are used to determine sentence importance.

Methodology

1. **Sentence Embedding:** Each sentence is passed through InLegalBERT to obtain a 768-dimensional semantic vector.
2. **Similarity Matrix Creation:** A cosine similarity matrix is constructed where each entry represents the similarity between two sentences.
3. **Sentence Scoring:** Sentences are scored based on their centrality, calculated as the sum of their similarity with other sentences.

4. **Legal Enhancement:** Sentences containing important legal phrases (like “under Section 302”, “prosecution”, “evidence”) are optionally given additional weighting.
5. **Top Sentence Selection:** Based on the scores, top-ranked sentences are selected to form a factual and legally representative extractive summary.

This ensures high factual accuracy while maintaining legal integrity.

(C) Abstractive Summarization using fine-tuned BART

BART (Bidirectional and Auto-Regressive Transformers) is a sequence-to-sequence model that combines the benefits of BERT (understanding) and GPT (generation). It is pre-trained using denoising autoencoding, meaning it learns to reconstruct original texts from corrupted inputs, making it excellent for summarization tasks.

BART encodes the entire input into a condensed format and then decodes it to generate a shorter, fluent version of the text.

Methodology

1. **Full-Text Encoding:** The input document (or its chunked parts) is passed through BART’s encoder, generating compressed latent representations.
2. **Summary Generation (Decoding):** The decoder generates a new sequence of sentences that captures the essence of the document, using beam search to optimize quality.
3. **Control Parameters:** Maximum and minimum summary lengths are controlled. No-repeat n-gram constraints and length penalties are applied to improve summary quality.
4. **Post-processing:** The final summary is cleaned for grammatical consistency and completeness.

This enables the system to generate fluent, interpretable, and reader-friendly

summaries even for non-experts.

(D) Hybrid Summarization

In the hybrid approach, the document is first divided into smaller chunks. Each chunk is individually summarized using BART. Finally, all partial summaries are concatenated and refined into a cohesive final summary.

This method balances:

- **Factuality** (by limiting the hallucination of long generation)
- **Readability** (by ensuring each chunk is properly summarized)

(E) IPC to BNS Section Mapping

Following the legislative shift from IPC (Indian Penal Code) to BNS (Bharatiya Nyaya Sanhita), it is crucial to accurately map old IPC sections cited in cases to their corresponding BNS sections for modern relevance.

Methodology

1. **Section Identification:** The case text is scanned for mentions of IPC sections using regex-based pattern matching and legal Named Entity Recognition (NER).
2. **Section Mapping:** Identified IPC sections are mapped to their corresponding BNS sections using a precompiled IPC-BNS mapping database.
3. **Mapping Insertion:** Where an IPC section is found, the corresponding BNS section reference is added in parentheses for updated interpretation.
4. **Validation:** The mappings are verified to ensure contextual relevance and legal consistency.

(F) Case Analysis Module

The system includes a Case Analysis component that performs deep analysis of legal documents by extracting key case elements. It is designed to interpret legal arguments and structure them into meaningful sections.

Methodology

1. **Text Chunking:** Large legal documents are segmented into smaller chunks based on a predefined word count.
2. **Batch Processing:** Multiple chunks are grouped together for efficient parallel analysis.
3. **Structured Analysis:** Each chunk is analyzed to extract and organize the following information:
 - Background of the case
 - Evidence presented
 - Judicial judgment
 - Precedent cases referred
 - Important legal sections invoked
 - Case outcome summarized in one sentence
4. **Result Merging:** Analyses from all chunks are combined to produce a complete structured case analysis.

The Case Analysis module enhances interpretability by presenting complex legal cases in an accessible and organized format.

3.3.6 UI/UX design

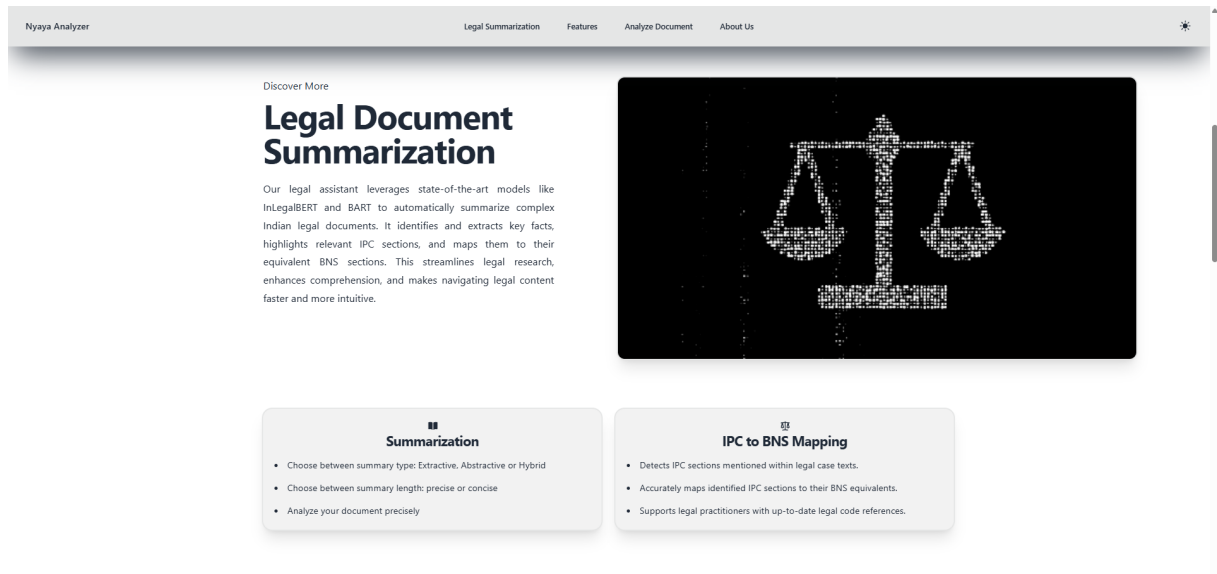


Figure 3.10: Homepage 1 of Summarization System

Figure 3.10 provides a brief introduction to the website, outlining its core functionalities and services. It serves as an entry point for users to understand the purpose of the platform, which includes legal document summarization and analysis. The homepage features a user-friendly interface with navigation options to access the summary generation, and analysis tools, offering a seamless experience to legal professionals and users.

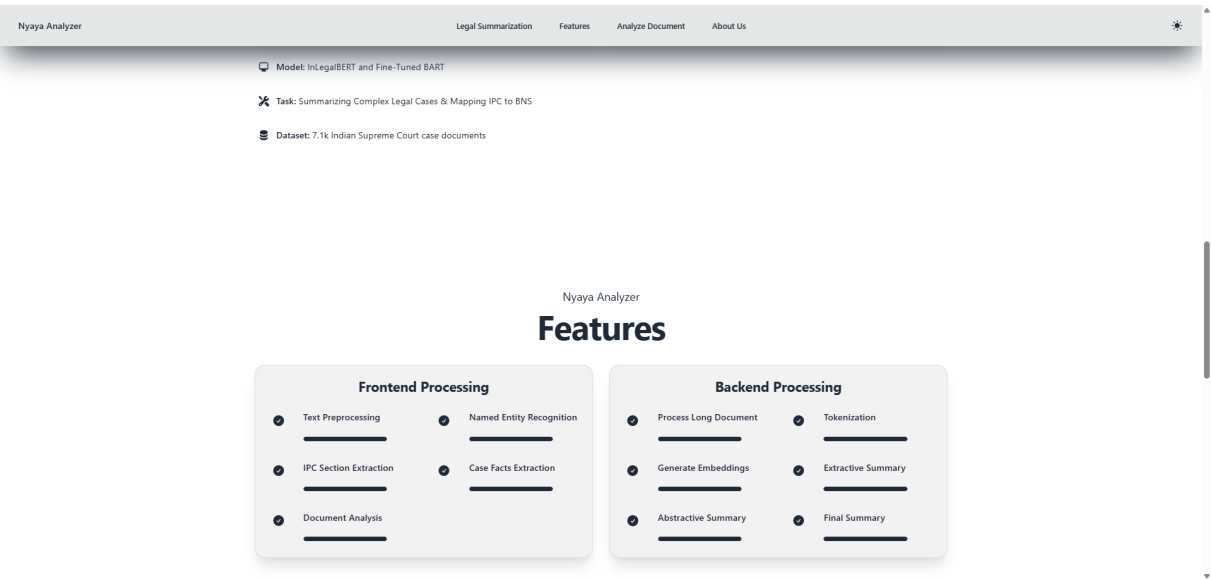


Figure 3.11: Features page of Summarization System

Figure 3.11 highlights the key frontend and backend functionalities of the system. On the frontend, it includes text preprocessing, IPC section extraction, Named Entity Recognition (NER), and overall document analysis to identify key legal elements. The backend processes long documents, performs tokenization, generates embeddings, and creates both extractive and abstractive summaries. These features work together to provide an efficient and accurate legal document analysis system.

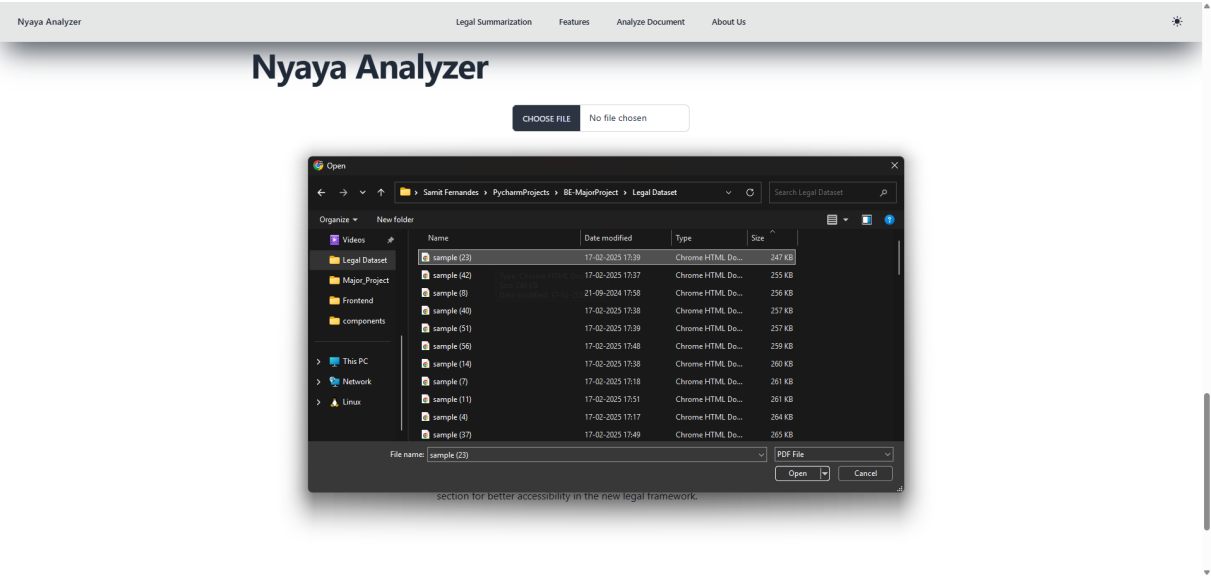


Figure 3.12: Upload the legal document that you want to summarize and analyze

Figure 3.12 shows that users can upload the legal document they wish to summarize and analyze by clicking the "Choose File" button and selecting the desired file from their device.

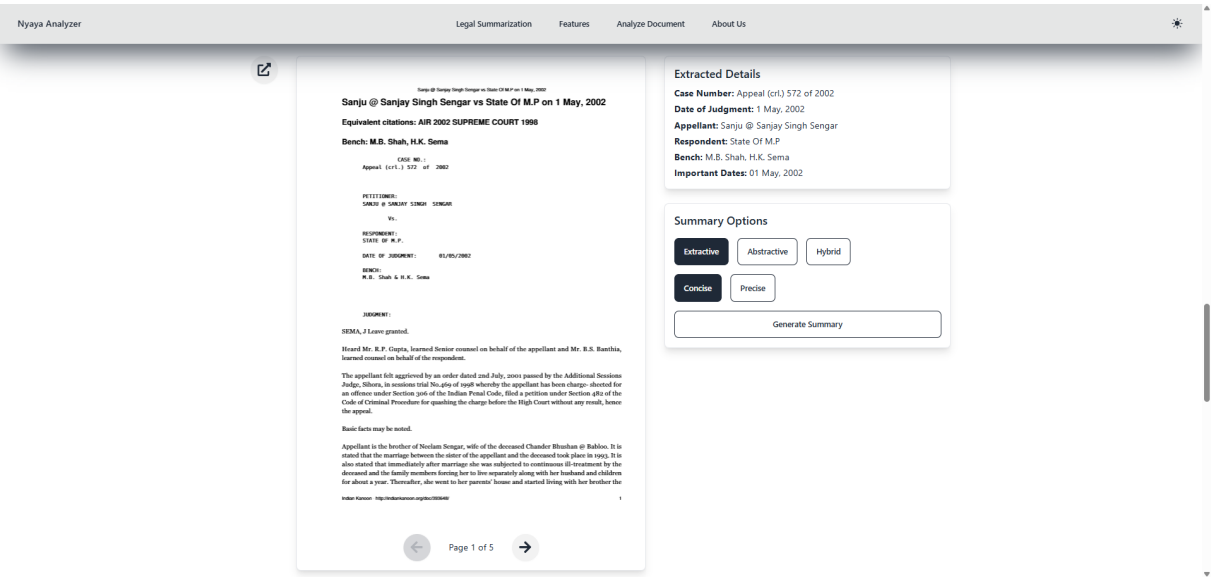


Figure 3.13: Preview the uploaded document and select the summary options desired

Figure 3.13 shows the system allowing users to preview the uploaded legal document's content, and choose summary options such as the desired summary length (precise or concise) and summary type (extractive, abstractive, or hybrid) before generating the output.

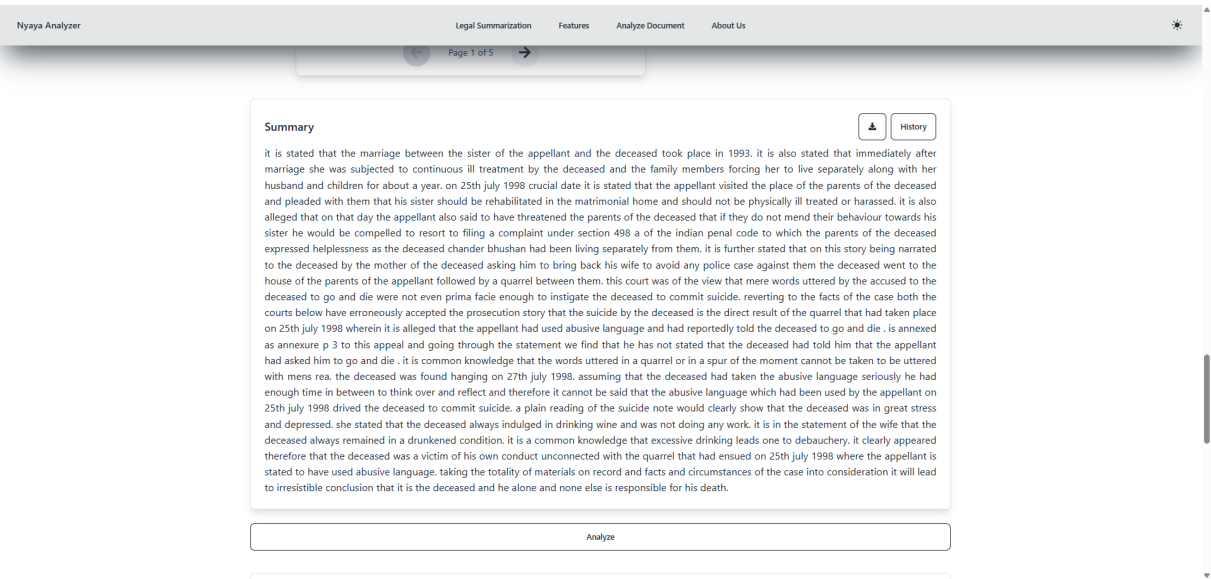


Figure 3.14: Displays the desired summary

After selecting the desired summary options (length and type), users can click the "Generate Summary" button. The system will then process the document and display the generated summary according to the chosen parameters as seen in Figure 3.14.

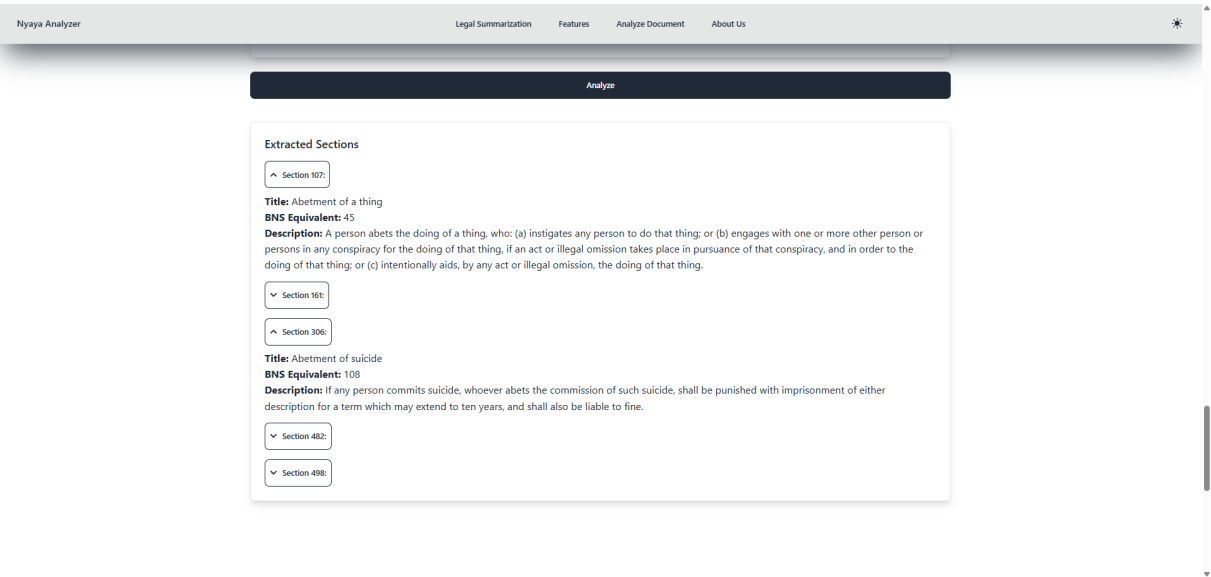


Figure 3.15: IPC Sections are extracted and its equivalent BNS details are displayed

The system extracts the IPC sections from the uploaded document and maps them to their corresponding BNS sections. For each extracted IPC section, the equivalent BNS details are displayed, including the BNS section number, title, and a brief description as shown in Figure 3.15.

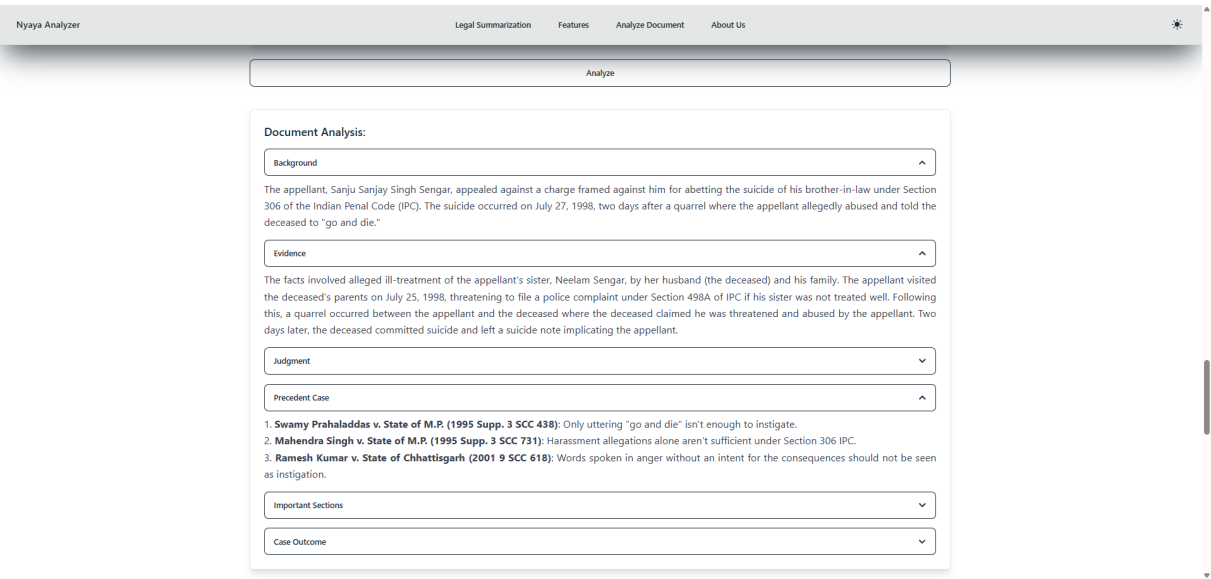


Figure 3.16: Analysis of the legal document is displayed to the user

On clicking the "Analyze" button, the system processes the uploaded legal document and displays a detailed analysis to the user. As seen in Figure 3.16, this includes key case details such as the case background, evidence, judgment, precedent cases, important sections, and case outcome.

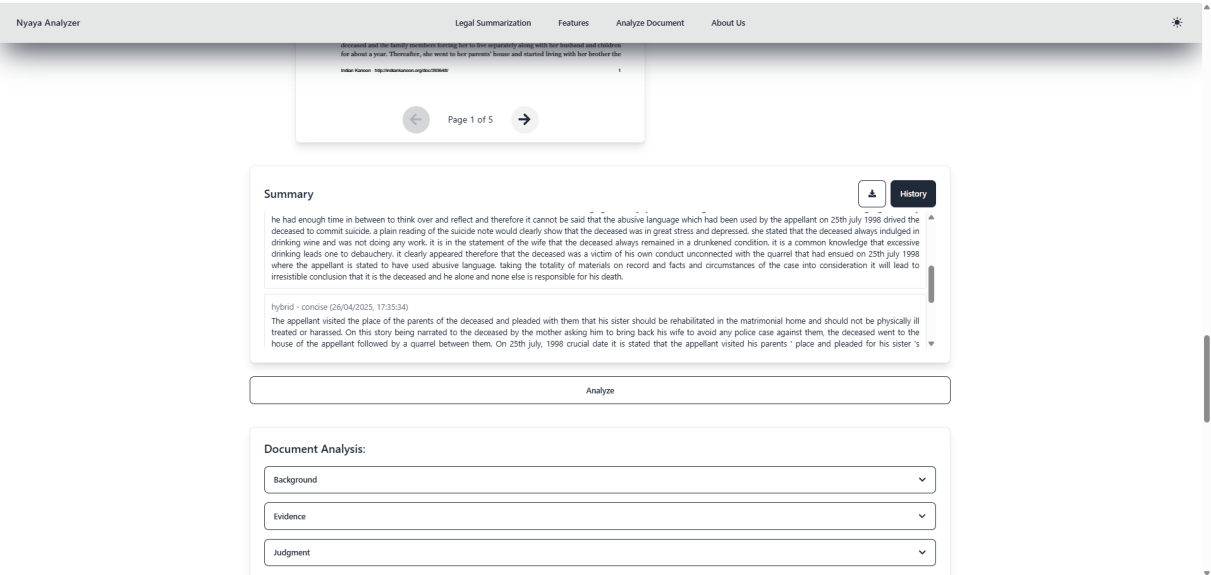


Figure 3.17: All the generated summary history is stored for quick access

All previously generated summaries are stored for quick access. By clicking the "History" button located at the top-right corner of the generated summary, users can view a history of past summaries, enabling them to easily revisit and refer to previous summaries as seen in Figure 3.17.

Chapter 4

Implementation Plan and Experimental Set up of the Proposed system

4.1 Experimental Set up

4.1.1 Details discussion of input/Dataset

The system handles input data as:

- **Uploaded Legal Documents:** Users upload their case files, which are processed for summarization and case analysis.

In addition to this, the IPC-BNS Mapping class plays a crucial role in mapping sections of the Indian Penal Code (IPC) to their corresponding sections in the BNS framework, providing critical context for document analysis.

Dataset Used for Model Training:

To fine-tune the summarization model, a publicly available dataset containing 7,100 Indian Supreme Court case documents was used. Key details about the dataset [20]:

- **Data Source:** Indian Supreme Court judgments.
- **Dataset Size:** 7,100 documents for training, 100 documents for testing.

- **Data Fields:**

- `id`: IndianKanoon Case ID (string)
- `num_doc_tokens`: Number of tokens in the full document (integer)
- `num_summ_tokens`: Number of tokens in the summary (integer)
- `document`: List of sentences forming the full document (List of strings)
- `summary`: List of sentences forming the abstractive summary (List of strings)

Each document consists of multiple sentences, and the corresponding summary captures the essence of the case, similar to legal headnotes.

Models Used: Summarization Module (SUMM):

- The summarization system (SUMM) automatically generates a coherent abstractive summary capturing the critical aspects of the uploaded case document.
- The pre-trained facebook/bart-large model was fine-tuned on the Indian Supreme Court dataset for this purpose.
- The generated summaries help users quickly understand the case without reading the full document.

4.1.2 Performance Evaluation Parameters

To evaluate the performance of the system, we focus on key parameters for summarization. These parameters assess the efficiency, quality, and relevance of the summaries produced by the system's models.

The following evaluation metrics were used to assess the performance of the summarization model:

- **ROUGE-1:** Measures the overlap of unigrams (single words) between the generated summary and the reference summary.

- **ROUGE-2:** Measures the overlap of bigrams (pairs of words) between the generated summary and the reference summary.
- **ROUGE-L:** Measures the longest common subsequence between the generated summary and the reference summary, accounting for sentence structure and ordering.
- **BLEU:** Evaluates the n-gram precision between the generated summary and the reference summary, emphasizing exact word matches.
- **METEOR:** Assesses synonym matching, stemming, and word reordering to evaluate the fluency and adequacy of the generated summary.
- **BERTScore Precision:** Measures how precisely the tokens in the generated summary match the meaning of tokens in the reference summary using contextual embeddings.
- **BERTScore Recall:** Measures how much of the meaning from the reference summary is captured in the generated summary using contextual embeddings.
- **BERTScore F1-Score:** The harmonic mean of BERTScore Precision and Recall, providing a balanced measure of semantic similarity.

4.2 Code

```
import torch
from transformers import AutoTokenizer, AutoModel,
BartForConditionalGeneration
from sklearn.metrics.pairwise import cosine_similarity
import fitz # PyMuPDF
import nltk
import re
import numpy as np
```

```
nltk.download("punkt")

device = torch.device("cuda" if torch.cuda.is_available()
else "cpu")

# Load models
bert_tokenizer = AutoTokenizer.from_pretrained("law-ai
/InLegalBERT")
bert_model = AutoModel.from_pretrained("law-ai/InLegalBERT")
.to(device)

bart_tokenizer = AutoTokenizer.from_pretrained("path/to
/fine_tuned_bart_legal")
bart_model = BartForConditionalGeneration.from_pretrained(
"path/to/fine_tuned_bart_legal").to(device)

# Step 1: Preprocessing and Sentence Extraction

def extract_text_from_pdf(pdf_path):
    doc = fitz.open(pdf_path)
    return "\n".join(page.get_text("text") for page in doc)
.strip()

def preprocess_text(text):
    text = re.sub(r"(?m)^\Indian Kanoon\s*-\s*http\S+\n\d+\s*",
    "", text, flags=re.IGNORECASE)
    text = text.lower()
    text = re.sub(r"[\w\s\.\?\!]", " ", text)
    text = re.sub(r"\s+", " ", text).strip()
    return text
```

```
def tokenize_sentences(text):
    return nltk.sent_tokenize(text)

# Step 2: Sentence Embedding with InLegalBERT

def get_sentence_embeddings(sentences):
    inputs = bert_tokenizer(
        sentences,
        return_tensors="pt",
        padding=True,
        truncation=True,
        max_length=512,
    ).to(device)
    with torch.no_grad():
        outputs = bert_model(**inputs)
    embeddings = outputs.last_hidden_state.mean(dim=1)
    return embeddings.cpu().numpy()

# Step 3: Sentence Scoring and Selection
(Extractive Summarization)

def select_top_sentences(sentences, embeddings, top_k=5):
    sim_matrix = cosine_similarity(embeddings)
    scores = sim_matrix.sum(axis=1)
    top_indices = np.argsort(scores)[-top_k:][::-1]
    selected_sentences = [sentences[i] for i in top_indices]
    return selected_sentences

# Step 4: Feeding Extracted Sentences to Fine-tuned BART

def generate_final_summary(selected_sentences):
```

```
input_text = " ".join(selected_sentences)
inputs = bart_tokenizer(
    input_text,
    return_tensors="pt",
    max_length=1024,
    truncation=True,
).to(device)
summary_ids = bart_model.generate(
    inputs["input_ids"],
    num_beams=4,
    length_penalty=2.0,
    max_length=512,
    early_stopping=True,
)
summary = bart_tokenizer.decode(summary_ids[0],
    skip_special_tokens=True)
return summary
```


Chapter 5

Proposed System: Analysis

5.1 Presentation and validation of the results for the proposed system

5.1.1 Quantitative and Qualitative results

Quantitative results

- **ROUGE-1:** Measures the overlap of unigrams (single words) between the generated summary and the reference summary.
- **ROUGE-2:** Measures the overlap of bigrams (pairs of words) between the generated summary and the reference summary.
- **ROUGE-L:** Measures the longest common subsequence between the generated summary and the reference summary, accounting for sentence structure and ordering.
- **BLEU:** Evaluates the n-gram precision between the generated summary and the reference summary, emphasizing exact word matches.
- **METEOR:** Assesses synonym matching, stemming, and word reordering to evaluate the fluency and adequacy of the generated summary.
- **BERTScore:** Measures semantic similarity between the generated summary and the reference summary based on contextual word embeddings, considering meaning rather than exact word matches.

The summarization model, fine-tuned on the Indian Supreme Court case documents using the facebook/bart-large architecture, demonstrated effective performance across multiple evaluation metrics.

The ROUGE-1 score of **0.3609** indicates that a significant portion of important unigrams from the reference summaries were successfully captured. The ROUGE-2 score of **0.1838** reflects the ability of the model to preserve key bi-gram sequences, while the ROUGE-L score of **0.2019** highlights the model’s effectiveness in maintaining the longest common subsequence structure between the reference and generated summaries.

Despite a relatively low BLEU score of **0.0072**, which is common in abstractive summarization tasks due to differences in surface form expressions, the model achieved a METEOR score of **0.1676**, emphasizing its capacity for semantic matching beyond exact word overlaps.

Furthermore, the BERTScore metrics — Precision (**0.8548**), Recall (**0.8326**), and F1-score (**0.8435**) — demonstrate that the generated summaries are highly semantically similar to the reference summaries. This shows that even if the wording differs, the essential meaning and context are well-preserved.

Overall, the fine-tuned BART model successfully generates coherent, semantically rich summaries suitable for legal case analysis under the BNS framework.

Qualitative results

- **Factual Accuracy:** Provided highly relevant and factual sentences directly from the source, preserving legal terminology and structure. However, transitions between sentences were sometimes abrupt.
- **Legal Coverage:** Generated fluent, coherent, and paraphrased content that enhanced readability. Some minor factual shifts were observed in highly technical sections.

- **Interpretability:** Offered the best of both worlds: factual correctness with improved coherence, making them ideal for legal professionals requiring quick yet trustworthy overviews.

5.1.2 Document-wise Performance Evaluation

For this section, we evaluate the performance of different summarization methods across two documents. The evaluation is based on summary statistics, execution times, and various metrics such as ROUGE, BLEU, METEOR, and BERT. The tables below summarize the key results from these evaluations.

Summary Statistics for First Document

Table 5.1 shows the summary statistics for the first document, including the minimum and maximum lengths of summaries generated using different methods. The total number of sentences in the first document is 96. For the Concise and Precise methods, the extractive summaries had lengths between 120 and 160 words, whereas the abstractive summaries ranged from 200 to 320 words. In the Hybrid method, the summaries were slightly longer, with extractive summaries ranging between 120 and 160 words and abstractive summaries reaching 200 to 320 words.

Execution Times for First Document

The execution times for different methods on the first document are shown below. The Concise and Precise methods took relatively shorter times, with Extractive taking 3.13 and 3.19 seconds, respectively. However, Abstractive methods were much slower, with execution times of 33.91 and 45.42 seconds for Concise and Precise, respectively. The Hybrid method, combining both extractive and abstractive strategies, had significantly higher execution times of 102.55 and 176.49 seconds for Concise and Precise summaries.

Evaluation Metrics for First Document

The evaluation metrics, shown in Table 5.2, assess the quality of summaries generated using various methods, including Extractive, Abstractive, and Hybrid

summarization. The results for ROUGE-1, ROUGE-2, ROUGE-L, BLEU, METEOR, and BERT scores are listed, with the highest scores for each metric highlighted in bold. Notably, the Abstractive Concise method achieved the highest ROUGE-1 (0.4444), ROUGE-2 (0.1311), and ROUGE-L (0.2126) scores, indicating that it produced the most informative summaries. Additionally, the BERT score was highest for Abstractive Concise (0.8449), suggesting that it better preserved the semantic meaning of the original text.

Summary Statistics for Second Document

For the second document, the total number of sentences is 59. As shown in Table 5.1, the summary statistics indicate that both extractive and abstractive summaries were generated similarly to the first document, with extractive summaries ranging between 120 and 160 words and abstractive summaries between 200 and 320 words. In the Hybrid method, the length of summaries was again slightly longer. The number of sentences retained was 10.17% for Concise Extractive, and 37.28% for Concise Hybrid, while for Precise, it was 18.64% for Extractive and 66.10% for Hybrid.

Execution Times for Second Document

As per Table 5.3, the execution times for the second document were considerably shorter than those for the first document. The Concise and Precise methods took around 7.04 and 7.35 seconds for Extractive, respectively. The Abstractive methods also took significantly less time than in the first document, with Concise Abstractive taking 11.79 seconds and Precise Abstractive taking 13.83 seconds. The Hybrid method, combining both strategies, took 14.69 seconds for Concise and 19.48 seconds for Precise.

Evaluation Metrics for Second Document

In Table 5.3, the evaluation metrics show that the Abstractive Concise method yielded the best ROUGE scores for the second document as well, with ROUGE-1 (0.4674), ROUGE-2 (0.1967), and ROUGE-L (0.2717) being the highest, suggesting that it generated summaries that retained a high degree of similarity to

the original text. The BERT score for Abstractive Concise (0.8660) also confirmed that it was the most semantically accurate summarization method.

5.1.3 Conclusion

From the evaluation of both documents, it is evident that the Abstractive method consistently outperforms the Extractive and Hybrid methods, particularly when it comes to ROUGE and BERT scores. The Concise Abstractive method achieved the best results across various evaluation metrics for both documents, including ROUGE, BLEU, and METEOR. Additionally, while the Hybrid method showed promise in preserving the structure of summaries, it required significantly more execution time, making it less efficient compared to Abstractive and Extractive methods.

The results suggest that while Abstractive methods yield higher-quality summaries, they come with increased computational costs, especially when summarizing larger documents. On the other hand, the Extractive method, although faster, might not always capture the essence of the document as effectively as the Abstractive method.

In conclusion, the choice of summarization method should be based on the specific requirements of the task, balancing execution time and the quality of the generated summaries.

Table 5.1: Summary Statistics for Concise and Precise Methods

Method	Summary Type	Min Length	Max Length
Concise	Extractive	120	160
	Abstractive	200	320
Precise	Extractive	120	160
	Abstractive	200	320

Table 5.2: Evaluation Metrics for Summarization Methods (First Document)

Method	Evaluation Type	ROUGE-1	ROUGE-2	ROUGE-L	BLEU	METEOR	BERT
Extractive	Concise	0.3639	0.0791	0.1989	0.0241	0.2704	0.8365
	Precise	0.3333	0.0886	0.1681	0.0258	0.2986	0.8381
Abstractive	Concise	0.4444	0.1311	0.2126	0.0298	0.2842	0.8449
	Precise	0.3977	0.1096	0.1949	0.0219	0.3127	0.8347
Hybrid	Concise	0.2681	0.0910	0.1492	0.0239	0.3016	0.8372
	Precise	0.1826	0.0806	0.1086	0.0178	0.2788	0.8250

Table 5.3: Evaluation Metrics for Summarization Methods (Second Document)

Method	Evaluation Type	ROUGE-1	ROUGE-2	ROUGE-L	BLEU	METEOR	BERT
Extractive	Concise	0.3881	0.1053	0.2090	0.0238	0.2356	0.8454
	Precise	0.3448	0.1227	0.1804	0.0282	0.2834	0.8458
Abstractive	Concise	0.4674	0.1967	0.2717	0.0494	0.3200	0.8660
	Precise	0.4180	0.1806	0.2697	0.0392	0.3343	0.8609
Hybrid	Concise	0.3346	0.1602	0.2140	0.0363	0.2903	0.8499
	Precise	0.2725	0.1443	0.1542	0.0248	0.3198	0.8577

5.2 Comparative Analysis with existing systems

The performance of different models for Indian legal document summarization has been compared based on ROUGE evaluation metrics, which are commonly used to assess the quality of summaries. The following table summarizes the ROUGE-1, ROUGE-2, and ROUGE-L scores for various models, highlighting their strengths and weaknesses as observed from the respective papers [2], [10], [11], and [12]. These scores reflect the models' ability to generate summaries that align closely with human-generated references, and the limitations provide insight into areas where each model falls short in handling legal documents.

Table 5.4: Comparative analysis of different models for Indian legal document summarization.

Model	ROUGE-1	ROUGE-2	ROUGE-L	Limitations
InLegalBERT + BART	0.3609	0.1838	0.2019	Fine-tuned for legal domain but struggles with highly complex multi-party cases and rare legal terminologies.
BART [10]	0.1840	0.1143	0.1814	Generic summarization model; lacks legal-specific context understanding, leading to loss of critical information.
LexRank [11]	0.3321	0.1694	0.3012	Purely extractive method; fails to paraphrase or re-structure arguments, making summaries lengthy and less coherent.
TextRank [2]	0.3104	0.1326	0.2757	Dependent on sentence similarity; cannot understand case law context or prioritize legally important information.
BERT-BART [12]	0.3600	0.0990	0.1980	Better at abstraction but inconsistent in handling domain-specific jargon and section referencing in Indian cases.

Chapter 6

Conclusion

This project presents a comprehensive legal document processing system tailored for analyzing cases under the Bharatiya Nyaya Sanhita (BNS) framework. By establishing a robust IPC-BNS mapping and leveraging real-world legal data from Indian Supreme Court case documents, the system successfully adapts existing legal text data to the new legislative context. The platform integrates advanced natural language processing models, particularly InLegalBERT for legal text representation and fine-tunes BART for abstractive summarization tasks. Using a curated dataset of 7,100 case documents, the system achieves strong summarization performance, as measured by standard evaluation metrics like ROUGE, BLEU, METEOR, and BERTScore. Through rigorous experimentation, the platform demonstrates its ability to generate coherent, relevant summaries and to assist in case analysis. Ultimately, this system offers a scalable and efficient tool for legal professionals, enabling rapid and accurate analysis of legal documents in alignment with the evolving Indian legal landscape under the BNS framework.

Appendix-I

The Criminal Code Summarization and Outcome Prediction system assists legal professionals by quickly summarizing criminal cases. It combines extractive and abstractive summarization techniques using InLegalBERT and fine-tuned BART model to enhance accuracy and relevance. The system automates text extraction, summarization, and analysis, mapping legal codes (IPC/BNS) and extracting legal entities to support decision-making.

Description: The system automates criminal case summarization using a hybrid approach that combines extractive and abstractive methods. It processes text extracted from PDFs, cleans and tokenizes it, selects key sentences, and maps relevant legal codes (IPC/BNS) while extracting legal entities.

Remediation: To enhance system performance, the following steps are recommended:

- **Text Extraction:** Improve PDF extraction accuracy, especially for complex formatting.
- **Preprocessing:** Refine text-cleaning techniques to handle legal jargon.
- **Model Refinement:** Fine-tune InLegalBERT and BART for better relevance and accuracy.
- **Validation:** Implement a framework to assess and improve summary quality.
- **Real-Time Processing:** Optimize for real-time document summarization.

References

- [1] S. Paul, A. Mandal, P. Goyal, and S. Ghosh, “Pre-trained Language Models for the Legal Domain,” Jun. 2023, doi: <https://doi.org/10.1145/3594536.3595165>.
- [2] U. Rani and K. Bidhan, “Comparative Assessment of Extractive Summarization: TextRank, TF-IDF and LDA,” *Journal of scientific research*, vol. 65, no. 01, pp. 304–311, 2021, doi: <https://doi.org/10.37398/jsr.2021.650140>.
- [3] A. W. Palliyali, M. A. Al-Khalifa, S. Farooq, J. Abinshed, A. Al-Ansari and A. Jaoua, ”Comparative Study of Extractive Text Summarization Techniques,” 2021 IEEE/ACS 18th International Conference on Computer Systems and Applications (AICCSA), Tangier, Morocco, 2021, pp. 1-6, doi: 10.1109/AICCSA53542.2021.9686867.
- [4] K. A. R. Issam, S. Patel, Subalalitha C. N., “Topic Modeling Based Extractive Text Summarization,” *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, no. 4, pp. 1710–1719, Apr. 2020, doi: <https://doi.org/10.35940/ijitee.f4611.049620>.
- [5] D. F. O. Onah, E. L. L. Pang and M. El-Haj, ”A Data-driven Latent Semantic Analysis for Automatic Text Summarization using LDA Topic Modelling,” 2022 IEEE International Conference on Big Data (Big Data), Osaka, Japan, 2022, pp. 2771-2780, doi: 10.1109/BigData55660.2022.10020259.
- [6] M. R. Ramadhan, S. N. Endah and A. B. J. Mantau, ”Implementation of TextRank Algorithm in Product Review Summarization,” 2020

- 4th International Conference on Informatics and Computational Sciences (ICICoS), Semarang, Indonesia, 2020, pp. 1-5, doi: 10.1109/ICICoS51170.2020.9299005.
- [7] K. Jewani, O. Damankar, N. Janyani, D. Mhatre and S. Gang wani, "A Brief Study on Approaches for Extractive Summariza tion," 2021 International Conference on Artificial Intelligence and Smart Sys-tems (ICAIS), Coimbatore, India, 2021, pp. 601-608, doi: 10.1109/I-CAIS50930.2021.9396031.
- [8] M. Gupta, "Text summarization using TextRank in NLP," Data Science in your pocket, Jul. 18, 2022. <https://medium.com/data-science-in-yourpocket/text-summarizationusing-textrank-in-nlp-4bce52c5b390>.
- [9] R. C. Kore, P. Ray, P. Lade, and A. Nerurkar, "Legal Document Summa-rization Using Nlp and Ml Techniques," International Journal of Engineer-ing and Computer Science, vol. 9, no. 05, pp. 25039–25046, May 2020, doi: <https://doi.org/10.18535/ijecs/v9i05.4488>.
- [10] S. Sharma, S. Srivastava, P. Verma, A. Verma, and Sachchida Nand Chaurasia, "A Comprehensive Analysis of Indian Legal Documents Sum-marization Techniques," SN Computer Science, vol. 4, no. 5, Aug. 2023, doi: <https://doi.org/10.1007/s42979-023-01983-y>.
- [11] P. Trivedi, D. Jain, Shilpa Gite, K. Kotecha, A. Bhatt, and N. Naik, "Indian Legal Corpus (ILC): A Dataset for Summarizing Indian Legal Proceed-ings Using Natural Language," Engineered Science, vol. 27, Nov. 2023, accessed: Apr. 27, 2025, <https://www.espublisher.com/journals/articledetails/1022>.
- [12] A. Shukla et al., "Legal Case Document Summarization: Extractive and Abstractive Methods and Their Evaluation," arXiv preprint, Oct. 2022, <https://arxiv.org/abs/2210.07544>.

- [13] A. Joshi, S. Paul, A. Sharma, P. Goyal, S. Ghosh, and A. Modi, “IL-TUR: Benchmark for Indian Legal Text Understanding and Reasoning,” arXiv (Cornell University), Jul. 2024, doi: <https://doi.org/10.48550/arxiv.2407.05399>.
- [14] M. Lewis et al., “BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension,” Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, doi: <https://doi.org/10.18653/v1/2020.acl-main.703>.
- [15] P. Kalamkar, A. Agarwal, A. Tiwari, S. Gupta, S. Karn, and V. Raghavan, “Named Entity Recognition in Indian court judgments,” arXiv (Cornell University), Jan. 2022, doi: <https://doi.org/10.48550/arxiv.2211.03442>.
- [16] Ilias Chalkidis, T. Pasini, S. Zhang, Letizia Tomada, Sebastian Felix Schwemer, and Anders Søgaard, “FairLex: A Multilingual Benchmark for Evaluating Fairness in Legal Text Processing,” Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Jan. 2022, doi: <https://doi.org/10.18653/v1/2022.acl-long.301>.
- [17] Ilias Chalkidis et al., “LexGLUE: A Benchmark Dataset for Legal Language Understanding in English,” Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Jan. 2022, doi: <https://doi.org/10.18653/v1/2022.acl-long.297>.
- [18] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, “LEGAL-BERT: The Muppets straight out of Law School,” Findings of the Association for Computational Linguistics: EMNLP 2020, 2020, doi: <https://doi.org/10.18653/v1/2020.findings-emnlp.261>.

- [19] "law-ai/InLegalBERT · Hugging Face," Huggingface.co, 2019. <https://huggingface.co/law-ai/InLegalBERT>.
- [20] E. Lab, "IL-TUR," Huggingface.co, 2024. <https://huggingface.co/datasets/Exploration-Lab/IL-TUR>.
- [21] "facebook/bart-large · Hugging Face," huggingface.co. <https://huggingface.co/facebook/bart-large>.

Acknowledgements

We are thankful to a number of individuals who have contributed towards our final year project and without their help; it would not have been possible. First of all, we would like to express our gratitude to Ms. K. Priya Karunakaran, our project guide, for their prompt suggestions, guidance and encouragement over the course of our entire project.

We sincerely appreciate Ms Jayshree Mittal and Ms Snehal Kulkarni, our project coordinators, for their tremendous assistance to our project. We also appreciate the support of the faculty in our department.

We express our sincere gratitude to our respected Director Br. Shantilal Kujur, our Principal Dr. Sincy George and our Head of Department (CMPN) Dr. Kavita Sonawane for providing the facilities, encouragement as well as a conducive environment for learning.

Lastly, we would like to thank our parents and friends who played a crucial role keeping us motivated throughout with their constant nurture and support.

Sincerely,

Samit Fernandes

Jaden Franco

Ralph Pereira