IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026

DOI: 10.17148/IJARCCE.2026.153112

AI BASED LEGAL SIMPLIFICATION AND CASE OUTCOME PREDICTION

Prof. Dnyaneshwar Thombre[1], Mahesh Bhosale[2], Vikas Maurya[3], Intaza Chaudhary[4], Mausam Yadav[5]

Professor, Department of Computer Engineering, Terna Engineering College, Navi Mumbai, Maharashtra, India[1]

Student, Department of Computer Engineering, Terna Engineering College, Navi Mumbai, Maharashtra, India[2-5]

Abstract: It's challenging for many people to understand the complexity of legal documents and legal process. Furthermore, most people do not know what type of legal knowledge they need to better understand legal policies, to forecast the outcome of a case, or to find a legal expert. This project is designed to develop an Artificial Intelligence (AI)-Powered Legal Document Simplifier and Legal Case Outcome Predictor. This project will assist people in understanding the legal document(s) related to their case and the potential outcome of the case.

The project will consist of three major parts. The first part will be the Legal Document Simplifier; using Natural Language Processing (NLP), this component will assist in simplifying legal documents. The second part will be the Case Outcome Predictor; utilizing machine learning algorithms (such as InLegalBERT) to provide a prediction of the outcome of a case (i.e., if the case will be approved or denied); and the rationale supporting the outcome. The final part will be the Lawyer Management System; through which individuals may submit their case data, and connect with a lawyer. This project will utilize AI, NLP, and web development technologies to develop a user-friendly interface that will enable the general public to better understand legal documents.

Keywords: Legal Document Simplification, Case Outcome Prediction, Natural Language Processing, InLegalBERT, Lawyer Management System.

I. INTRODUCTION

The legal system is critical to ensuring both justice and laws in society. Nonetheless, legal documents including policies, contracts, and court decisions contain complicated information that is difficult for the general public to understand [5]. Many members of society lack sufficient knowledge of legal terminology and processes, thereby creating a gap between legal professionals and the general public [9]. The emergence of artificial intelligence (AI) has provided the opportunity to develop AI based legal systems that will assist in simplifying legal information and deriving useful information from legal cases [10]. This can help in making legal information more accessible to the general public by transforming complex legal information into simple forms [2].

The project suggests the creation of an AI-based system named JustiFi that will help to enhance the accessibility and understanding of legal information [9].

The proposed project aims to create an AI-based system called JustiFi to provide better legal information to common persons in an easily comprehensible manner. The proposed system comprises three important components: Legal Document Simplifier, Case Outcome Predictor, and Lawyer Management System. In the Legal Document Simplifier module, a Large Language Model from the Hugging Face Transformers library can be used to create simpler legal document summaries [2], [7]. In the Case Outcome Predictor module, InLegalBERT, a domain-specific language model, can be used to analyze legal case data and provide predictions regarding whether a legal case is accepted or rejected, along with a short explanation of the prediction [1], [6]. Furthermore, the Lawyer Management System will enable users to upload case-related information and contact lawyers for further legal advice.

The integration of AI technology with a user-friendly interface will help to fill the gap between complicated legal information and the general public's understanding [9], [10]. This system will not only increase legal awareness, but it will also help users to take more informed decisions regarding their legal issues.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026

DOI: 10.17148/IJARCCE.2026.153111

II. LITERATURE REVIEW

Recent technological advancements in Artificial Intelligence (AI), and Natural Language Processing (NLP) have greatly impacted legal research. These two most relevant fields to this technology development are Legal Text Summarization (LTS), which aims to improve access to justice, improve efficiency in analyzing case law and scrutinize regulations that govern justice systems; and Legal Judgment Prediction (LJP), which can be used to automatically analyze, mine and parse complex legal judgment data.

1. Legal Text Summarization (LTS)

Summarizing Legal researchers have focused primarily on using Natural Language Processing (NLP) for legal document analysis and summarization. The primary focus of legal document analysis and summarization has been to identify the most relevant information contained within lengthy legal documents. Statistical methods and rule-based methods were traditionally used to summarize lengthy legal documents. However, researchers have increasingly turned to Deep Learning and transformer-based models to develop new systems to perform legal document analysis and summarization. D. Singh [5], identified the importance of NLP in legal document analysis, and further identified three essential components of NLP for legal document analysis including Tokenization, Named Entity Identification and Dependency Parsing. These components were critical in enabling legal document analysis tasks such as legal document summarization and classification. In subsequent work, S. Sharma and P. P. Singh [8], expanded upon the foundational concepts of D. Singh [5]. They proposed a Recursive Summarization Algorithm (RecSumm) to enable recursive processing of long legal documents. RecSumm processed the long legal document by first segmenting the document into multiple segments, then processing each segment separately to create a summary of each segment. The summaries created through RecSumm were then merged to create a single final summary. D. Jain et al. [2], based on the research conducted by S. Sharma and P. P. Singh [8], proposed an Extract-then-Assign (ETA) strategy for constructing an abstractive dataset to overcome the challenges posed by working with long legal documents and data scarcity. ETA was composed of two distinct phases in which the first phase belongs to extractive summarization, was used to extract the most salient information from the source document. In the further phase, abstractive summarization, was then used to transform the extracted salient information into an abstract. The abstract produced by ETA was then used to train and fine-tune overarching models, such as the BART and PEGASUS models.

The RAG model utilizes a retrieval-based approach similar to that of BM25 to obtain both factual and domain correctness in generating a summary.

The authors of [4], Y.-H. Liu and Y.-F. Lin have developed LASG (Legal AI Summarization), a novel application of Artificial Intelligence designed to improve judicial judgments through optimizing the production of condensed versions of briefs and summaries of cases. The work has shown that the integration of Generative Artificial Intelligence with Structured Case Metadata may result in increased reliance on the generated summaries. Similarly, [6] L. Yue et al. have developed Circumstance-Aware Neural Architectures, which provide additional explanation for the summarization process through the addition of context to the case-specific circumstances and reasoning. The authors feel that this will lead to a faster method of using Artificial Intelligence for summarization while maintaining transparency in the process.

2. Legal Judgment Prediction (LJP)

Legal Judgment Prediction (LJP) is focused on predicting the outcome of a legal case based upon textual descriptions of the case, legal statutes, and relevant case law precedents. As such, LJP provides a tool for judicial decision making as well as an opportunity to expedite the classification and allocation of resources for each case.

A. George et al. [1], developed a novel framework, called LegalBERT Embedding with AdaBoost Classifier, that predicts writ petitions, which also provided an example of how specializations can be achieved in using transformers for legal semantic embeddings. The authors have captured some of the potential strengths of transformer-based embeddings in capturing legal semantics. T. Ansari et al. [11] applied a variety of machine learning algorithms (Support Vector Machine (SVM) and Random Forest) for forecasting cases of law with successful outcomes. They validated that traditional machine learning models for classification are competitive when used in conjunction with feature engineering. A. Taming LJP into social justice. J. Wang et al. [3] presented a novel approach of LJP by introducing Graph Reasoning with Supervised Contrastive Learning (GraSCL). GraSCL is a new way of examining LJP as a node classification task. GraSCL enables the reconstruction of relational dependencies among legal elements, e.g., charges, statutes, etc. while overcoming the limitations of LJP, GraSCL offers higher interpretability and robustness to noise.

L. Yue et al. [6] championed a notable, noteworthy extension known as NeurJudge, a Circumstance-Aware Neural Framework mimicking judicial thinking patterns. This method disentangles the factual components into several aspects

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026

DOI: 10.17148/IJARCCE.2026.153111

(e.g., facets in a crime scenario) and studies the hierarchical dependencies. Extending the research - Excited by Neural Reasoning (E-Net), E-NeurJudge promises to bring explainability into the fold through rationale extraction. In this way, we also present a solution to the "black-box" conundrum of Ai-enabled judicial systems.

T. Y. Basha et al., [10] proposed that, as a next step, Generative AI will go beyond predictions to help create documents and analyze them. This will make it inevitable that Legal JP and Legal Document Generation will converge. S. Pandi et al. [9], likewise, emphasized the potential social benefits that can arise from using AI to assist in legal tasks, and called for platforms that combine LTS (Legal Task Support) and LJP (Legal Judgment Prediction) to increase access, clarity and efficiency in delivering justice.

III. SYSTEM ARCHITECTURE & DESIGN

a) System Architecture Overview:

A proposed system will consist of a modular and scalable design with a web interface, an instance of services, artificial intelligence and cloud services. Next.js will be used to construct the user interface layer because it provides role-based user interface designs to the citizens, lawyers and administrators. Using these user interface designs, users will be in a position to upload legal documents, get summaries of the documents, posing legal questions and analyzing data related to the case. The vercel front-end application will be hosted on Vercel which is an efficient content delivery provider. The front-end application will be linked to the backend application with the help of RESTful APIs through HTTPS and JSON protocols. Socket.IO will also provide real-time updates in order to maximize the response delivery of the front-end application.

It will have the backend application developed in Node.js (Express and TypeScript). This will do all the request processing, business logic and other back-end services. Express.js will be used to process the request and then middleware components will be utilized to perform user authentication, role check and validation of the request using JWT-based security protocols. Controllers will be used to process the request and will communicate with service modules to make the system functionalities run. The service layer will be connected to the AI/ML subsystem, which will perform such functions as summarization of legal texts, answering legal questions, and prediction of outcomes with the help of such models as LED Transformer, LLaMA, and InLegalBERT. The explainability layer will also be included to give interpretability and transparency to predictions that the AI will make.

To manage the data, PostgreSQL will be taken, which will be hosted on Supabase. This will provide an appropriate data management in order to store user data, documents, and cases. The data will be handled by a database known as Drizzle ORM that will ease the querying process. In order to enhance the performance, a caching system will be introduced, and it will be stored in Redis. This will enhance performance due to the minimization of the cost of computation. The infrastructure will be comprised of various hosting platforms of the frontend and the backend and the AI can run on various platforms such as Render, VM or cloud hosts.

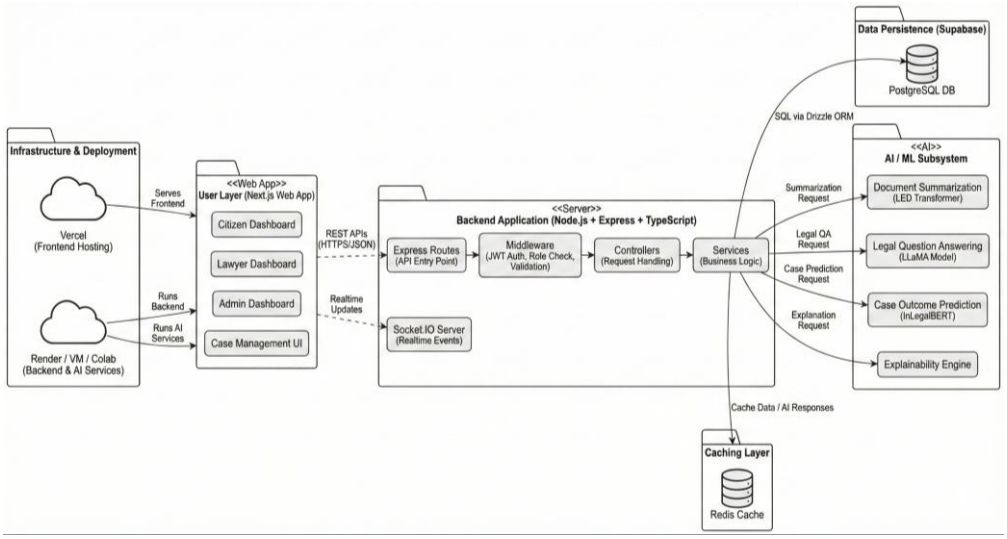

Fig 1: System Architecture

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026

DOI: 10.17148/IJARCCE.2026.153111

b) Document Summarization Using Transformer Model:

The document summarization module has been developed on the basis of creating a concise yet meaningful summary of long legal documents. It starts with a user input of either text data or PDF document using the application interface. The system will read the document and extract the text and convert it into a machine-readable form in the case of PDF input. After the input data is captured it will be sent to the document summarization part and then a transformer based model will be used to process the document. The system will be based on Longformer Encoder-Decoder (LED) model, which is effective in working with the long texts data, including legal documents.

The document summarization module will provide an abstractive summary of the original legal document, and the summary will have the same meaning; this will be in a much smaller quantity of text. Besides this, it also has the offering of flexibility in the output in the different length of the summaries that may be offered according to the needs of the user. The last summary is subsequently forwarded to the user via the application interface.

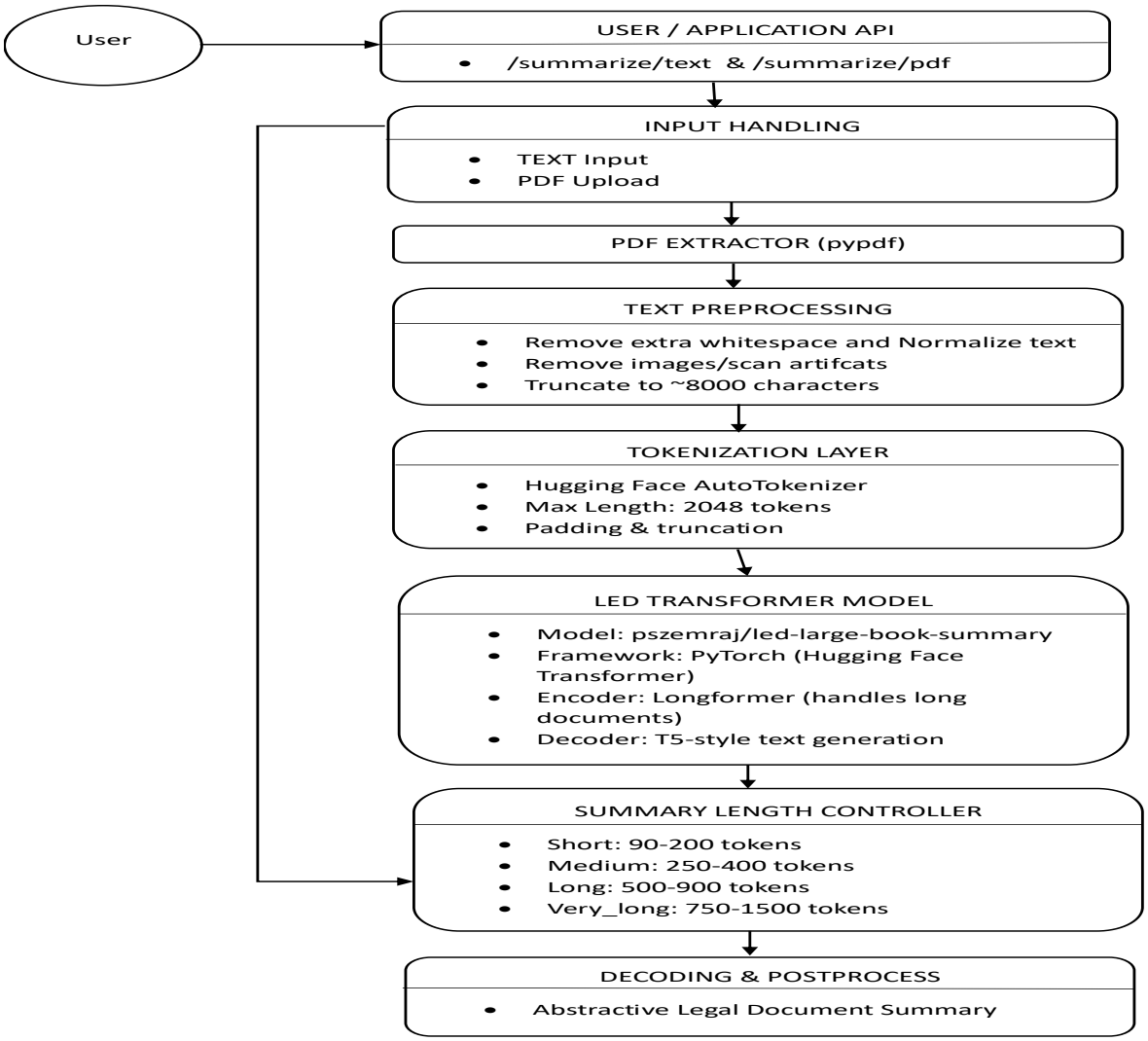

Fig 2: Summarization Module

c) Case Outcome Prediction using InLegalBERT:

The case outcome prediction module will be in charge of analysis of the legal documents it will be supplied with and prediction of the probable result of the case. When the user has supplied a legal document via the application interface, the request is sent via the API layer to the AI model layer.

The system utilises InLegalBERT, a domain transformer model, that was trained on legal text. This enables it to know the language of the law. The model is used to process the document and come up with the necessary features that will be used to predict the probable outcome of the case.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026

DOI: 10.17148/IJARCCE.2026.153111

To be able to work with long legal documents which could have numerous pages of text the system relies on a segmentation approach which assists in dissecting the text into smaller segments that could be processed in order to provide the prediction of the probable outcome of the case. The analysis is then made on the possible outcome of the case to derive the prediction. Besides the forecasted case result, the system offers the user more insights, which can be used in further knowledge of the prediction.

The user can receive the output in the application interface and be able to get the predicted outcome of the case and the explanation of the prediction in an easy-to-understand manner.

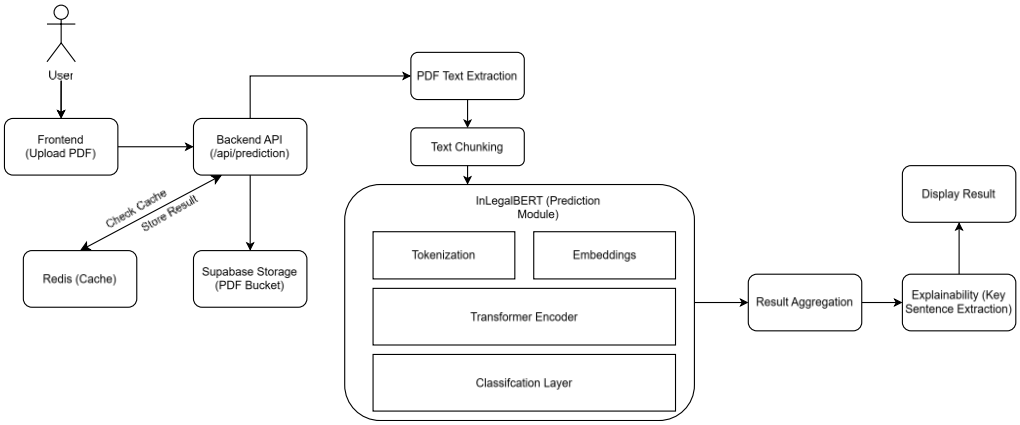

Fig 3: Case outcome prediction model

IV. IMPLEMENTATION AND METHODOLOGY

a) Dataset description:

The data at this research is publicly accessible at different repositories of Indian law, including CJPE (Court Judgments Prediction Evaluation) and NyayaAnumana. This is a real world legal case judgment dataset which was collected in the Supreme Court, different Indian High Courts and Indian tribal courts. They are valid case judgments, which are typically applied in multiple studies of AI, in particular, in the case outcome prediction.

The dataset comprises more than 300,000 legal cases and it can be considered a large-scale dataset that may be successfully utilized to train the deep learning models. The data set is categorical with two (2) groups, one representing the ACCEPT (1) and the other one the REJECT (0) depending on the main outcome of each case. This data is balanced, i.e. there are the same number of two classes which can exclude bias. The dataset is primarily composed of English text with different lengths with average of 3,000-4,000 and a maximum of up to 50,000 words.

Textual characteristics of the dataset, including the number of characters, words, and sentences are as illustrated below:

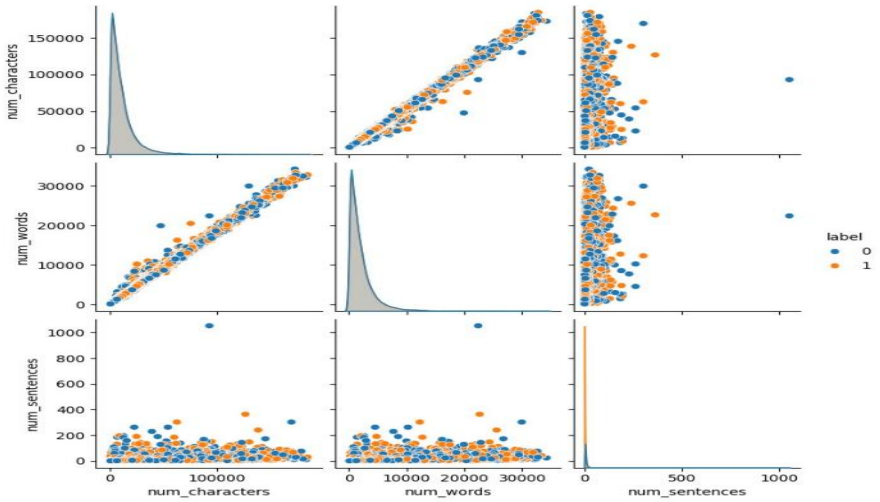

Fig. 4: Distribution and Relationship of Textual Features in the Dataset

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026
DOI: 10.17148/IJARCCE.2026.153111

The visualization is showing the differences in document length and it also shows a strong correlation among text data features like characters and words, which signifies the nature of legal documents.

The preprocessing in this study is mainly minimal because transformer-based models like InLegalBERT already include tokenization and normalization in their architecture. The input text data is directly fed into the model using the BERT tokenizer to transform text into subwords that can be input into the model. Furthermore, due to the nature of legal document length and the input limit for transformer models, the input text data is divided into fixed-size segments consisting of 512 tokens. The predictions are then made at the segment level and aggregated to arrive at the final case outcome.

The data is split into training, validation, and testing sets in line with typical ML practices. In this study, 70% (around 210,000 cases) of the data is used for training, 15% (around 45,000 cases) for validation, and another 15% (around 45,000 cases) for testing.

The data in this study is publicly available legal documents, which is in line with ethical data usage in AI/ML studies. However, it should be noted that the data is only related to Indian legal cases and thus may not be applicable to other legal systems or domains. Nonetheless, this dataset forms a robust basis for developing domain-specific AI models like InLegalBERT for legal outcome prediction tasks.

Property	**Description**
Dataset Name	CJPE (Court Judgments Prediction Evaluation) / NyayaAnumana
Domain	Legal case judgments (Indian courts)
Source	Public legal repositories (Supreme Court, High Courts, Tribunals)
Dataset Size	300,000+ legal cases
Task	Binary Classification – Case Outcome Prediction
Labels	ACCEPT (1) / REJECT (0)
Class Distribution	Approximately balanced (≈50% ACCEPT, ≈50% REJECT)
Language	English Text - legal text
Average Text Length	Average of 3,000–4,000 words
Maximum Text Length	Maximum to 50,000 words
Preprocessing	Miminal preprocessing – handled within transformer
Tokenization	BERT tokenizer – subword tokenization
Chunking Strategy	Split the documents into 512 token segments
Training Split	69% (~3,00,000 cases)
Validation Split	21%(~45,000 cases)
Testing Split	10%(~90,000 cases)
Ethical Consideration	Publicly available data – does not contain any personal information
Limitation	Domain is limited to Indian courts – may not generalize across other jurisdictions

Table 1: Dataset Characteristics

b) Document Summarization Module:

The implementation of the document summarization module is successfully managed and achieved by utilizing a transformer sequence-to-sequence model, which is the Longformer Encoder-Decoder (LED) model, It is implemented in the ‘pszemraj/led-large-book-summary’ package. This Longformer Encoder-Decoder model is particularly designed to suit this problem since it is optimized to process long input sequences, which is quite typical in legal documents.

The legal document is first and foremost preprocessed by conversion of it to a tokenized version, where each word is properly represented by a unique numerical identifier, while the input sequence is significantly padded to a maximum length of 2048 tokens. The attention mask is simultaneously generated to filter the relevant tokens from the available padded tokens, ensuring that the model processes the relevant information correctly.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026

DOI: 10.17148/IJARCCE.2026.153111

The tokenized input is then transferred to the encoder, which is a component of the Longformer Encoder-Decoder model. This encoder is usually an implementation of the Longformer model, which is a transformer sequence-to-sequence model optimized to process long input sequences. The Longformer uses a particularly hybrid attention mechanism, which is a combination of local and global attention. Local attention neatly allows each token to be in attendance to neighboring tokens, while global attention enables some tokens to attend to any token in the sequence. The hybrid attention mechanism has reduced computational complexity to O(n) from O(n^2) achieved by the traditional self-attention mechanism, which is typical in the transformer models.

The encoded input is then sent to the decoder, which generates the sequence by predicting each token in sequence, one token at a time, while generating the sequence in an auto-regressive manner. The probable chances of producing the next token is defined as:

$$\boldsymbol{P(yt \mid y < t, x) = softmax(W \cdot ht + b)}$$

Where, $\boldsymbol{yt}$: The next word (token) to be generated

$\boldsymbol{y < t}$: All the previously generated words (tokens)

$\boldsymbol{x}$: The input document (your text)

$\boldsymbol{ht}$: Hidden state of the decoder

$\boldsymbol{W \cdot ht + b}$: Applying linear transformation

- W = matrix (weightwise)
- b = bias

$\boldsymbol{softmax()}$**:** scores are converted into probabilities

To genuinely increase the quality of the generated summaries, beam search is mainly utilized in the decoding procedure, which also maintains several candidate sequences and selects from this the most probable output. In addition to this, various sampling procedures, such as temperature scaling, top-k sampling, and nucleus (top-p) sampling, are used to balance diversity and coherence in the particular generated text. Moreover, repetition penalty is used to avoid redundant text in the generated summaries to increase readability.

The model generates an abstractive summary, which means it creates new text rather than copying text from the original text. The generated token sequence is decoded back to text in a natural form by making use of the tokenizer, followed by a post-processing step to remove special tokens and make it grammatically correct.

Thus, the LED-based summarization module is used to effectively manage long legal documents while maintaining contextual information, which is important in the legal domain, to create summaries in a manner that is easy to understand.

c) Case outcome prediction module:

The case outcome prediction module is built on top of a pre-trained model named InLegalBERT. This model is specifically tailored to capture the linguistic and contextual characteristics of legal documents. This model is best suited to predict the outcome of cases. The problem is considered to be a binary classification problem. The model predicts the outcome of cases to be either ACCEPTED (1) or REJECTED (0).

The legal document is first passed through the WordPiece tokenizer. This tokenizer converts the legal document into subword representations. The length of the output is limited to 512 tokens. The input token is then converted to a dense vector representation using the embedding layer. This layer uses token embeddings, positional embeddings, and segment embeddings to represent the input.

The input is then passed through a stack of 12 transformer encoder layers. This model uses the self-attention mechanism to capture contextual relationships between the words in the legal document. This allows the model to understand legal arguments and relationships effectively. The attention mechanism is mathematically represented by the formula:

$$\boldsymbol{Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V}$$

where Q, K, and V are query, key, and value matrices, respectively.

Finally, the hidden representation of the [CLS] token is obtained, which represents a summary of the entire document. This is passed through a fully connected classification layer to produce logits for two output classes. The final prediction is obtained using the softmax function as follows:

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026

DOI: 10.17148/IJARCCE.2026.153111

$$y = Softmax(Wx + b)$$

where x is the hidden representation of the [CLS] token, W and b are learnable parameters, and y is the probability distribution over output classes.

Since legal documents are usually lengthy, they are beyond the 512-token limit. Therefore, a chunking approach is used to handle this This generally means that the input document is split into separate overlapping chunks uitilzing a sliding window process. Each chunk will have about 512 tokens, they are with a stride or overlap of 256 tokens. This ensures that there exists some overlap between chunks to capture contextual information. Each chunk is then passed through the particular model to produce prediction probable percentages for each of the chunk.

To produce the final prediction for the entire document, a **probability averaging technique** is used:

$$P(y = k \mid x) = \frac{1}{N}\sum_{i=1}^{N} P(y = k \mid x_i)$$

where **N**: The number of chunks

$\boldsymbol{P(y = k \mid x_i)}$**:** The predicted probability of a class k from the i^{th} chunk which is attainable.

The predicted class label is computed by applying the function argmax to the probabilities.

Result & Evaluation:

The performance of this given proposal of the case outcome prediction model is evaluated in terms of accuracy, precision, recall, and also F1-score. This model is tested with a small dataset of 700 samples, which uses a balanced distribution of classes. The classification accuracy of this model mainly indicates its ability to learn and gather knowledge from the legal documents, and that is found to be 70.29%. The precision and recall for this model for both of these classes are well balanced, indicating its accurate and consistent performance. Going to the REJECT class, with a value of 0, the precision of this model is founded to be 0.7127, and with a recall of 0.7127. In the case of the ACCEPT class, with a value of 1, the precision and recall of the model For Acceptance are discovered to be 0.6923. Further inside, the F1 score of the model for both of the classes is found to be close to approximately 0.70.

Metric	REJECT (0)	ACCEPT (1)
Precision	0.7127	0.6923
Recall	0.7127	0.6923
F1-Score	0.7127	0.6923

Table 2: Class-wise Performance Metrics of InLegalBERT

The model's performance without any bias towards a particular class is further confirmed with the evaluation of the macro-average score and the weighted-average score, which are found to be approximately 0.70.

Metric	Value
Accuracy	0.7029
Macro Average	0.7025
Weighted Avg	0.7029
Total Samples	700

Table 3: Overall Evaluation Metrics of InLegalBERT

Other than the evaluation of the model's classification performance, its learning behavior during training is also mainly analyzed using the evaluation of loss curves. The training loss and validation loss for this model decreases consistently during the training process, showing and indicating its effective learning.

Also further, the validation loss of the model closely monitors and follows the training loss, indicating its minimal overfitting.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026

DOI: 10.17148/IJARCCE.2026.153111

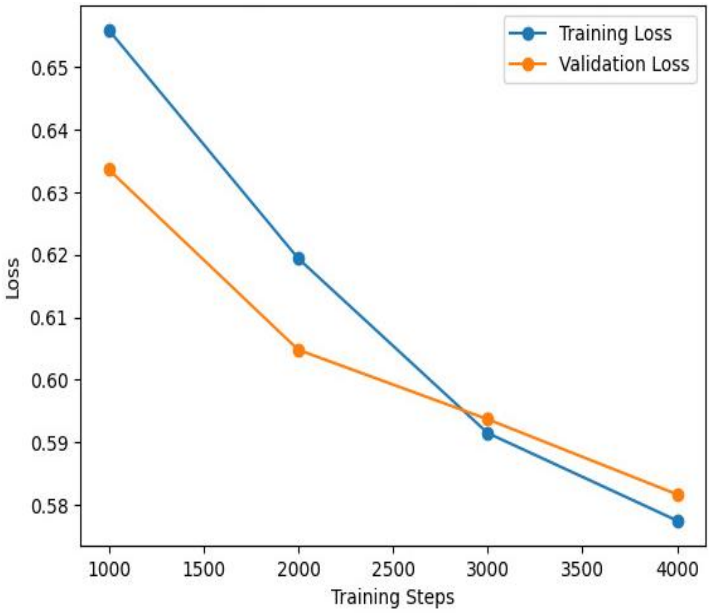

Fig 5: Training vs Validation loss

Besides of this, the precision-recall tradeoff analysis of this model also indicates its consistent balance between two things the precision and recall during the training process. While the precision of the model increases with the number of training steps, its recall shows a bit of variations, indicating that the tradeoff between precision and recall.

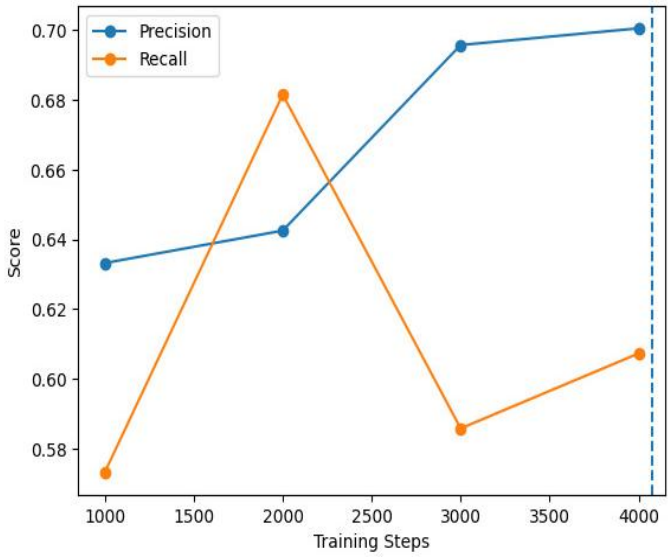

Fig 6: Precision-Recall tradeoff during training

Hence, the results of the analysis of this model indicates that the integration of InLegalBERT with a particular chunking aggregation strategy quite significantly improves the performance of the model, making it very efficient for real-world applications.

d) Case Management System:

The proposed system has a layered and role-based architecture, which provides efficient coordination between system components. The layer management particularly a system which is designed to manage interactions with the users, provide access control mechanism, and manage business logic, also provide integration with artificial intelligence services. The application layer in this provides a clean web-based application utilizing the Next.js framework, which then supports various different user roles, these roles include citizens, lawyers, and administrators. Each of these roles has specific functionalities assigned to them, which can be accessed and used by users. The application layer then provides these users the ability to handle the system using ordered and structured application requests.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026
DOI: 10.17148/IJARCCE.2026.153111

The **authentication and authorization modules** here provide a secure system access utilizing JWT-based authentication system. The mechanism keeps storage of user credentials using hashed password technology, providing security for password storage. This system also provides role-based access control, which further grants users permissions to do specific actions on this system. The middleware components in this verify user identification and provide role-based permissions, which allow users to perform specific actions on the system.

The **backend service layer** mainly provides a processing unit, which processes user requests. The system provides a Node.js-based service using the Express.js framework, which then provides a TypeScript-powered service. The user requests are processed using controllers and service modules, moving to provide case management, proposal processing, message sending, and document procedure. The backend service layer other than this provides AI/ML module processing. The **integration layer** for this AI utilizes various other models that have specialization to deliver enhanced legal assistance. This includes all the particular models that are based on transformers to assist with summarization, question answering, and also prediction. The backend gets all requests that are sent to specialized AI services. Here, the input is processed heavily to get the particular results. These results are then wired back to the backend and finally to the user interface.

The **data management layer** belongs where the system stores and retrieves the data relevant for use. In such case, structured data is handled by PostgreSQL, which is hosted on Supabase. The structured data may include user profiles, case details, and proposals. Large documents, on the other hand, are handled by cloud storage. In order to effectively store large legal documents, the system has then implemented a chunking mechanism. In addition, the system has implemented a caching mechanism, which is powered on Redis, to store the most frequently requested data. The flow mechanism of the legal system is designed in a way that allows user requests to be provided to the backend, where they are then processed by the business logic and AI models, and in the end sent back to the user interface. The structured workflow ensures that the legal system is easy to maintain, hence enhancing the integration of role-based functionality with AI legal services.

V. CONCLUSION

This paper proposes an AI-based legal service system, which includes a legal document summarization system, a case outcome prediction system, and a role-based legal service system. The mechanism is expected to manage simplification for complex legal information and also increase accessibility for users, including for different roles like citizens, lawyers, and admins, using an intelligent architecture system.

This proposed system is a successful combination of modern Natural Language Processing techniques and full-stack powered web development. The document summarization module, which utilizes the Longformer Encoder-Decoder (LED) model, has successfully achieved creating different summaries from lengthy legal documents, which makes it easier to comprehend legal text. The case outcome prediction module utilizes InLegalBERT, which is a transformer based model, to analyze judicial legal texts and predict the outcome with an accuracy close to 70.29%. The use of chunking and probability aggregation is to effectively process lengthy documents while maintaining contextual information. In addition, the system provides capabilities in AI, as well as a role-based legal management system, which includes full case lifecycle management, interactions between lawyers and clients, proposal-based case allocation, and document collaboration. The system provides secure authentication, RESTful APIs, and service layers, which allow seamless communication between system components. The layered design actually allows for scalability, maintainability, and also efficient coordination among the frontend, backend, and AI components.

These results show that the proposed system approach can be quite helpful in legal document analysis and getting decision support while offering a realistic developed system for legal service. The proposed approach, mainly combines AI technology with system design, and it can contribute to the modern conversion of legal practices and further make legal service more accessible. Overall, this paper shows the huge potential of transformer architecture and smart system design in developing an efficient system in the legal domain.

VI. FUTURE SCOPE

The proposed system clearly indicates the potential of using artificial intelligence in simplifying legal processes. There are a number of improvements that can be implemented in the system, which can further enhance its effectiveness. One of the improvements includes the integration of multilingual support, which can be implemented in the system. This can help the system analyze regional language documents, thus increasing its accessibility. The performance of the case prediction module can be enhanced in the system. This can be achieved by training the system on a larger dataset,

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026
DOI: 10.17148/IJARCCE.2026.153111

including different case law. The use of advanced transformers can also be integrated in the system, which can further enhance its performance. The use of explainable AI can also be increased in the system.

From a system perspective, it can be enhanced by adding real-time features such as notifications, live chat, and video consultation between lawyers and clients. The inclusion of secure payment gateway services and appointment scheduling will add to the practicality of the system.

In addition, the system can be expanded to accommodate cross-jurisdictional legal analysis. This will enable it to handle international legal cases. Advanced analytical tools can be incorporated to offer personalized lawyer suggestions.

In conclusion, future enhancements of the system will be beneficial to it. They will add to the system's scalability, accuracy, practicality, and global applicability. This will make this a more enhanced and comprehensive solution for intelligent legal assistance.

REFERENCES

[1] A. George, P. Prabhakar, A. Jose, P. B. Pati, A. F, and S. Parida, "Enhancing Legal Decision Making: WRIT Case Outcome Prediction with LegalBERT Embeddings and AdaBoost Classifier," *2024 IEEE International Conference on Contemporary Computing and Communications (InC4)*, 2024.

[2] D. Jain, M. D. Borah, and A. Biswas, "Summarization of Lengthy Legal Documents via Abstractive Dataset Building: An Extract-then-Assign Approach," *Expert Systems With Applications*, Volume 237, 2024, pp 121571.

[3] J. Wang, Y. Le, D. Cao, S. Lu, Z. Quan, and M. Wang, "Graph Reasoning With Supervised Contrastive Learning for Legal Judgment Prediction," *IEEE Transactions on Neural Networks and Learning Systems*, Volume 36, Issue 2, 2025, pp 2801-2815.

[4] Y.-H. Liu and Y.-F. Lin, "LASG: Streamlining Legal Adjudication with AI-Enabled Summary Generation," *2024 IEEE 48th Annual Computers, Software, and Applications Conference (COMPSAC)*, 2024, pp 2242-2247.

[5] D. Singh, "Legal documents Text Analysis using Natural Language Processing (NLP)," *2024 2nd International Conference on Self Sustainable Artificial Intelligence Systems (ICSSAS)*, 2024, pp 1302-1307.

[6] L. Yue, Q. Liu, B. Jin, H. Wu, and Y. An, "A Circumstance-Aware Neural Framework for Explainable Legal Judgment Prediction," *IEEE Transactions on Knowledge and Data Engineering*, Volume 36, Issue 11, 2024, pp 5453-5467.

[7] S. A. Mukund and K. S. Easwarakumar, "Optimizing Legal Text Summarization Through Dynamic Retrieval-Augmented Generation and Domain-Specific Adaptation," *Symmetry*, Volume 17, Issue 5, 2025, Article no. 633.

[8] S. Sharma and P. P. Singh, "Advancing Legal Document Summarization: Introducing an Approach Using a Recursive Summarization Algorithm," *SN Computer Science*, Volume 5, 2024, Article no. 927.

[9] S. Pandi, A. M. Farook, S. K. Kannaiah, and W. Kingston, "Enlightening Justice: Empowering Society Through Al-driven Legal Assistance," 2024 Second International Conference on Advances in Information Technology (ICAIT), 2024.

[10] T. Y. Basha, B. Kalyani, and Y. Sandeep, "Generative Artificial Intelligence in Legal Drafting," 2024 International Conference on Computational Intelligence for Green and Sustainable Technologies (ICCIGST), 2024.

[11] T. Ansari, H. S. Dhillon, and M. Singh, "Machine Learning Model to Predict Results of Law Cases," 2024 Fourth International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT), 2024.

[12] P. Szemerényi, "LED Large Book Summary Model," https://huggingface.co/pszemraj/led-large-book-summary

[13] Legal AI Lab, "InLegalBERT Model," https://huggingface.co/law-ai/InLegalBERT

