AN INTELLIGENT AI CHATBOT FOR LEGAL CASE PREDICTION USING BERT

A PROJECT REPORT

Submitted by

U AKSHIDHA [211420104014]

DEEPTHI L S [211420104058]

MONISHA D [211420104167]

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University,Chennai)

MARCH 2024

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report **“AN INTELLIGENT AI CHATBOT FOR LEGAL CASE PREDICTION USING BERT”** is the bonafide work of “**AKSHIDHA U (211420104014) DEEPTHI L S (211420104058) MONISHA D (211420104167)** ” who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
DR L.JABASHEELA M.E., Ph.D., **PROFESSOR AND HEAD,**	**DR KAVITHA SUBRAMANI M.E., Ph.D.,** **SUPERVISOR** **PROFESSOR**
Department of CSE, Panimalar Engineering College, Chennai - 123	Department of CSE, Panimalar Engineering College, Chennai - 123

Certified that the above candidate(s) was examined in the End Semester Project Viva-Voce Examination held on ....

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION BY THE STUDENT

We **AKSHIDHA U (211420104014), DEEPTHI L S(211420104058), MONISHA D (211420104167)** hereby declare that this project report titled "**AN INTELLIGENT AI CHATBOT FOR LEGAL CASE PREDICTION USING BERT**", under the guidance of **Dr. KAVITHA SUBRAMANI, M.E., Ph.D.,** is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

AKSHIDHA U

DEEPTHI L S

MONISHA D

ACKNOWLEDGEMENT

Our profound gratitude is directed towards our esteemed Secretary and Correspondent, **Dr. P. CHINNADURAI, M.A., Ph.D**., for his fervent encouragement. His inspirational support proved instrumental in galvanizing our efforts, ultimately contributing significantly to the successful completion of this project

We want to express our deep gratitude to our Directors, **Tmt. C. VIJAYARAJESWARI, Dr. C. SAKTHI KUMAR, M.E., Ph.D., and Dr. SARANYASREE SAKTHI KUMAR, B.E., M.B.A., Ph.D.,** for graciously affording us the essential resources and facilities for undertaking of this project.

Our gratitude is also extended to our Principal, **Dr. K. MANI, M.E., Ph.D.,** whose facilitation proved pivotal in the successful completion of this project.

We express our heartfelt thanks to **Dr. L. JABASHEELA, M.E., Ph.D.,** Head of the Department of Computer Science and Engineering, for granting the necessary facilities that contributed to the timely and successful completion of project.

We would like to express our sincere thanks to **Project Coordinator** and **Project Guide**, **Dr. KAVITHA SUBRAMANI, M.E., Ph.D.,** and all the faculty members of the Department of CSE for their unwavering support for the successful completion of the project.

AKSHIDHA U

DEEPTHI L S

MONISHA D

ABSTRACT

Law and society are the most inseparable attainments of mankind. The legal system and jurisprudence have the greatest significance in the development of a society. With the advent of human civilization began the formulation of the judicial system. A legal case prediction chatbot that employs cutting-edge BERT (Bidirectional Encoder Representations from Transformers) - based natural language processing models promises to streamline legal research and improve decision-making processes. The chatbot assists legal practitioners in forecasting case outcomes, locating important precedents, and providing insights into potential legal strategies through analyzing and comprehending the intricate details of legal texts. This novel use of BERT in the legal arena seeks to give legal professionals and stakeholders with an efficient and accurate tool for predicting case outcomes, hence promoting informed decision-making in the legal landscape. The proposed BERT method is compared to its different variants such as the BERT (base) and BERT (large). We have acquired an accuracy of 80% and above using the BERT(base) and which performed better than the BERT(large) eventhough it has less number of encoders than the BERT(large) model.

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI	-	Artificial Intelligence
BERT	-	Bidirectinal Encoder Representations for Transformers
CNN	-	Convolutional Neural Network
CSS	-	Cascading Style Sheets
DCRS	-	Dialogue-based Conversational Recommender Systems
DFD	-	Data Flow Diagram
HTML	-	Hyper Text Markup Language
JS	-	Java Script
NLP	-	Natural Language Processing
PLM	-	Pre-trained Language Models
UI	-	User Interface
UML	-	Unified Modelling Language

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

1. INTRODUCTION

1.1 PROBLEM STATEMENT

"Law and society are inextricably linked in our human experience. The rules we follow evolved alongside civilization. Understanding the legal system plays a crucial role in the development of communities. Making sure everyone follows the rules helps to shape our civilization. However, as our environment evolves, so does our understanding of laws. Unfortunately, many schools do not provide adequate legal education, leading to a lack of understanding of the legal system. Dealing with a lawsuit can be challenging, especially for those who are unfamiliar with the process. Legal decision systems have been established, based on AI, by predicting verdicts automatically to support lawyers. A legal automation system called AI lawyer was invented to predict verdicts in the United States on May 2016, and since, constant efforts have been made to develop its accuracy. It reads a vast number of judgment documents and analyzes the contents based on a special algorithm to draw decisions to judge a case automatically. It becomes more difficult and time consuming for human lawyers to sentence correct verdicts in legal judgments, because there have been significantly increased numbers of lawsuits in the recent past.

In recent years, the convergence of artificial intelligence (AI) and the legal realm has seen transformational advances, altering the landscape of legal services. In the rapidly evolving landscape of legal practice, the integration of artificial intelligence has ushered in transformative capabilities, and the deployment of intelligent AI chatbots represents a groundbreaking development in the realm of legal case prediction. One notable stride in this direction is the advent of the Bidirectional Encoder Representations from Transformers (BERT) model, a breakthrough in natural language processing (NLP). Leveraging the power of BERT, we propose an intelligent AI chatbot designed to revolutionize legal case prediction. The evolution of Natural Language Processing (NLP) and chatbots in the legal domain has been transformative, enhancing efficiency and accessibility in legal services. Initially, legal professionals faced challenges in adapting to evolving language patterns, complex legal documents, and vast amounts of textual information. However, advancements in NLP have enabled the development of specialized legal chatbots that can comprehend legal jargon, analyze case law, and provide relevant information swiftly. These tools streamline legal research, offering lawyers quick access to precedents, statutes, and case-related insights, thereby augmenting the decision-making process and saving valuable time. Legal chatbots have evolved from basic rule-based systems to sophisticated models incorporating machine learning and contextual understanding. They facilitate client interaction, offering legal guidance, document review, and even generating draft contracts. The integration of NLP with legal chatbots continues to progress, presenting opportunities for improved user experiences, enhanced legal research capabilities, and the potential for predictive analytics to forecast case outcomes. As these technologies advance, the legal domain witnesses a significant paradigm shift, making legal services more accessible, efficient, and responsive to the evolving needs of legal professionals and clients alike. Legal cases related to privacy infringement are outcomes of interactions between society and technological factors, since the cases often comprise violation of law, human errors, personal information owners' perception, etc. It means that recent privacy legal cases have shown different characteristics from other legal incidents. For example, the case of stealing someone's property is always considered illegal, regardless of time and space. However, in most privacy infringement cases, it is very difficult to clearly identify a responsible party. Legal cases regarding improper usage of adware to invade an individual's privacy is a representative case that demonstrates that legal judgments may be influenced by technological and social environmental factors.

Legal professionals grapple with vast volumes of information, complex case precedents, and nuanced interpretations of the law. The traditional processes of legal case analysis can be time-consuming and resource-intensive, often leading to delays and potential oversights. In response to these challenges, an intelligent AI chatbot for legal case prediction emerges as a technological ally, driven by the fusion of AI and legal expertise.The motivation behind developing such a chatbot stems from the need to enhance legal workflows, streamline decision-making processes, and empower legal practitioners with predictive insights. By harnessing the power of AI, this chatbot aims to revolutionize how legal professionals approach case analysis, offering a tool that not only accelerates the pace of legal research but also augments the depth and accuracy of predictions.

The integration of BERT into the legal domain addresses the unique challenges posed by the specialized language and intricacies of legal documents. Traditional keyword-based searches often fall short in capturing the nuanced relationships within legal texts, whereas BERT excels in contextual comprehension. By employing a BERT-based model within an AI chatbot framework, legal professionals gain an intelligent virtual assistant capable of not only predicting case outcomes but also engaging in natural language conversations. This conversational aspect enhances user interaction, allowing legal professionals to query the chatbot for insights, explanations, and guidance, thereby democratizing access to legal knowledge and fostering a more collaborative approach to case analysis. The legal profession is inherently information-intensive, with practitioners traversing massive amounts of legal texts, precedents, and case law in order to deliver effective advice and predict case outcomes. The complexity and dynamic nature of legal language provide significant problems to typical information retrieval systems. The limits of keyword-based searches and inflexible rule-based techniques demand novel solutions that can understand the complexities of legal speech. The significance of this research lies in its potential to empower legal professionals with a cutting-edge tool that streamlines case analysis, optimizes research efforts, and enhances the overall efficiency of legal practice. By integrating the power of the BERT model into a user-friendly chatbot, we aim to contribute to the advancement of intelligent systems in the legal domain.

As the legal landscape continues to evolve, an intelligent AI chatbot for legal case prediction powered by BERT emerges as a cutting-edge solution, bridging the gap between traditional legal research and the capabilities offered by advanced natural language processing models. This technology stands poised to reshape how legal professionals approach case analysis, providing them with a sophisticated tool that not only predicts outcomes but also enhances the efficiency, accuracy, and accessibility of legal research and decision-making. Our intelligent AI chatbot aims to bridge this gap by incorporating BERT, a cutting-edge language representation model, into the core of legal case prediction. By amalgamating cutting-edge AI technology with the intricacies of legal reasoning, our intelligent AI chatbot aspires to be a valuable companion for legal professionals, enhancing their capabilities, reducing research time, and improving the overall efficiency of legal case analysis and prediction. In the subsequent sections, we will look at the technical architecture, training approaches, and ethical considerations that have guided the development of this efficient AI-powered legal assistant.

CHAPTER 2
LITERATURE REVIEW

2. LITERATURE REVIEW

2.1 LITERATURE SURVEY

McConnell, Zhu, Pandya, and Aguiar [1] utilize ensemble learning methods, such as Adaboost and tree-based models, to predict legal case outcomes. Their study reveals superior accuracy in forecasting motion to strike outcomes in vehicular cases compared to tort cases, except for SVM. However, a limitation arises from courts occasionally granting motions to strike in part. Addressing this challenge could enhance prediction accuracy in legal decision-making. Song, Gao, He, and Schilder [2] present an overview of pre-trained Language Models (PLMs) in legal assistance, highlighting their strengths and limitations. They propose text selection as a promising direction to enhance PLMs' effectiveness in legal tasks, emphasizing the importance of optimizing input data. This research provides valuable insights for leveraging PLMs in legal domains and suggests avenues for future exploration. Amato, Fonisto, Giacalone, and Sansone [3] develop an intelligent AI for question answering using SBERT, with the fine-tuned quora-distilbert-multilingual model showing superior performance. However, challenges, primarily due to dataset size limitations, are identified. Future research could focus on addressing these challenges to enhance the effectiveness of SBERT-based question answering systems.

Another research by Ge, Huang, Shen, Li, and Hu [4] demonstrates the potential of Multi-level Matching Networks in significantly improving recommendation accuracy through fine-grained fact-article correspondences. However, challenges arise due to the semi-structured nature of judgment documents, hindering automatic extraction of crucial elements like facts and cited articles. Overcoming these challenges is essential to fully leverage the benefits of this approach in recommendation systems. Said, Khudoberganov, Sharopov, and Abduvaliev analyze [5] international legal frameworks and practices regarding AI integration in courtrooms, offering recommendations based on primary and secondary sources. They advocate for ethical guidelines, transparency in algorithms, sound data management, human oversight, professional education, and international cooperation. Concerns about fairness, bias, accuracy, and accountability underscore the need for critical evaluation and ethical considerations in AI integration within legal systems. Hassani and Silva [6] conduct a review of ChatGPT's role in data science, highlighting its potential to address skill gaps, generate synthetic data, and drive innovation. However, concerns include inaccuracies, biased training data, and limitations in handling complex inquiries, contrasting with Google's search algorithms' reliability. Future research should focus on mitigating these challenges to enhance ChatGPT's reliability and utility in data science. In [7], Franca, Boaro, dos Santos, et.al utilize XGBoost in ensemble learning, demonstrating its ability to balance sensitivity and specificity for accurate predictions. However, a drawback lies in its inability to handle high cardinality categorical variables effectively. Addressing this limitation could further enhance its applicability in predictive modelling tasks. BERT variations for multilingual legal document classification was explored by Niklaus, Chalkidis, and Stürmer [8], finding that innovative models like Long BERT and Hierarchical BERT outperform standard BERT, particularly in Macro F1 Score. While performance is consistent across German and French subsets, challenges persist in the Italian subset, suggesting the need for further investigation.

Giacalone and Salehi [9] explore the integration of fair division theory and algorithmic-based dispute resolution in civil law, highlighting the potential of game-theory algorithms. While these methods offer efficiency and fairness enhancements, challenges in harmonizing them within diverse legal landscapes are identified. Future research could focus on developing adaptive approaches to accommodate legal diversity while ensuring fairness in dispute resolution. In their research Masha Medvedeva, Michel Vols and Martijn Wieling [10] employed super vector machines that has shown promise in predicting legal outcomes with 75% accuracy using simple input data. However, a significant drop in performance occurs with a 10-year data gap, likely due to reduced training data availability. Future research should focus on addressing data gaps to enhance SVM predictive accuracy in legal decision-making. Queudot, Charton, and Meurs [11] utilize diverse NLP models to analyze discussions without specialized annotations, showcasing their versatility across different datasets.

However, their study reveals a tendency for the model to produce conservative, uninteresting responses. Future work should focus on refining algorithms to encourage more engaging and diverse outputs in discussion analysis. Zheng, Liu, and Sun [12] introduce KD-BERT, a knowledge distillation-based legal decision prediction model known for its faster inference speed compared to other BERT models. However, the model's extensive 115M encoder parameters pose challenges for practical application in decision prediction algorithms. Future research could focus on optimizing model complexity while maintaining predictive accuracy in legal contexts. Limsopatham [13] investigates BERT's adaptation for legal text classification, highlighting its proficiency in various NLP tasks. While BERT shows promise, challenges like high pre-training costs and handling long documents impact its performance. Future research should focus on mitigating these challenges to enhance BERT's utility in legal text classification.

Wehnert, Sudhi, Dureja, Kutty, Shahania, and De Luca [14] explore combining Sentence-BERT and LEGAL-BERT with TF-IDF vectors, employing BERT Score for similarity scoring in statute retrieval tasks. Their research involves data enrichment, augmentation, and hyperparameter optimization, leading to improved recall and winning outcomes. However, challenges include the lack of explainability in BERT-based methods, increased complexity in combining Sentence-BERT with TF-IDF, and variability in BERTScore's performance. Despite these challenges, their strategies demonstrate significant advancements in statute law retrieval tasks. Biresaw and Saste [15] explore the impact of Artificial Intelligence (AI) on legal research, particularly through Legal AI tools. They highlight challenges in addressing legal issues requiring human empathy and creativity while meeting client expectations. Concerns include limited contextual understanding, ethical considerations, and potential job displacement. Balancing AI benefits with human qualities is crucial for navigating these challenges in the legal profession. In their study on critiquing mechanisms for Dynamic Conversation-based Recommendation Systems (DCRS), Cai, Jin, and Chen [16] identify effective techniques to improve user interaction with recommendation chatbots across diverse needs. While personalized recommendations enhance user engagement, they caution that over-alignment with current preferences could narrow users' exploration space. Future research should focus on adaptive critiquing strategies to balance personalization and recommendation diversity in DCRS. CNN with PCA for legal data analysis was implemented by Shang [17], achieving superior results compared to benchmarks across four datasets. However, the CNN model requires further fine-tuning to adapt to changing legal needs effectively. This research highlights the promising potential of deep learning techniques in legal data analysis while emphasizing the importance of ongoing refinement for optimal performance.

2.2 SURVEY TABLE

Table 2.2.1 Survey Table

Title of the paper	**Authors**	**Methodology**	**Advantages**	**Disadvantages**
Improving Access to Justice with Legal Chatbots,MDPI,2020.	Marc Queudot, Eric Charton, Marie-Jean Meurs.	Uses several NLP models like bag-of-words, word embeddings & Recurrent Neural Networks (RNN)	The only data needed are discussions: no special annotation is required, so many corpora are usable.	The model tends to favor low-risk responses, which often amount to very short and uninteresting answers.
Study of Deep Learning-Based Legal Judgment Prediction in Internet of Things Era, Hindawi , 2022.	Min Zheng, Bo Liu, Le Sun	Implemented a knowledge distillation-based legal decision	The KD-BERT model’s inference speed is also much faster than the	It has more than 115M encoder parameters, which seriously hinders the

		prediction model, called KD-BERT.	other BERT models.	application of decision prediction algorithms.
Effectively Leveraging BERT for Legal Document Classification,ACL,2021	Nut Limsopatham	Investigates BERT adaptation for legal text classification, fine-tuning variants on legal datasets and comparing their performance with models pre-trained on different documents,	BERT excels in various NLP tasks on legal texts, leveraging transfer learning from pre-training on large corpora, adapting for longer documents, and showing improved performance through domain-specific pre-training.	High cost associated with pre-training BERT on large corpora, challenges in effectively handling long legal documents which can impact performance, and the risk of data loss when discarding parts of documents.
Legal Norm Retrieval with Variations of the BERT Model Combined with TF-IDF Vectorization,ACM,2021.	Sabine Wehnert, Viju Sudhi, Shipra Dureja, Libin Kutty, Saijal	Focusing on combining Sentence-BERT and LEGAL-BERT with	Multiple strategies, like data decomposition, augmentation, two-stage TF-	BERT-based methods may lack explainability compared to TF-IDF, potentially

	Shahania, Ernesto W. De Luca	TF-IDF vectors	IDF with Sentence-BERT, thresholding, and ensemble methods.	compromising precision; combining Sentence-BERT with TF-IDF and data enrichment adds complexity
The Impacts of Artificial Intelligence on Research in the Legal Profession, SciencePG,2022.	Samuel Maireg Biresaw, Abhijit Umesh Saste	Artificial Intelligence, Legal Rersearch Disruption,	Legal problems that will reqiure human empathy, judgment and creativity and satisfy client expectations.	Limited contextual understanding, ethical considerartions and potential job displacement concerns.
Task-Oriented User Evaluation on Critiquing-Based Recommendation Chatbots, IEEE, 2022.	Wanling Cai, Yucheng Jin, and Li Chen.	Critiquing mechanisms for DCRS.	The findings suggest effective critiquing techniques to enhance the interaction between users and the recommendation chatbot	Personalized recommendations to users are too aligned with their current preferences, which may lead to increasingly narrower exploration space over time.

A Computational Intelligence Model for Legal Prediction and Decision Support, Hindawi, 2022.	Xuerui Shang	Implemented CNN with PCA.	Compared to the benchmark method, their proposed algorithm achieved the best results on four datasets.	The CNN model requires further fine-tuning to be robust to the changing needs of law.

CHAPTER 3
THEORETICAL BACKGROUND

3. THEORETICAL BACKGROUND

3.1 EXISTING SYSTEM

Several existing systems in the legal domain leverage chatbot technology to enhance various aspects of legal practice. One such example is ROSS Intelligence, a legal research chatbot powered by artificial intelligence. ROSS uses natural language processing to understand legal queries and provides relevant case law, statutes, and legal precedents. While it streamlines legal research, its limitations may include potential challenges in handling highly specialized or nuanced legal topics and the need for continuous updates to stay current with evolving legal standards.

Another notable system is LawBot, which focuses on providing legal information to individuals. LawBot assists users in understanding their legal rights by asking questions in a conversational manner. Limitations may arise in cases where legal situations are highly complex, requiring personalized advice that surpasses the capabilities of a generalized chatbot. Additionally, there could be challenges in ensuring the accuracy and comprehensiveness of the legal information provided.

Furthermore, DoNotPay is a legal chatbot designed to help users navigate and contest parking tickets. It simplifies the legal process by generating appeal letters and providing guidance on contesting fines. However, the scope of DoNotPay is limited to specific legal issues, and its effectiveness may vary depending on the jurisdiction and the complexity of the legal matter. These limitations highlight the need for continued advancements in chatbot technology to address domain-specific complexities.

DISADVANTAGES OF EXISTING SYSTEM

- Personalized recommendations to users are aligned with their current preferences, which may lead to increasingly narrower exploration space over time.

- They validate their results with only music recommendation domain. The results need to be further validated in other domains.

3.2 PROPOSED METHODOLOGY

The proposed system introduces an intelligent AI chatbot designed to revolutionize legal case prediction by incorporating state-of-the-art technologies, specifically leveraging the Bidirectional Encoder Representations from Transformers (BERT) model. This advanced chatbot aims to assist legal professionals in making well-informed decisions by predicting case outcomes, analyzing legal trends, and enhancing overall case management.

The core functionality of the chatbot revolves around its ability to comprehend and analyze legal documents with a high level of contextual understanding, courtesy of the BERT model. BERT's bidirectional training allows the chatbot to grasp intricate relationships within legal language, ensuring nuanced interpretation of case details. Through a natural language interface, legal professionals can interact seamlessly with the chatbot, posing queries related to specific cases or seeking predictions on potential outcomes.

The key features of the proposed system include:

- Contextual Understanding: The chatbot utilizes the BERT model to capture contextual nuances in legal language, enhancing its ability to interpret and analyze case details effectively.
- Predictive Analytics: Leveraging machine learning techniques, the chatbot predicts case outcomes based on historical data, legal precedents, and the unique details of the case at hand. This predictive analytics feature aids legal professionals in anticipating potential results.
- Legal Research Assistance: The chatbot assists legal practitioners in conducting comprehensive legal research by extracting relevant information from diverse

legal sources. This feature enhances the efficiency of legal professionals in staying abreast of current legal standards.

3.2.1 SYSTEM ARCHITECTURE

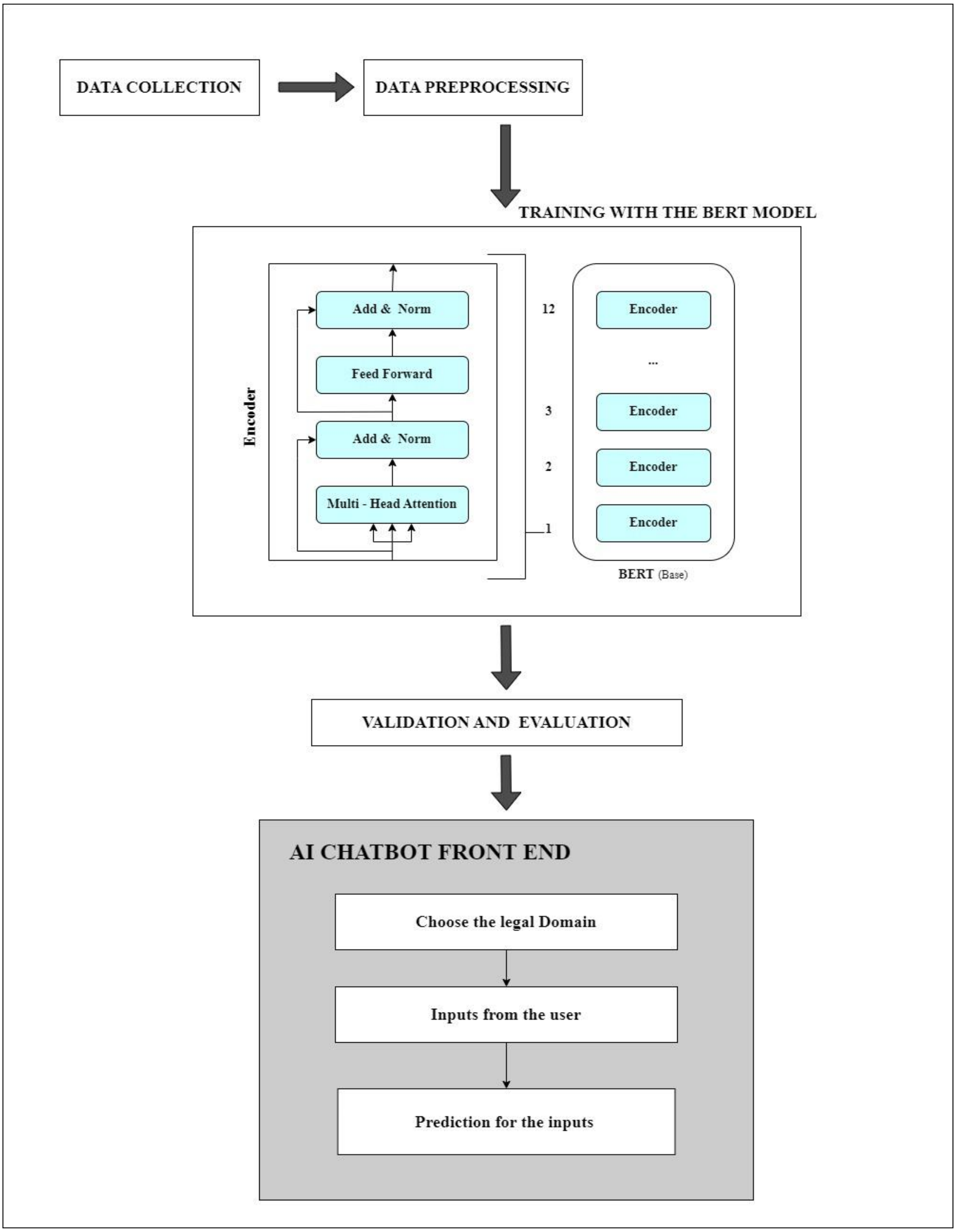

Figure 3.2.1 System Architecture Diagram

The system architecture for our intelligent AI chatbot designed for legal case prediction is a sophisticated framework that encapsulates the complexities of legal language and effectively employs transformer-based models, particularly BERT (Bidirectional Encoder Representations from Transformers). This architecture is meticulously designed to handle various stages, starting from data collection to the integration of BERT for model training, validation, and evaluation, culminating in a user interface that facilitates intuitive interactions.. At its core, the system leverages BERT, a powerful transformer-based model known for its ability to capture intricate contextual relationships in natural language, making it particularly well-suited for the nuanced complexities of legal documents. We'll delve into the intricate modules of data collection, preprocessing, training using BERT, validation and evaluation, and the user interface to provide a comprehensive understanding of the architecture.

The initial module of data collection involves gathering an extensive and diverse dataset of legal documents, judgments, and case law. This corpus forms the foundation for training and fine-tuning the BERT model, enabling the chatbot to learn patterns and nuances inherent in legal language. The collected data is curated to cover a wide spectrum of legal scenarios, ensuring the model's adaptability and robustness. Subsequently, the preprocessing module transforms raw textual data into a format suitable for BERT's input requirements.

The heart of the system architecture lies in the training module, where BERT is utilized to imbue the chatbot with a deep understanding of legal semantics. The BERT model, chosen for its ability to capture bidirectional context in language, is fine-tuned on the legal dataset, adapting its parameters to the intricacies of legal texts. This phase involves leveraging transfer learning, utilizing the pre-trained BERT model's knowledge on general language understanding and fine-tuning it for the specialized domain of legal prediction. The training module ensures that the chatbot can effectively interpret and generate predictions based on the legal corpus.

A crucial consideration in optimizing the system architecture is the choice between BERT and BERT Large. While BERT provides an excellent baseline, BERT Large, with a more extensive architecture comprising 340 million parameters, offers increased model capacity. The comparative analysis involves evaluating the trade-offs between computational resources and performance. BERT Large, with its higher capacity, has the potential to capture more intricate patterns and relationships within legal texts, potentially enhancing the chatbot's predictive capabilities. However, it comes at the cost of increased computational requirements. The decision between BERT and BERT Large is influenced by factors such as the size of the dataset, available computational resources, and the complexity of the legal predictions required.

The validation and evaluation module plays a pivotal role in assessing the chatbot's performance and ensuring its reliability in real-world legal scenarios. The system architecture incorporates validation sets to fine-tune hyperparameters and optimize the model's generalization capabilities. Rigorous evaluation metrics, such as precision, recall, and F1-score, are employed to measure the chatbot's accuracy in predicting legal outcomes. The validation and evaluation process iterates until the model achieves satisfactory performance on the legal dataset.

The user interface, the final module, encapsulates the chatbot's interaction with legal professionals and stakeholders. The architecture prioritizes a user-friendly experience, allowing users to input queries, receive predictions, and engage in a transparent dialogue with the chatbot. The interface is designed to display not only predictions but also the reasoning behind them, fostering user trust and facilitating better-informed decision-making. Comparative analysis between the user interface experiences of BERT and BERT Large involves assessing factors such as response time, interpretability of predictions, and overall usability. While BERT Large may provide more accurate predictions, the interface analysis considers whether the enhanced accuracy justifies potential delays in responses.

3.2.2 IMPLEMENTATION ENVIRONMENT

The implementation environment for an intelligent AI chatbot for legal case prediction using BERT involves a combination of hardware, software, and data resources. For hardware, a powerful computing infrastructure with GPUs is recommended to facilitate efficient training and inference processes, as BERT models can be computationally intensive. Cloud services such as AWS, Google Cloud, or Azure may be leveraged for scalability and flexibility. On the software side, frameworks like TensorFlow or PyTorch are essential for developing and fine-tuning the BERT model. The implementation may also require integration with natural language processing libraries, legal databases, and web frameworks. Additionally, a robust version control system and collaborative tools are crucial for managing the development workflow. Proper security measures must be implemented to ensure the confidentiality and integrity of legal data.

3.2.2.1 SOFTWARE REQUIREMENTS

Operating System : Windows 10 or above

Backend : Python, Flask Api

Frontend : React JS, HTML,CSS

Tools : Google Colab, Visual Studio Code

3.2.2.2 HARDWARE REQUIREMENTS

Processor : Intel i3 or above

Hard disk : Minimum 16 GB or above

Memory : Minimum 4 GB or above

GPU : Compatible with CUDA

3.2.3 DATASET DESCRIPTION

The database design for training the model includes three different datasets with respect to three different categories of legal cases which are as follows:

- Wild Animals
- Trade Secret
- Fourth Amendment

Each one of these datas has several columns which are features or questions with respect to the particular legal case category. Each of these columns has Yes or No as records. All three datasets had about 100 rows and WildAnimals dataset has 17 columns, TradeSecret dataset has 22 columns and Fourth Amendment has 46 columns.

Table 3.2.3.1 Data dictionary for WildAnimals

Variable	Description
PSport	Was the plaintiff pursuing the quarry for sport?
PGain	Did the plaintiff seek to personally gain from the quarry?
PLiving	Was the plaintiff pursuing the quarry for their livelihood?
DSport	Was the defendant pursuing the quarry for sport?
DGain	Did the defendant seek to personally gain from the quarry?
Dliving	Was the defendant pursuing the quarry for their livelihood?
Malice	Was the defendant malicious in their motive?
HotPursuit	Was the plaintiff in hot pursuit of the quarry?
NotCaught	Was the quarry not caught by the plaintiff?
LegalOwner	Was the plaintiff the legal owner of the land?
Impolite	Was the interference of the defendant in the plaintiff's pursuits impolite?
Nuisance	Did the defendant's interference with the plaintiff's pursuit amount to a nuisance?
Assault	Did an assault prevent the plaintiff from retaining possession of the quarry?
Resident	Did the quarry reside on the land?
Convention	Is the possession of the quarry governed by convention?
NoBlame	Was the defendant blameless in the interference of the plaintiff's?
Outcome	The target variable whether to find for the plaintiff or not.

Table 3.2.3.2 Data dictionary for Trade Secret

Variable	Description
BribeEmployee	Did the defendant offer the plaintiff's current or former employee an incentive to work for the defendant?
EmployeeSoleDeveloper	Was the defendant the sole developer of the product whilst employed by the plaintiff?
AgreedNotToDisclose	Had the defendant entered into a non-disclosure agreement with the plaintiff?
SecurityMeasures	Did the plaintiff take measures to ensure the security of its information?
BroughtTools	Did an employee of the plaintiff give product development information to the defendant?
CompetitiveAdvantage	Did the plaintiff's product information allow the defendant to save time or expense?
OutsiderDisclosuresRestricted	Was the plaintiff's disclosure to outsiders subject to confidential restrictions?
NoncompetitionAgreement	Had the plantiff and the defendant entered into a noncompetition agreement?
RestrictedMaterialsUsed	Did the defendant use materials that were subject to confidentiality restrictions?
UniqueProduct	Is the product of the plaintiff unique?
InfoReverseEngineerable	Could the plaintiff's product information have been learned by reverse engineering?
InfoIndependentlyGenerated	Did the defendant develop its product through independent research?
IdenticalProducts	Was the defendant's product identical to the plaintiff's?
NoSecurityMeasures	Did the plaintiff not adopt any security measures to maintain the secrecy of their information?
InfoKnownToCompetitors	Was the plaintiff's information known to competitors?
KnewInfoConfidential	Did the defendant know the plaintiff's information was confidential?
InvasiveTechniques	Did the defendant use invasive techniques to gain access to the plaintiff's information?
WaiverOfConfidentiality	Had the plaintiff entered into an agreement

	that waived confidentiality?
InfoObtainableElsewhere	Could the information have been obtained from publicly availiable sources?
InfoReverseEngineered	Did the defendant discover the plaintiff's product information through reverse engineering?
Deception	Did the defendant obtain the plaintiff's information through deception?
Outcome	The target variable whether trade secret was misappropriated.

Table 3.2.3.2 Data dictionary for Fourth Amendment

Variable	Description
car	If the vehicle that was searched was an automobile like car?
goods_container	If a large container was searched select the correct type?
mobile_home	If a movable container was searched select the correct type?
foot_locker	If by default there was no risk to lose evidence?
ExigencyWhenApproached	Was there a risk of losing evidence?
motorhome	What type of vehicle licence, if any, was held?
police_station	Did the search take place in a restricted area?
glovebox	Were the items not viewable to the public in one of the following?
car_trunk	Was the vehicle any of the following?
moving	Was the vehicle capable to move for any of the following reasons?
parked	Where was the vehicle parked, if anywhere?
highway	In what location was the vehicle?
public_informant	Did the original probable cause come from information received from public?

agent_officer	Was the original probable cause observed from?
the_public	Was the original probable cause a procedure such as?
inspection_regulation	Was the main reason to search due to?
illegal_substance	Had any of these crimes preempted the search?
searched	Have all the vehicle parts been searched?
all_parts	Were any of the following parts of the vehicle searched?
automobile_location	Where was the vehicle searched?
boot	Which, if any, of these goods were being carried?
locked	What was the protection level of the goods?
bedroom	Does the vehicle accomodation consist of?
suitable_accomodation_space	Are there any rooms in the accomodation space?
Outcome	The target variable whether warrantless search violates or not?

3.2.4 INPUT DESIGN (UI)

For an intelligent AI chatbot designed to predict legal case outcomes using BERT, the input design is crucial in ensuring accurate and relevant predictions. The first step involves gathering user input, which typically consists of case details such as facts, evidence, legal arguments, and relevant jurisdictional information. This input needs to be structured in a way that BERT can effectively process and analyze.

The UI is designed in such a way that the end user initially gets to sign up if they are a new user. Once signed up, they get directed to the login page where they can use their credentials to log in. The user then gets to choose between three different legal case types – Wild Life, Trade Secret, Fourth Amendment. Once they have chosen the type, the user should give a few details about the case. The chatbot then asks for a set of questions that the user should answer in order to get a prediction or suggestion about the case. The interactive user interface was created using React JS, HTML, and CSS to deliver a positive user experience and friendliness when interacting with the chat bot for assistance.

In our user interface, the user is asked for a few inputs such as Case Name, Defendant Name, Claim Date, Plaintiff Name, History of the Case on both sides, verdict which would be followed by a few questions about the case.

3.2.5 MODULE DESIGN

The modules involved in this project includes:

- Data Collection
- Dataset Preprocessing
- Bidirectional Encoder Representations From Transformers (BERT)
- Prediction for Given Input
- Deployment using Flask

Data Collection

The data collection module involves identifing the data sources relevant to the project which includes collecting court proceeding relating to different legal domains. For our project we acquired the data from a fellow reseacher who has collected this data from different legal proceeding documents.

Dataset Preprocessing

The data preprocessing module involves cleaning the dataset by removing irrelevant information, handling missing values, and dealing with any inconsistencies. Preprocessing the text data for BERT input, which typically includes tokenization, padding, and encoding.

Bidirectional Encoder Representations From Transformers (BERT)

The Bidirectional Encoder Representations From Transformers (BERT) which is the model training module, it involves utilizing the pre-trained BERT models for natural language understanding task and fine-tuning the pre-trained BERT model on your specific dataset if necessary, depending on the task.

Prediction for Given Input

The Prediction for Given Input module involves implementing the prediction logic using the fine-tuned BERT model or the pre-trained BERT model if fine-tuning is not required, preparing the input data in the format expected by BERT (tokenization, padding, etc.) and feeding the input data into the BERT model for prediction.

Deployment using Flask

The Deployment in Flask module involves setting up a Flask web application to serve your BERT model predictions, defining routes to handle different types of requests (e.g., GET, POST), implement the prediction logic within your Flask application, preparing the input data received from requests for BERT processing and returning the prediction results to the client.

3.2.5.1 UML DIAGRAMS

3.2.5.1.1 USE CASE DIAGRAM

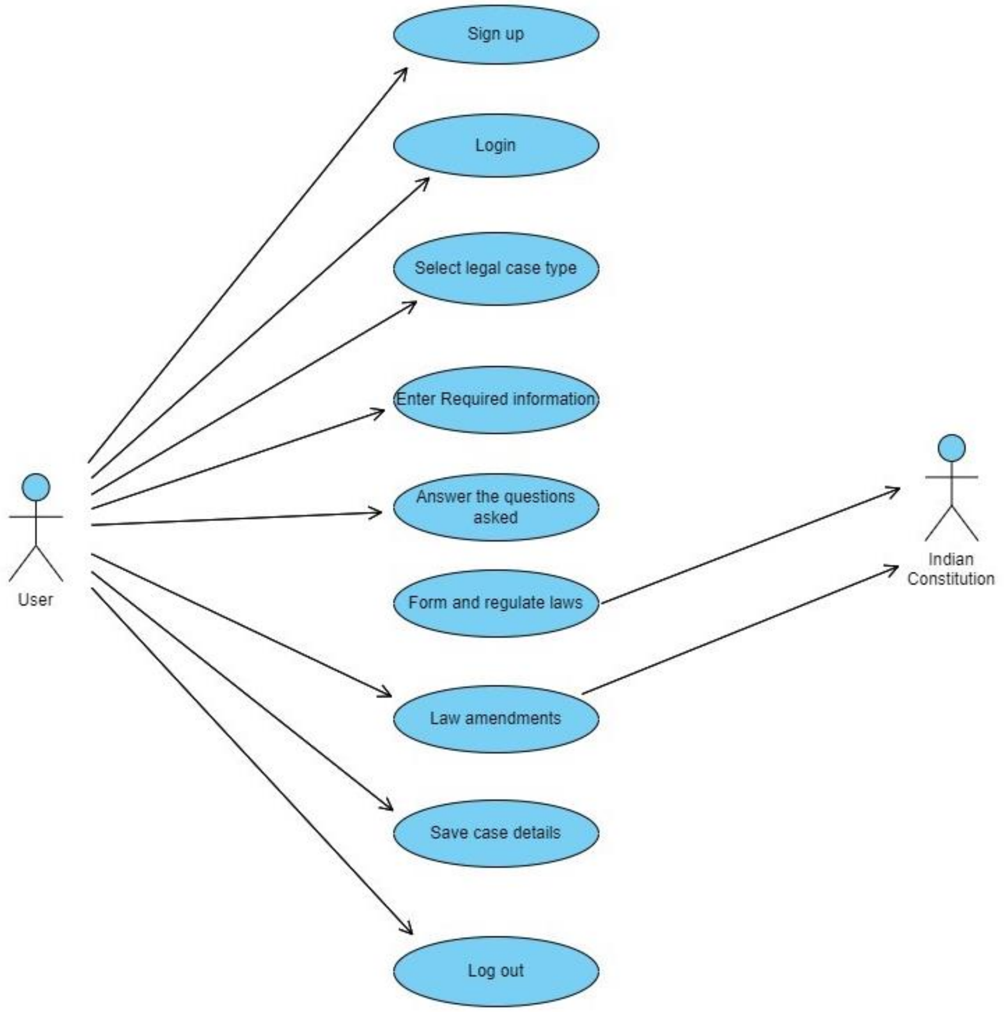

Figure 3.2.5.1.1 Use Case Diagram

A UML use case diagram is a visual representation of how various users or external systems interact with a system to achieve specified objectives or features. It captures a system's functional requirements from a user's perspective and depicts the numerous use cases, actors, and their interactions. In a use case diagram, actors are external entities such as users or other systems, and use cases are the system's specific functionalities or features. Lines linking actors and use cases represent the interactions or relationships between them.

3.2.5.1.2 DATA FLOW DIAGRAM

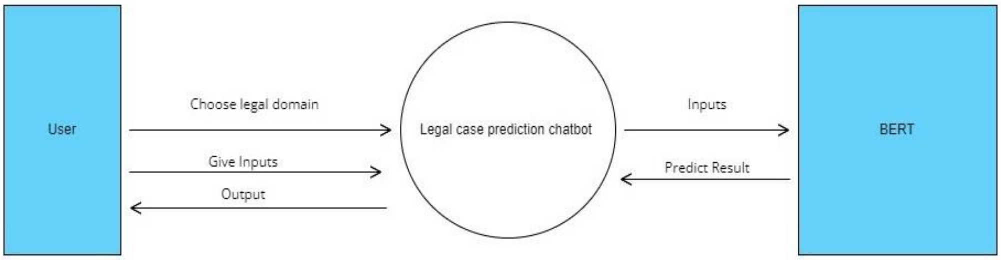

Figure 3.2.5.1.2 Data Flow Diagram – Level 0

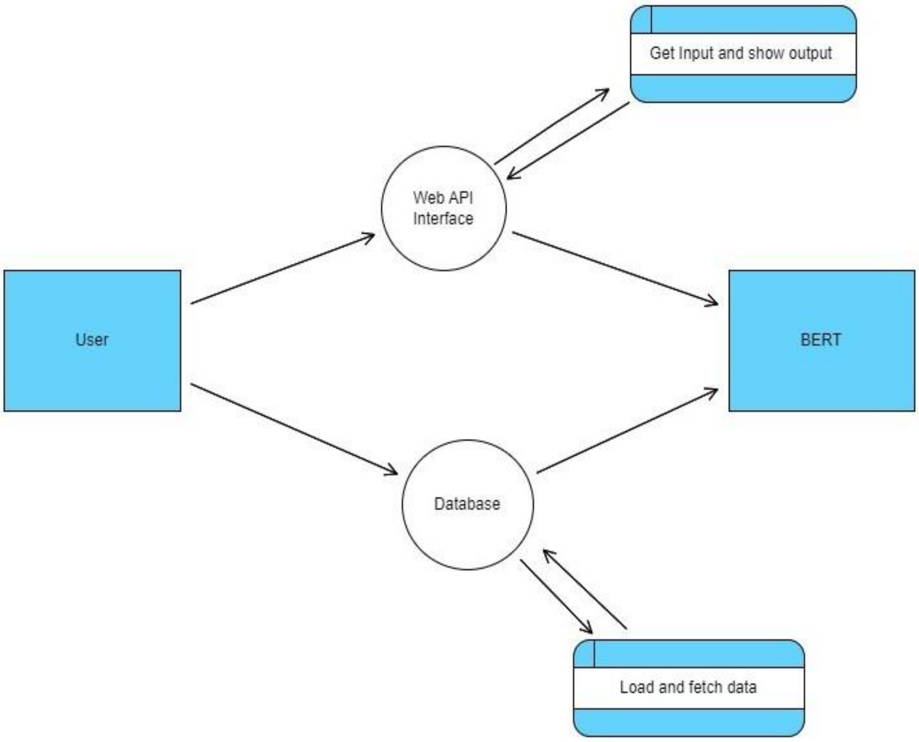

Figure 3.2.5.1.2 Data Flow Diagram – Level 1

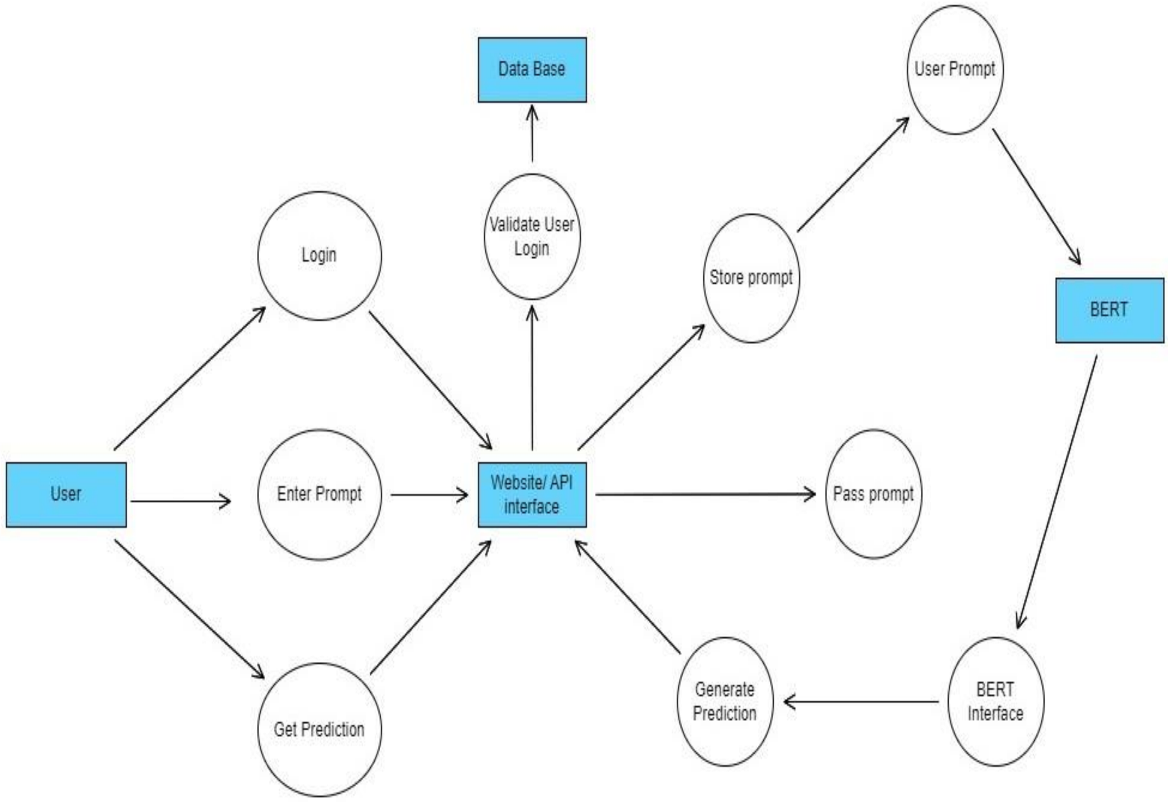

Figure 3.2.5.1.2 Data Flow Diagram – Level 2

A UML Data Flow Diagram (DFD) is a graphical representation of how data flows through a system or process. The DFD diagram typically consists of processes, data stores, data flows, and external entities. It is an effective tool for visualizing how data flows through various system components and how different processes interact. By giving a clear, high-level picture of data movement, DFDs assist analysts and stakeholders in understanding the information flow in a system and identifying potential areas for optimization or development.

3.2.5.1.3 ACTIVITY DIAGRAM

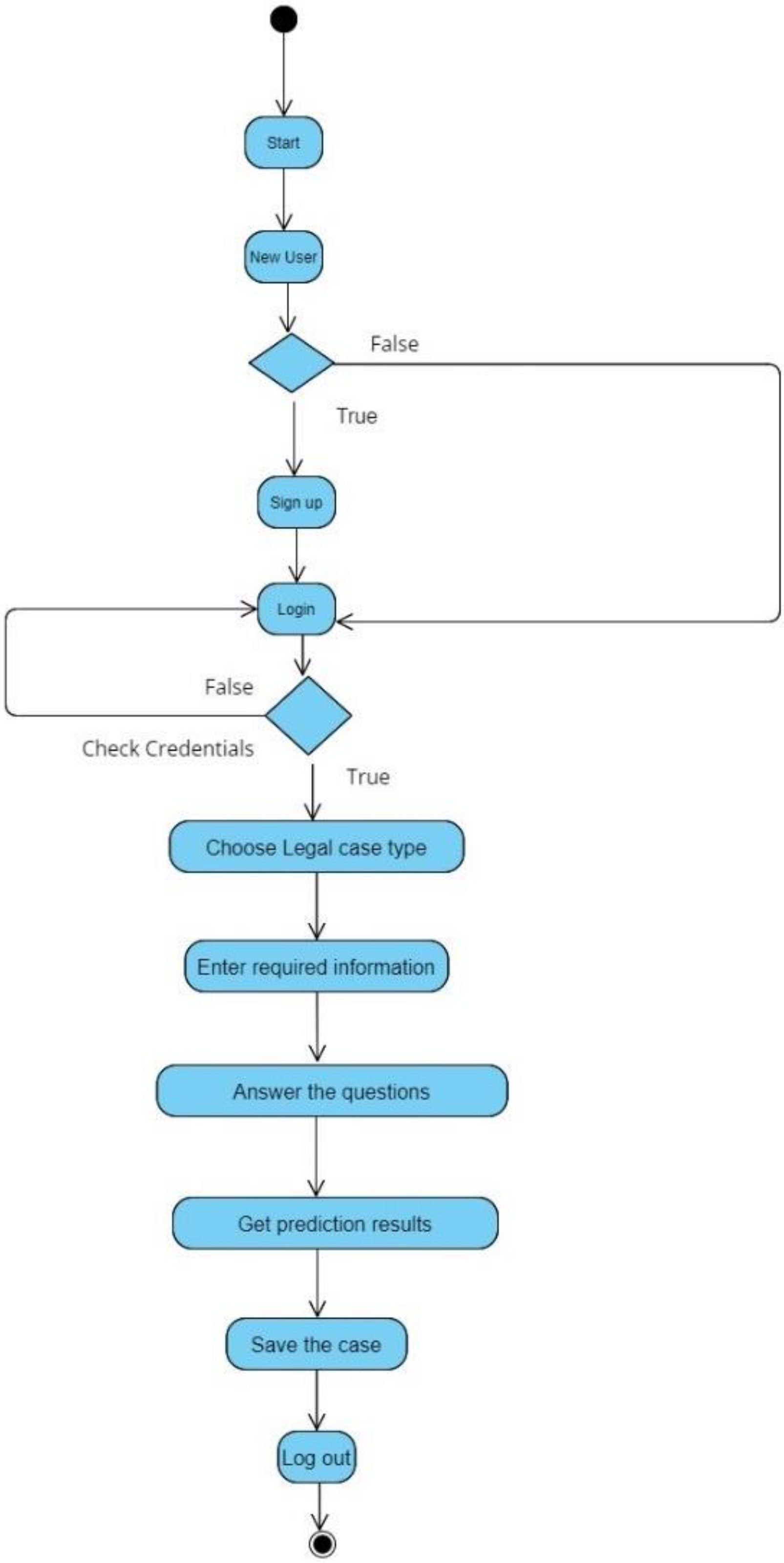

Figure 3.2.5.1.3 Activity Diagram

An activity diagram is a type of UML diagram that illustrates the dynamic characteristics of a system or process by displaying activities, actions, and the flow of control between them. It offers a high-level picture of how different system components interact and work together to achieve a certain goal or business activity. Activities are shown as rounded rectangles, and transitions between activities are indicated by arrows that indicate the direction of control flow.

CHAPTER 4
SYSTEM IMPLEMENTATION

4. SYSTEM IMPLEMENTATION

4.1 DATA COLLECTION

The data collection process for an intelligent AI chatbot for legal case prediction is a crucial phase, as the quality and diversity of the dataset directly impact the model's effectiveness. To build a robust and accurate prediction model, a multi-faceted approach to data collection is essential. Firstly, legal text corpora must be gathered to train the AI model effectively. This includes legal documents, court opinions, statutes, and case summaries.. It's imperative to cover diverse legal domains, ensuring the model's ability to handle a wide range of cases.

Additionally, historical case data, including details on case outcomes, legal arguments, and relevant contextual information, is instrumental for the training process. This dataset aids the chatbot in learning patterns and correlations between case characteristics and their resolutions. Careful attention should be paid to anonymize sensitive information, adhering to privacy and ethical considerations. For our project we acquired the data from a fellow reseacher who has collected this data from different legal proceeding documents and is performing the research on the same data but on a different objective [18] and we reformed it according to our requirements.

4.2 DATASET PREPROCESSING

The data collected needs to be further preprocessed before passing it on the model. Data preprocessing is an important stage in the machine learning pipeline that involves cleaning and transforming raw data so that it may be used to train and evaluate models. This procedure has a substantial impact on the performance and effectiveness of machine learning algorithms. The first part of data preparation is dealing with missing or erroneously entered values. This issue is addressed by techniques such as imputation or the elimination of instances with missing values, which ensure that the data used for training is complete and accurate. Feature engineering is often required to convert raw data into a format that machine learning models can comprehend. This

includes scaling numerical features to a common range, encoding categorical variables as numerical representations, and developing new features to collect relevant data.
The data preprocessing carried out in this project includes:

- Handling the missing values.
- Label encoding target variable.
- Feature Concatenation of Independent Variables

4.2.1 Handling the Missing Values

Missing values can have an adverse effect on model performance, and properly treating them is critical for accurate predictions. Several strategies are used to manage missing data, including imputation and elimination. Imputation is the process of filling in missing numbers with approximated or calculated values based on existing data. Mean or median imputation is a common method for numerical characteristics, while mode imputation is used for categorical features. More advanced techniques, such as regression or machine learning-based imputation, can be used to improve accuracy. Alternatively, removal entails eliminating occurrences with missing values from the dataset. While straightforward, this method is only appropriate when the number of missing values is low and eliminating instances has no substantial influence on the total dataset. We had no missing values in our data hence there was no neccessity to treat them.

4.2.2 Label Encoding Target Variable

Label encoding of the target variable is a preprocessing technique used frequently in machine learning when dealing with categorical target variables. This procedure converts categorical labels into numerical representations, enabling machine learning algorithms more efficient. Each unique category is allocated a unique number, converting the target variable into a format that algorithms can easily grasp. Label encoding simplifies the modelling process by translating categorical labels into numerical values, making them compatible with algorithms that require numerical input. However, it is important to note that this encoding assumes an ordinal relationship between the encoded values, which may not necessarily be correct. For this project we have encoded our target variables as 0,1, and 2.

4.2.3 Feature Concatenation of Independent Variables

Concatenating independent variables before introducing them to a BERT model is typical in natural language processing instances where the input data contains text and non-text elements. It is common for BERT models to need a single input format, so concatenation lets you combine many feature types into a single input sequence. This unified format simplifies the model's input representation while also providing a uniform framework for processing. BERT, which was built for natural language processing applications, processes text sequences by default. When working with datasets with a variety of attributes, such as numerical or categorical data, concatenation ensures a consistent input structure.

4.3 BIDIRECTIONAL ENCODER REPRESENTATIONS FROM TRANSFORMERS (BERT)

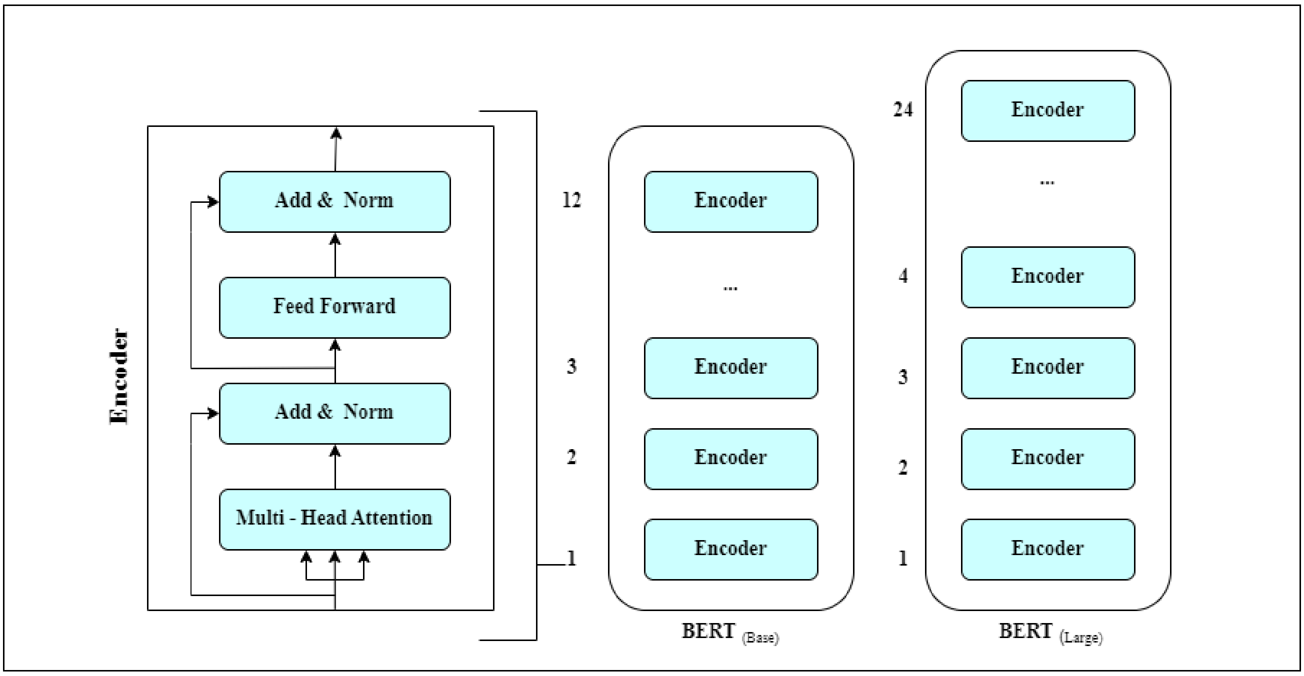

Figure 4.3.1 BERT Architecture

After data pre-processing, it will be fed for training with the machine learning algorithm such as BERT.

BERT is an open-source machine learning framework for natural language processing (NLP). BERT is designed to help computers understand the meaning of ambiguous language in text by using surrounding text to establish context. The BERT framework was pre-trained using text from Wikipedia and can be fine-tuned with question-and-answer datasets. BERT, which stands for Bidirectional Encoder Representations from Transformers, is based on Transformers, a deep learning model in which every output element is connected to every input element, and the weightings between them are dynamically calculated based upon their connection. (In NLP, this process is called attention.)

Historically, language models could only read text input sequentially -- either left-to-right or right-to-left -- but couldn't do both at the same time. BERT is different because it is designed to read in both directions at once. This capability, enabled by the introduction of Transformers, is known as bidirectionality. Using this bidirectional capability, BERT is pre-trained on two different, but related, NLP tasks: Masked Language Modelling and Next Sentence Prediction. The objective of Masked Language Model (MLM) training is to hide a word in a sentence and then have the program predict what word has been hidden (masked) based on the hidden word's context. The objective of Next Sentence Prediction training is to have the program predict whether two given sentences have a logical, sequential connection or whether their relationship is simply random.

The goal of any given NLP technique is to understand human language as it is spoken naturally. In BERT's case, this typically means predicting a word in a blank. To do this, models typically need to train using a large repository of specialized, labelled training data. This necessitates laborious manual data labelling by teams of linguists.
BERT, however, was pre-trained using only an unlabelled, plain text corpus (namely

the entirety of the English Wikipedia, and the Brown Corpus). It continues to learn unsupervised from the unlabelled text and improve even as its being used in practical applications (ie Google search). Its pre-training serves as a base layer of "knowledge" to build from. From there, BERT can adapt to the ever-growing body of searchable content and queries and be fine-tuned to a user's specifications. This process is known as transfer learning.

Bidirectional Encoder Representations from Transformers (BERT) is a pioneering innovation in natural language processing (NLP), transforming how computers interpret and generate human language. BERT was first introduced in a landmark research article by Google researchers in 2018, and it has since been a cornerstone in different NLP applications, setting new benchmarks in tasks like as text classification, named entity recognition, and question answering. BERT is based on the transformer architecture, a neural network architecture presented in the 2017 publication "Attention is All You Need" by Vaswani et al. The transformer design employs a feature known as self-attention, which allows the model to analyze input data in parallel, making it extremely efficient for handling sequential data such as natural language.

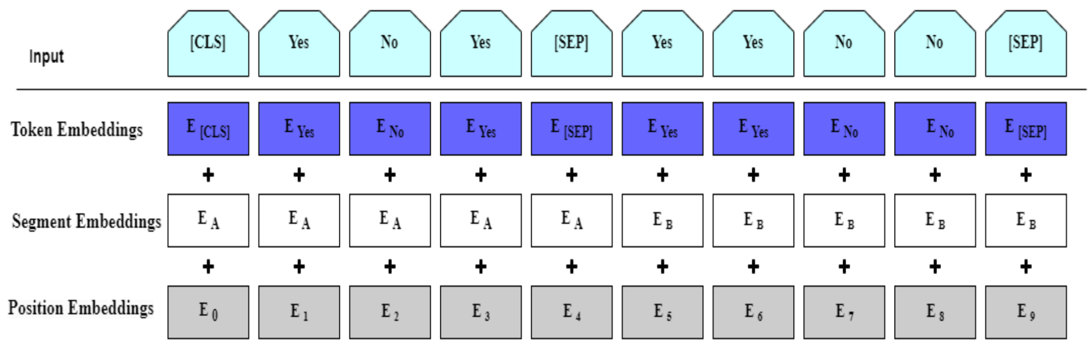

Figure 4.3.2 Training Inputs

BERT's defining feature is its bidirectional training method. Unlike standard models that analyse text in a unidirectional manner (either left-to-right or right-to-left), BERT reads a word's whole context, taking into account both left and right context in all levels of the neural network. This bidirectional training enables BERT to catch complex linguistic links and nuances, hence considerably enhancing its contextual awareness. BERT's pre-training involves a masked language model (MLM) objective. During pre-training, random words in a phrase are masked, and the model is taught to predict the masked words using the context provided by the surrounding words. This approach generates contextualized word embeddings, which means that the representation of a word is determined by its context inside a sentence.

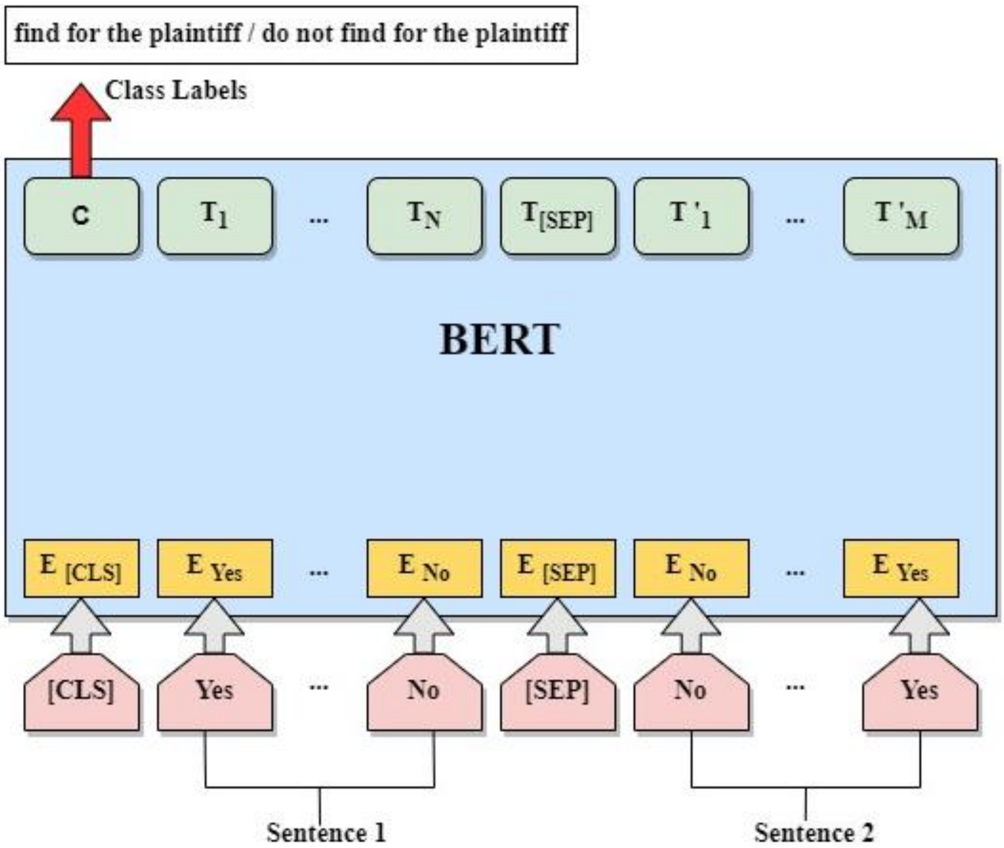

Figure 4.3.3 Single Sentence Classification

Pretraining and fine-tuning are crucial steps in leveraging BERT (Bidirectional Encoder Representations from Transformers) for specific natural language processing (NLP) tasks. The BERT architecture is pretrained on a large corpus of text data and then fine-tuned on a task-specific dataset. Let's delve into each phase:

1. Pretraining:

- In the pretraining phase, BERT is pretrained on a massive amount of unlabeled text data using two unsupervised learning objectives:

- Masked Language Model (MLM): BERT randomly masks some words in the input text, and the model is trained to predict the masked words based on the context provided by the surrounding words.
- Next Sentence Prediction (NSP): BERT is trained to predict whether a randomly selected sentence follows another sentence in a given text pair. This encourages the model to understand the contextual relationships between sentences.
- After pretraining, BERT learns contextualized word representations that capture rich semantic information and contextual nuances.

The MLM objective can be expressed using the negative log likelihood (cross-entropy) loss. For each masked position *i*, the loss is calculated as follows:

$$\mathcal{L}_{MLM} = -\sum_{i} (\log P(\mathcal{Y}_i | x_1, x_2, \dots, x_n)) \quad (1)$$

where:

- $x_1, x_2, \dots, x_n$ are input tokens
- $\mathcal{Y}_i$ is the ground truth token for the masked position *i*.
- $P(\mathcal{Y}_i | x_1, x_2, \dots, x_n)$ is the predicted probability for the correct token.

The Next Sentence Prediction objective can be formulated using the binary cross-entropy loss as follows:

$$\mathcal{L}_{NSP} = -(IsNext(S_1, S_2). \log P\,(IsNext = 1 |\, S_1, S_2) + \left(1 - IsNext(S_1, S_2)\right). \log P(IsNext = 0 |\, S_1, S_2)) \quad (2)$$

where:

- $IsNext(S_1, S_2)$ is a binary indicator (1 if S_2 is the next sentence after S_1, 0 $otherwise$)
- $\log P\,(IsNext = 1 |\, S_1, S_2)$ is the predicted probability for the sentences being consecutive.
- $\log P\,(IsNext = 0 |\, S_1, S_2)$ is the predicted probability for the sentences not being consecutive.

Overall, the BERT training objective can be given as below:

$$\mathcal{L}_{BERT} = \mathcal{L}_{MLM} + \lambda . \mathcal{L}_{NSP} \tag{3}$$

where:

- $\mathcal{L}_{BERT}$ is the combined loss.
- λ is a hyperparameter that controls the trade-off between the MLM and NSP objectives.

2. Fine-Tuning:

- While pretrained BERT is capable of understanding general language context, fine-tuning is necessary to adapt it to specific NLP tasks (e.g., sentiment analysis, named entity recognition, question answering).
- Fine-tuning requires a task-specific labeled dataset. The dataset includes examples related to the target task, each annotated with the correct output.
- The pretrained BERT model is initialized with learned weights.
- Additional task-specific layers (e.g., classification layers) are added to the pretrained BERT architecture.
- The entire model is then trained on the task-specific dataset, and the weights are updated through backpropagation.
- The fine-tuned BERT model generalizes its understanding from the pretrained knowledge to the specific task, capturing task-specific features and patterns.

BERT is pre-trained on massive volumes of unlabeled text data, so it can learn general language patterns and structures. The pre-training phase entails exposing the model to a wide range of datasets, which will provide it with a broad understanding of language. The model learns to predict missing words in sentences, resulting in detailed contextual embeddings that capture semantic relationships in the text. BERT's bidirectional training and contextualised embeddings make it extremely effective in a wide range of downstream NLP applications. When fine-tuned on task-specific labelled datasets, BERT applies its pre-trained knowledge to perform remarkably well on tasks such as sentiment analysis, text classification, and named entity identification. BERT's versatility and transfer learning capabilities make it an effective alternative for a wide range of language understanding applications.

4.4 PREDICTION FOR GIVEN INPUT

The primary objective of this research is to identify the optimal prediction model, specifically the most effective Machine Learning techniques, for classifying the type of legal cases within the case type chosen. By employing various machine learning algorithms, the goal is to provide comprehensive answers during the prediction process, enabling the straightforward determination of the legal case type. Consequently, this project aims to enhance the predictive capabilities for lawyers, assisting them in efficiently identifying the category of legal cases through the integration of a chatbot application and cutting-edge machine learning technologies.

4.5 DEPLOYMENT USING FLASK

Flask is a Python-based web framework that is both lightweight and adaptable. Flask is well-known for its simplicity and versatility, giving developers the tools they need to construct web-based applications fast and efficiently. Flask's basic core enables developers to select and integrate components as needed, making it adaptable to a variety of project requirements. It adheres to the WSGI (Web Server Gateway Interface) standard, making it interoperable with various web servers.

Flask has a "micro" design philosophy, providing the necessary components for web development while allowing developers to use extra libraries for specific capabilities. Despite its basic design, Flask has powerful functionality like URL routing, request handling, and template rendering. It has extensions for integrating features such as authentication, database connectivity, and RESTful APIs.

Additionally, Flask promotes the use of RESTful principles for designing web services. It provides tools to handle HTTP methods, making it suitable for building RESTful APIs. Flask-WTF, an extension for handling forms, simplifies the process of collecting and validating user input, enhancing the development of interactive web applications.While Flask is often chosen for its simplicity, it can be extended to handle more complex applications through the integration of various extensions and middleware. Its open-source nature and active community contribute to its ongoing development and popularity, making Flask a versatile and powerful tool for web developers building applications across a range of scales and complexities.

CHAPTER 5
RESULTS & DISCUSSION

5. RESULTS & DISCUSSION

The performance metrices used to evaluate and validate the proposed model includes:

- Precision
- Recall
- F1 Score
- Learning Curve
- Accuracy

5.1 PRECISION

A performance measure called precision is employed in machine learning to assess how precisely the classification model's positive predictions are achieved. A model with a high precision value is one that makes fewer errors when predicting positive outcomes and has fewer false positive predictions. The precision can be measured using the following formula:

$$\textbf{Precision} = \frac{\textbf{True Positive}}{\textbf{True Positive} + \textbf{True Negative}} \qquad (4)$$

Table 5.1.1 Precision scores for the different legal case categories.

Legal Case Category	$\textbf{BERT}_{\textbf{Base}}$	$\textbf{BERT}_{\textbf{Large}}$
Wild Animals	0.75	0.60
Trade Secret	0.95	0.95
Fourth Amendment	0.86	0.67

5.2 RECALL

Along with precision, recall is a performance indicator that quantifies the percentage of positive occurrences that the model correctly recognized as positive. A model with a high recall value will have few incorrect negative predictions and be able to correctly identify the majority of positive events. The recall for a model can be calculated as given below:

$$\textbf{Recall} = \frac{\textbf{True Positive}}{\textbf{True Positive} + \textbf{False Negative}} \tag{5}$$

Table 5.2.1 Recall scores for the different legal case categories.

Legal Case Category	**BERT$_{Base}$**	**BERT$_{Large}$**
Wild Animals	1.00	1.00
Trade Secret	0.95	1.00
Fourth Amendment	1.00	1.00

5.3 F1 SCORE

It is prevalently used in models to classify data into two groups for assessing the algorithms's efficacy. It is referred to as the harmonic mean of recall and precision. It normally ranges from 0 to 1, with 1 representing the highest possible f1 score. The formula to calculate the f1 score is as follows:

$$\textbf{F1 Score} = \frac{\textbf{2 x Precision x Recall}}{\textbf{Precision} + \textbf{Recall}} \tag{6}$$

Table 5.3.1 F1 scores for the different legal case categories.

Legal Case Category	**BERT$_{Base}$**	**BERT$_{Large}$**
Wild Animals	0.86	0.75
Trade Secret	0.95	0.97
Fourth Amendment	0.92	0.80

5.4 LEARNING CURVE

A learning curve is a graphical representation that illustrates how a model's performance changes over time, typically across training epochs or iterations. Learning curves are essential tools for monitoring and analyzing the training process of a machine learning or deep learning model. They provide insights into the model's convergence, performance on training and validation data, and potential issues like overfitting or underfitting.

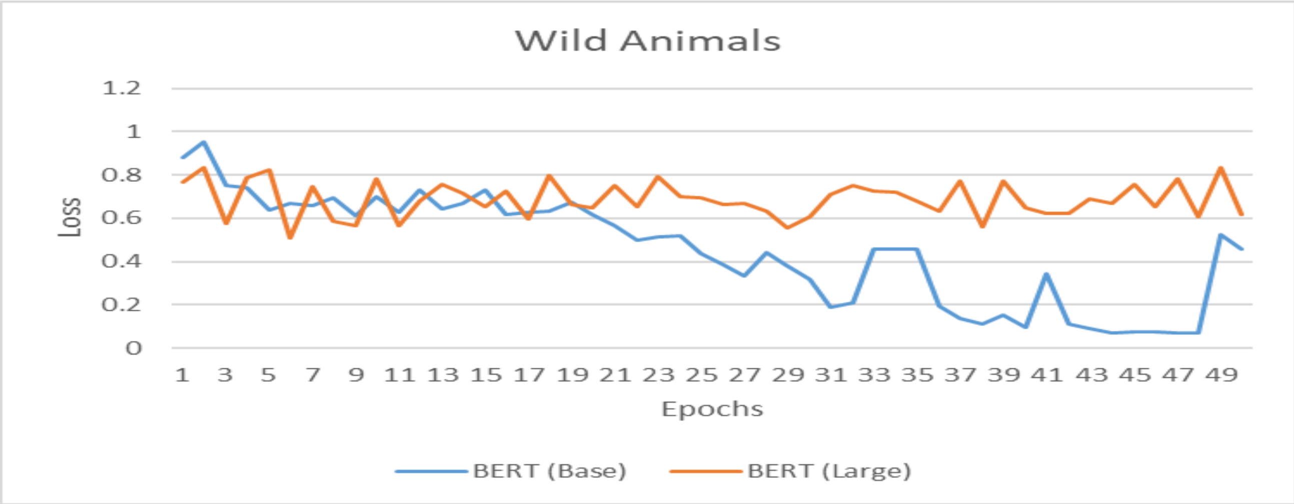

Figure 5.4.1 Graph for loss curve of Wild Animals

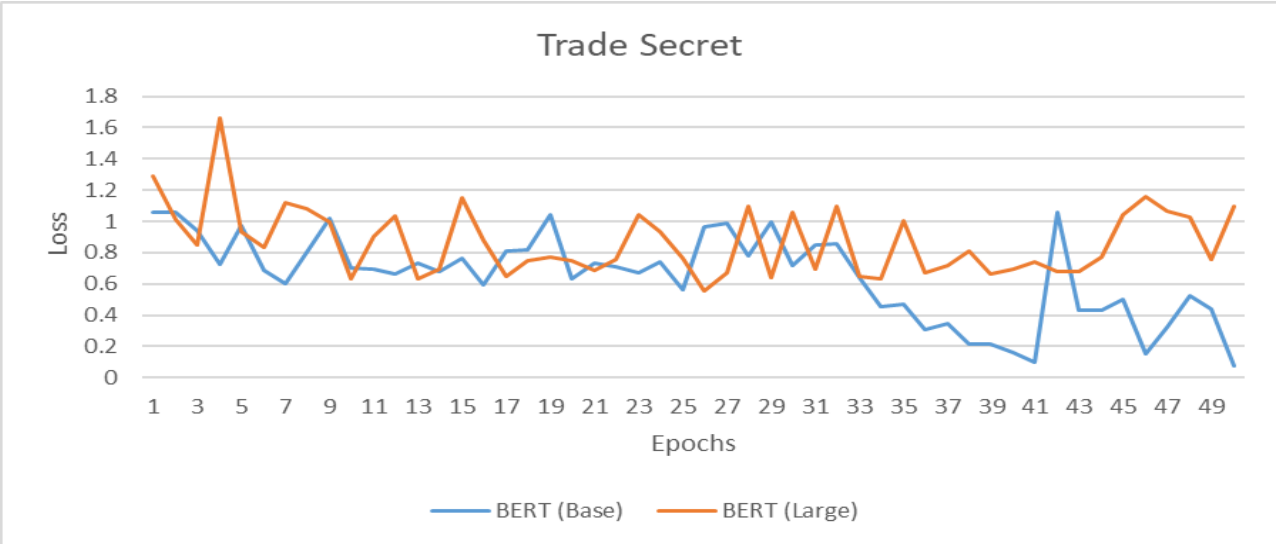

Figure 5.4.2 Graph for loss curve of Trade Secret dataset

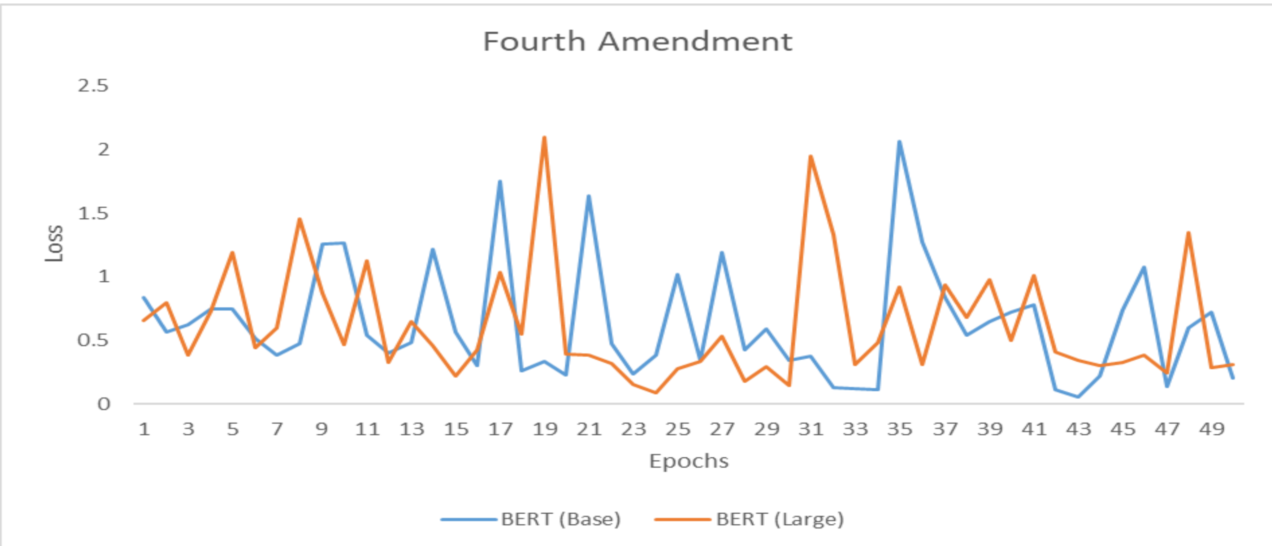

Figure 5.4.3 Graph for loss curve of Fourth Amendment dataset

The graph starts of with a high error value since the model has not seen the data. As the training progresses, the error values gradually decrease reflecting improved performance. The ideal decrease in the loss value is due to the fact that the model learnd the patterns from the training data. It can be clearly seen from the graphs that for Wild Animals dataset eventhough the model starts with a high error it decreases to about 0.5 at the end of the training, as for trade secrets and fourth amendments the error reaches almost 0 at the end. For all three datasets it can be clearly noticed from the graphs that the error value reaches the least for BERT(base) model suggesting that it performs better than the BERT(large) model in this context.

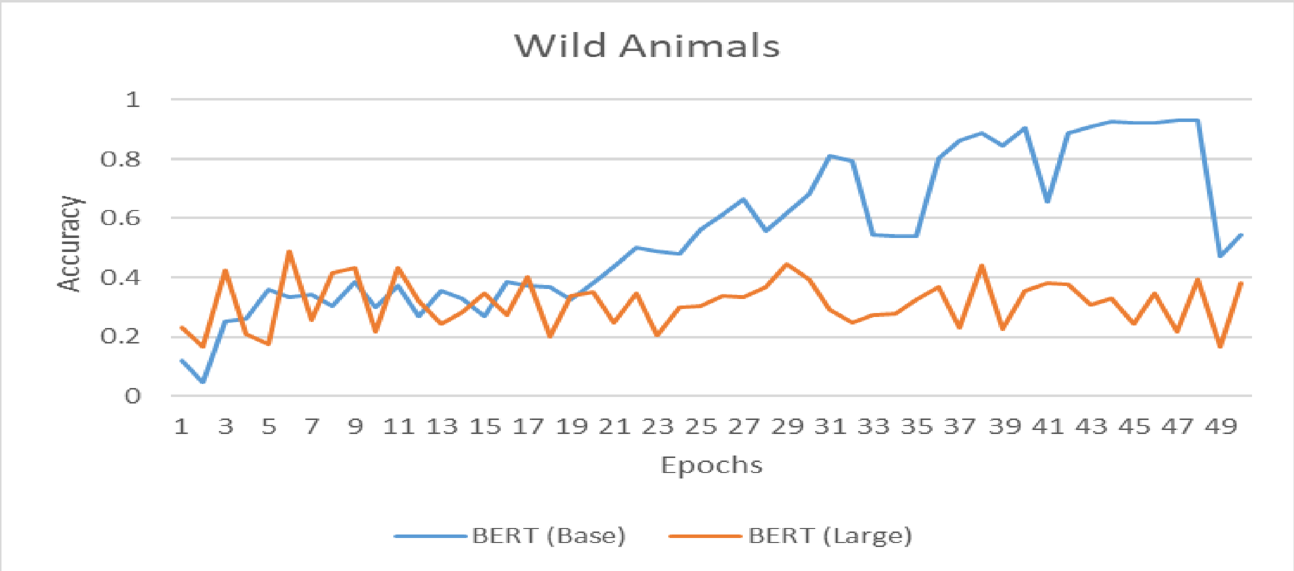

Figure 5.4.4 Graph for accuracy curve of Wild Animals dataset

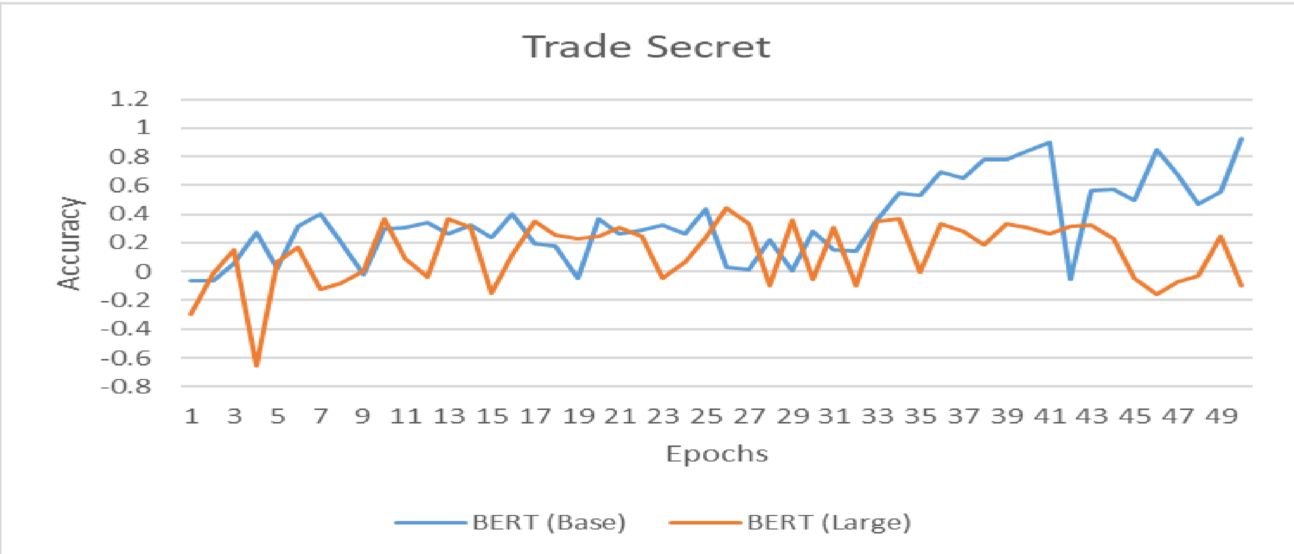

Figure 5.4.5 Graph for accuracy curve of Trade Secret dataset

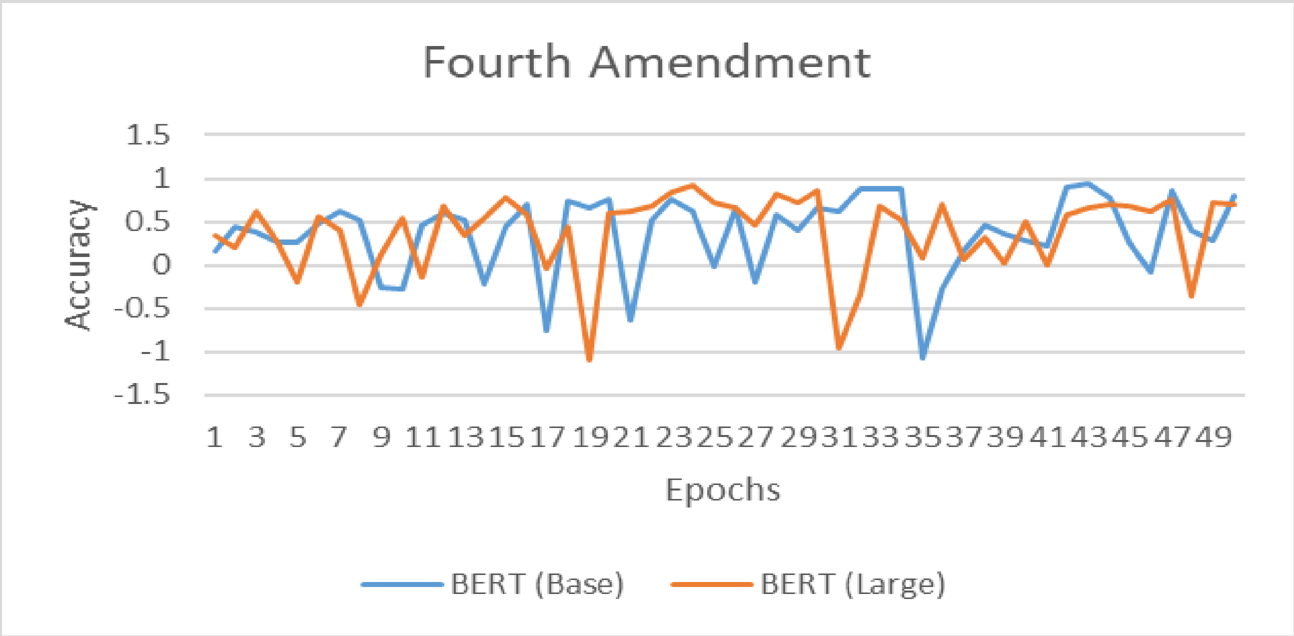

Figure 5.4.6 Graph for accuracy curve of Fourth Amendment dataset

As for the accuracy curve the graph starts of with a lower accuracy value since the model has not seen the data. As the training progresses, the accuracy values gradually increases reflecting improved performance. It can be clearly seen from the graphs that for Wild Animals dataset eventhough the model starts with a lower accuracy value it increases to about 0.5 at the end of the training, as for trade secrets and fourth amendments the accuracy reaches almost 0.7 at the end. For all three datasets it can be clearly noticed from the graphs that the accuracy reaches the highest for BERT(base) model suggesting that it performs better than the BERT(large) model with respect to accuracy.

5.5 ACCURACY

Accuracy In machine learning, accuracy is a frequently used performance metric to assess the accuracy of a model's predictions. It is the measure of the ratio of the number of correct predictions made by the model to the total number of predictions. It is the metric that best captures the performance of the model when the independent variable has a balanced distribution. The formula to calculate accuracy is as follows:

(7)

$$\textbf{Accuracy} = \frac{\textbf{Number of correct predictions}}{\textbf{Total number of predictions}}$$

Table 5.5.1 Accuracy for the different legal case categories.

Legal Case Category	**$\text{BERT}_{\text{Base}}$**	**$\text{BERT}_{\text{Large}}$**
Wild Animals	80.00	60.00
Trade Secret	90.62	84.37
Fourth Amendment	88.88	66.66

Eventhough we started with the BERT(large) model which has 24 encoders anticipating that it would perform better than the BERT(base) model with 12 encoders but it was the other way around. While the previous authors work was only able to give about 70% accuracy with model such as SVM [10], our model was able to perform well with an accuracy of 80% and above.

In the work done by Joel Niklaus et al. [8], they have worked on BERT Standard, BERT Large and Hierarchical BERT where the BERT large and hierarchical BERT performed well for their Swiss Federal Supreme Court dataset. While the BERT standard didn’t perform well than the other two it still was better than the linear models.

CHAPTER 5
CONCLUSION & FUTURE WORK

6. CONCLUSION

In conclusion, the implementation of an intelligent AI chatbot for legal case prediction based on the BERT model constitutes a significant step forward in the integration of artificial intelligence and legal services. The incorporation of BERT, with its bidirectional training and contextualised word embeddings, gives the chatbot an unprecedented ability to comprehend the nuances of legal language and anticipate case outcomes with complexity and accuracy that was before unreachable. By delivering a user-friendly interface, the chatbot extends its potential to legal practitioners, researchers, and the general public, enabling legal case analysis more accessible and efficient. Alongside predictions, the chatbot functions as an exhaustive legal research assistant, offering applicable precedents, statutes, and case law, decreasing research time and helping to enhance legal practice.

FUTURE WORK

The proper application of AI in the realm of law is more than merely a goal; continual scrutiny and enhancing of the chatbot's operations are required to maintain the highest standards of legal practice. Enhancing an intelligent AI chatbot for legal case prediction using a BERT model necessitates ongoing refinement and adaptability to meet evolving demands.

- Introduction multi-modal capacities, allowing users for entering information via text, speech, or pictures.
- Further expantion of the categories of legal cases so that the users get assisted on all major legal case types.

APPENDICES

A.1 SDG GOALS

PEACE, JUSTICE, AND STRONG INSTITUTIONS

Our project is centered on Sustainable Development Goal 16 – Peace, Justice, and Strong Instituitions. It addresses the crucial need to foster peaceful and inclusive societies, ensure equal access to justice for everyone, and establish strong, accountable institutions. As one of the 17 goals defined by the United Nations as part of the 2030 Agenda for Sustainable Development, SDG 16 acknowledges the interconnectivity of peace, justice, and effective governance in achieving global sustainable development. SDG 16 aspires to minimise violence, strengthen the rule of law, and develop transparent and accountable institutions capable of efficiently meeting community needs. Peace is fundamental for the prosperity of nations, and SDG 16 recognizes this by calling for the reduction of all forms of violence, including armed conflicts, domestic violence, and abuse. Access to justice is a key component of SDG 16, emphasizing the importance of providing legal identity, ensuring equal access to legal resources, and promoting the rule of law. Justice systems that are fair, transparent, and accessible contribute to the resolution of disputes, the protection of human rights, and the establishment of a foundation for peaceful coexistence.Strong institutions form the third pillar of SDG 16, focusing on building accountable, transparent, and effective governance structures. This involves promoting inclusive decision-making processes, combating corruption, and ensuring that institutions at all levels are responsive to the needs of the people they serve. The goal emphasizes the importance of good governance as a driver of sustainable development, recognizing that institutions with integrity are essential for fostering trust within societies.

A.2 SAMPLE DATASET

Source for dataset: https://aclanthology.org/2021.acl-long.313

	PSport	PGain	PLiving	DSport	DGain	Dliving	Malice	HotPursuit	NotCaught	LegalOwner	Impolite	Nuisance	Assault	Resident	Convention	NoBlame	outcome
0	No	No	Yes	Yes	No	No	Yes	No	Yes	Yes	No	Yes	No	No	No	No	find for the plaintiff
1	Yes	No	No	No	No	No	No	Yes	Yes	No	Yes	No	No	No	No	No	do not find for the plaintiff
2	No	No	Yes	No	No	Yes	No	Yes	Yes	No	Yes	No	No	No	No	No	do not find for the plaintiff
3	No	No	Yes	No	No	Yes	No	No	Yes	No	No	No	No	No	Yes	Yes	find for the plaintiff
4	No	No	No	No	Yes	No	No	Yes	Yes	No	No	No	Yes	No	No	Yes	do not find for the plaintiff

Figure A.2.1 Sample of the dataset – Wild Animals

	BribeEmployee	EmployeeSoleDeveloper	AgreedNotToDisclose	SecurityMeasures	BroughtTools	CompetitiveAdvantage	OutsiderDisclosuresRestricted	NoncompetitionAgreement
0	No	No	No	No	No	No	No	No
1	No	No	Yes	Yes	No	No	Yes	No
2	No	No	Yes	Yes	No	No	No	No
3	No	No	No	No	No	No	No	No
4	No	No	Yes	Yes	No	No	No	No

Figure A.2.2 Sample of the dataset – Trade Secret

	car	mobile_home	foot_locker	goods_container	paper_bag	suitcase	ExigencyWhenApproached	near_court	motorhome	motorhome.1	...	garage	money	closed	locked	double_locked
0	Yes	No	No	No	No	No	No	No	No	No	...	No	No	No	No	No
1	Yes	No	No	No	No	No	No	No	No	No	...	No	No	No	No	No
2	Yes	No	No	No	No	No	No	No	No	No	...	Yes	No	No	No	No
3	Yes	No	No	No	Yes	No	No	No	No	No	...	No	No	No	No	Yes

Figure A.2.3 Sample of the dataset – Fourth Amendment

A.3 SOURCE CODE

BACKEND

1. Importing necessary packages

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import tensorflow as tf
from sklearn.preprocessing import LabelEncoder
from keras.utils import to_categorical
import logging
import torch
from sklearn.metrics import accuracy_score
import pickle
```

2. Reading the datatsets

```
df_WildAnimals =
pd.DataFrame(pd.read_excel("/content/gdrive/MyDrive/legal_chat_project/Legal_chat_dataset.xlsx",sheet_name='WildAnimals'))

df_TradeSecret =
pd.DataFrame(pd.read_excel("/content/gdrive/MyDrive/legal_chat_project/Legal_chat_dataset.xlsx",sheet_name='TradeSecret'))

df_FourthAmendment =
pd.DataFrame(pd.read_excel("/content/gdrive/MyDrive/legal_chat_project/Legal_chat_dataset.xlsx",sheet_name='FourthAmendment'))
```

3. Label encoding target varibles

```
df_WildAnimals['outcome']=df_WildAnimals["outcome"].replace("find for the plaintiff",1)
df_WildAnimals['outcome']=df_WildAnimals["outcome"].replace("do not find for the plaintiff",0)

df_TradeSecret['outcome']=df_TradeSecret["outcome"].replace("a trade secret was misappropriated, find for plaintiff", 1)
df_TradeSecret['outcome']=df_TradeSecret["outcome"].replace("no trade secret was misappropriated,find for plaintiff", 0)
df_TradeSecret['outcome']=df_TradeSecret["outcome"].replace("no trade secret was misappropriated, find for defendant", 2)

df_FourthAmendment['outcome'] =
df_FourthAmendment["outcome"].replace("warantless search did not violate the fourth amendment", 1)
df_FourthAmendment['outcome'] =
df_FourthAmendment["outcome"].replace("warantless search violates the fourth amendment", 0)
```

3. Feature concatenation of the independent variables

```
value = []
for data in  df_WildAnimals.iloc[:,:-1].values:
    value.append(" ".join(data))
print(value)

df_value = pd.DataFrame(value)
df_values = pd.concat([df_value,df_WildAnimals.iloc[:,-1]],axis = 1)
df_values.rename(columns = {0:'text',"outcome":"labels"}, inplace = True)
```

```
value_data = []
for data in  df_TradeSecret.iloc[:,:-1].values:
    value_data.append(" ".join(data))
print(value_data)
df_value = pd.DataFrame(value_data)
df_value = pd.concat([df_value,df_TradeSecret.iloc[:,-1]],axis = 1)
df_value.rename(columns = {0:'text',"outcome":"labels"}, inplace = True)

data = []
for value in  df_FourthAmendment.iloc[:,:-1].values:
    data.append(" ".join(value))
print(data)
df_data = pd.DataFrame(data)
df_data = pd.concat([df_data,df_FourthAmendment.iloc[:,-1]],axis = 1)
df_data.rename(columns = {0:'text',"outcome":"labels"}, inplace = True)
df_data.head()
```

4.Implementing the BERT model

```
model_args = ClassificationArgs(num_train_epochs=50,
                                no_save=True,
                                no_cache=True,
                                overwrite_output_dir=True)
model = ClassificationModel("bert",
                            'bert-base-cased',
                            args = model_args,
                            num_labels=3,
                            use_cuda=cuda_available)
model.train_model(df_data[["text","labels"]])
```

5. Deploying using Flask

```
from flask import Flask
from flask import Flask, app, request
import json

from base64 import b64decode, b64encode
from flask_cors import CORS, cross_origin
import requests
app = Flask(__name__)
cors = CORS(app)
# run_with_ngrok(app)   #starts ngrok when the app is run
ngrok_url = "https://96f7-34-125-16-210.ngrok-free.app/"
@app.route('/legal', methods=['GET','POST'])
def a():
    data,data1 = request.json['uri'],request.json['uri1']
    print(data)
    r = requests.post(ngrok_url+"/legal", data={'uri': data,"uri1":data1})
    return r.json()
    # return {'data': True}
@app.route('/login', methods=['POST'])
def log_in():
        firstname = request.json['firstname']
        password = request.json['password']
        r = requests.post(ngrok_url+"/login", data={'firstname':
firstname,"password":password})
        return r.json()
@app.route('/case', methods=['POST', 'GET'])
def caseDetails():
        if request.method == 'POST':
```

```
            casedetail = request.json['casedetail']
            print(casedetail)
            r = requests.post(ngrok_url+"/case",json=casedetail,
headers={"ContentType":"application/json"})
            print(r)
            return r.json()
        else:
            r = requests.get(ngrok_url+"/case")
            print(r.json())
            return r.json()
@app.route('/signup', methods=['POST'])
def store():
        firstname = request.json['firstname']
        lastname = request.json['lastname']
        password = request.json['password']
        print(firstname)
        print(lastname)
        print(password)
        r = requests.post(ngrok_url+"/signup", data={'firstname':
firstname,"lastname":lastname,"password":password})
        return r.json()
if __name__ == '__main__':
    app.run(host="0.0.0.0", debug=True)
```

FRONTEND

SignUp.js

```
function SignUp1(props) {
    const [firstname, setfirstname] = useState("");
    const [lastname, setlastname] = useState("");
```

```
    const [newpassword, setnewpassword] = useState("");
    const [confirmpassword, setconfirmpassword] = useState("");


    function SignUp(e) {
        e.preventDefault();
        if (firstname == "" && lastname == "" && newpassword == "" &&
confirmpassword == "") {
            alert("Please Enter Required Details !");
        }
        else if(newpassword != confirmpassword){
            alert("New Password and confirm password does not match !");
        }
        else {
            console.log(firstname,lastname,newpassword,confirmpassword)
            const getStocksDataid = {
            "firstname": firstname,
            "lastname": lastname,
            "password": confirmpassword,
        }
        const getStocksData = {
            url: api.SIGNUP,
            method: 'POST',
            headers: {
                'Content-Type': 'application/json',
            },
            data: JSON.stringify(getStocksDataid)
        }
        axios(getStocksData)
            .then(response => {
```

```
            alert("User Created Successfully.")
            window.location.reload();

          })
          .catch(function (e) {
            if (e.message === 'Network Error') {
              alert("No Internet Found. Please check your internet connection")
            }
            else {
              alert("Sorry, something went wrong. Please try again after sometime. If
the issue still persists contact support.")
            }
          });
      }
    }

    return (
      <Aux>
        <div className="auth-wrapper" style={{
          // backgroundImage: `url(${backgroundImage})`,
          // backgroundImage: 'linear-gradient(to right, "black"',
          // backgroundColor:'line',
          backgroundPosition: 'center',
          backgroundSize: 'cover',
          backgroundRepeat: 'no-repeat',
          width: '100vw',
          height: '100vh'
        }}>
          <div className="auth-content">
```

```
                <div className="card">
                    <form onSubmit={SignUp}>
                        <div className="card-body text-center">
                        <div className="mb-1">
                                <img src={require('../../../assets/logo.png')} alt=""
style={{width:140,height:120}}/>

                            </div>
                            <h3 className="mb-4" style={{ fontFamily: 'Poppins-
SemiBold', fontSize: 28 }}>SignUp</h3>
                            <div className="input-group mb-3">
                                <input style={{ fontFamily: 'Poppins-SemiBold', fontSize: 17
}} type="text" className="form-control" placeholder="First Name*"
value={firstname} onChange={(e) => setfirstname(e.target.value)} />
                            </div>

                            <div className="input-group mb-3">
                                <input style={{ fontFamily: 'Poppins-SemiBold', fontSize: 17
}} type="text" className="form-control" placeholder="Last Name*"
value={lastname} onChange={(e) => setlastname(e.target.value)} />
                            </div>
                            <div className="input-group mb-3">
                                <input style={{ fontFamily: 'Poppins-SemiBold', fontSize: 17
}} type="text" className="form-control" placeholder="New Password*"
value={newpassword} onChange={(e) => setnewpassword(e.target.value)} />
                            </div>
                            <div className="input-group mb-4">
                                <input style={{ fontFamily: 'Poppins-SemiBold', fontSize: 17
}} type="password" className="form-control" placeholder="Confirm Password*"
value={confirmpassword} onChange={(e) => setconfirmpassword(e.target.value)} />
```

```
                        </div>


                        <button className="btn btn-primary shadow-2 mb-4"
onClick={SignUp} type="submit" style={{ fontFamily: 'Poppins-SemiBold',
fontSize: 17 }}>SignUp</button>
                      </div>
                    </form>
                  </div>
                </div>
              </div>
            </Aux>
  );
}
export default windowSize(SignUp1);
```

Home.js

```
function Dashboard() {
  const CaseManagementDetails = (data) => {
    window.history.replaceState(null, null, "/DomainPage");
    localStorage.setItem('topbarname', data);
    window.location.reload();
  }
    return (
      <>
        <div style={{ backgroundColor: '#282829', display: 'flex', alignItems:
'center', flexDirection: 'row', justifyContent: 'space-between' }}>
          <div>
            <h2 style={{ color: 'white', fontSize: 22, textAlign: 'center', padding:
20, marginTop: 10,fontFamily:'Poppins-SemiBold' }}>Home</h2>
```

```
                    </div>
                    <div className="mb-1" style={{ marginRight: 20, borderRadius: 20 }}>
                        <img src={require('../../assets/logo.png')} alt="" style={{ width: 50,
height: 50, borderRadius: 30 }} />

                    </div>
                </div>
                <Aux  >
                    <div style={{position:'relative',display:'block',padding:20}}>
                    <h6 className=" d-flex align-items-center m-b-1 ml-1" style={{
fontFamily: 'Poppins-SemiBold', fontSize: 21, lineHeight: 1.2, margin: 30
}}>SELECT THE DOMAIN ? </h6>
                    <Row >
                        <Col xl={4} lg={6} md={6} sm={12} xs={12} onClick={() =>
CaseManagementDetails("FOURTH AMENDMENT")} >
                            <Card style={{ borderRadius: 25 }}>
                                <Card.Body >
                                    <Row>
                                        <Col xl={8} lg={8} md={8} sm={8} xs={8}>
                                            <h6 className=" d-flex align-items-center m-b-1 ml-1"
style={{ fontFamily: 'Poppins-Bold', fontSize: 21, padding: 30 }}>FOURTH
AMENDMENT</h6>
                                        </Col>
                                    </Row>
                                </Card.Body>
                            </Card>
                        </Col>
                        <Col xl={4} lg={6} md={6} sm={12} xs={12} onClick={() =>
CaseManagementDetails("TRADE SECRETS")} >
```

```
                  <Card style={{ borderRadius: 25 }}>
                    <Card.Body >
                      <Row>
                        <Col xl={8} lg={8} md={8} sm={8} xs={8}>
                          <h6 className=" d-flex align-items-center m-b-1 ml-1"
style={{ fontFamily: 'Poppins-Bold', fontSize: 21, padding: 30 }}>TRADE
SECRETS</h6>
                        </Col>
                      </Row>
                    </Card.Body>
                  </Card>
                </Col>
                <Col xl={4} lg={6} md={6} sm={12} xs={12} onClick={() =>
CaseManagementDetails("WILD ANIMALS")} >
                  <Card style={{ borderRadius: 25 }}>
                    <Card.Body >
                      <Row>
                        <Col xl={8} lg={8} md={8} sm={8} xs={8}>
                          <h6 className=" d-flex align-items-center m-b-1 ml-1"
style={{ fontFamily: 'Poppins-Bold', fontSize: 21,padding: 30 }}>WILD
ANIMALS</h6>
                        </Col>
                      </Row>
                    </Card.Body>
                  </Card>
                </Col>
              </Row>
              </div>
            </Aux>
          </>
```

```
    );
}
export default Dashboard;
```

WildAnimals.json

```
[
  {
    "id": 1,
    "question": "Was the plaintiff pursuing the quarry for sport?",
    "option1": "Yes",
    "option2": "No",
    "answer":"No"

  },
  {
    "id": 2,
    "question": "Did the plaintiff seek to personally gain from the quarry?",
    "option1": "Yes",
    "option2": "No",
    "answer":"No"

  },
  {
    "id": 3,
    "question": "Was the plaintiff pursuing the quarry for their livelihood?",
    "option1": "Yes",
    "option2": "No",
    "answer":"Yes"
  },
  {
    "id": 4,
```

"question": "Was the defendant pursuing the quarry for sport?",
"option1": "Yes",
"option2": "No",
"answer":"No"

},
{
"id": 5,
"question": "Did the defendant seek to personally gain from the quarry?",
"option1": "Yes",
"option2": "No",
"answer":"No"

},
{
"id":6,
"question": "Was the defendant pursuing the quarry for their livelihood?",
"option1": "Yes",
"option2": "No",
"answer":"Yes"
},
{
"id":7,
"question": "Was the defendant malicious in their motive?",
"option1": "Yes",
"option2": "No",
"answer":"No"
},

{

"id": 8,

"question": "Was the plaintiff in hot pursuit of the quarry?",

"option1": "Yes",

"option2": "No",

"answer":"Yes"

},

{

"id": 9,

"question": "Was the quarry not caught by the plaintiff?",

"option1": "Yes",

"option2": "No",

"answer":"No"

},

{

"id": 10,

"question": "Was the plaintiff the legal owner of the land?",

"option1": "Yes",

"option2": "No",

"answer":"Yes"

},

{

"id": 11,

"question": "Was the interference of the defendant in the plaintiff's pursuits impolite?",

"option1": "Yes",

"option2": "No",

"answer":"Yes"

},
{
"id": 12,
"question": "Did the defendant's interference with the plaintiff's pursuit amount to a nuisance?",
"option1": "Yes",
"option2": "No",
"answer":"Yes"
},
{
"id": 13,
"question": "Did an assault prevent the plaintiff from retaining possession of the quarry?",
"option1": "Yes",
"option2": "No",
"answer":"No"
},

{
"id": 14,
"question": "Did the quarry reside on the land?",
"option1": "Yes",
"option2": "No",
"answer":"Yes"

},
{
"id": 15,
"question": "Is the possession of the quarry governed by convention?",
"option1": "Yes",

"option2": "No",
"answer":"No"

},

{
"id": 16,
"question": "Was the defendant blameless in the interference of the plaintiff's pursuit?",
"option1": "Yes",
"option2": "No",
"answer":"Yes"
}
]

A.4 SCREENSHOTS

```
df_WildAnimals.isna().sum()
```

```
PSport        0
PGain         0
PLiving       0
DSport        0
DGain         0
Dliving       0
Malice        0
HotPursuit    0
NotCaught     0
LegalOwner    0
Impolite      0
Nuisance      0
Assault       0
Resident      0
Convention    0
NoBlame       0
outcome       0
dtype: int64
```

Figure A.4.1 Checking for missing values

```
df_WildAnimals['outcome']=df_WildAnimals["outcome"].replace("find for the plaintiff",1)
df_WildAnimals['outcome']=df_WildAnimals["outcome"].replace("do not find for the plaintiff",0)
df_WildAnimals['outcome']
```

```
0    1
1    0
2    0
3    1
4    0
```

Figure A.4.2 Label encoding target variables

```
value = []
for data in  df_WildAnimals.iloc[:,:-1].values:
    value.append(" ".join(data))
print(value)
```

```
['No No Yes Yes No No Yes No Yes Yes No Yes No No No No', 'Yes No No No No No Yes Yes No Yes No No No No No', 'No No Yes No No Yes No Yes Yes No Yes No No No No No',
```

Figure A.4.3 Feature Concatenation of independent variables.

```
print(100*(accuracy_score(df_values["labels"],preds)))
```

```
80.0
```

```
from sklearn.metrics import classification_report
print(classification_report(df_values["labels"],preds))
```

```
              precision    recall  f1-score   support

           0       0.75      1.00      0.86         3
           1       1.00      0.50      0.67         2

    accuracy                           0.80         5
   macro avg       0.88      0.75      0.76         5
weighted avg       0.85      0.80      0.78         5
```

Figure A.4.4 Results from BERT base for WildAnimals

```
print(100*(accuracy_score(df_value["labels"],prediction)))
```

```
90.625
```

```
from sklearn.metrics import classification_report
print(classification_report(df_value["labels"],prediction))
```

```
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         1
           1       0.95      0.95      0.95        19
           2       0.85      0.92      0.88        12

    accuracy                           0.91        32
   macro avg       0.60      0.62      0.61        32
weighted avg       0.88      0.91      0.89        32
```

Figure A.4.5 Results from BERT base for Trade Secret

```
print("accuracy score is",100*(accuracy_score(df_data["labels"],predictions)))
```

```
accuracy score is 88.88888888888889
```

```
from sklearn.metrics import classification_report
print(classification_report(df_data["labels"],predictions))
```

```
              precision    recall  f1-score   support

           0       1.00      0.67      0.80         3
           1       0.86      1.00      0.92         6

    accuracy                           0.89         9
   macro avg       0.93      0.83      0.86         9
weighted avg       0.90      0.89      0.88         9
```

Figure A.4.6 Results from BERT base for Fourth Amendment

```
print(100*(accuracy_score(df_values["labels"],predsb)))
```

```
60.0
```

```
from sklearn.metrics import classification_report
print(classification_report(df_values["labels"],predsb))
```

```
              precision    recall  f1-score   support

           0       0.60      1.00      0.75         3
           1       0.00      0.00      0.00         2

    accuracy                           0.60         5
   macro avg       0.30      0.50      0.37         5
weighted avg       0.36      0.60      0.45         5
```

Figure A.4.7 Results from BERT large for WildAnimals

```
print(100*(accuracy_score(df_value["labels"],prediction)))
```

```
84.375
```

```
from sklearn.metrics import classification_report
print(classification_report(df_value["labels"],prediction))
```

```
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         1
           1       0.95      1.00      0.97        19
           2       1.00      1.00      1.00        12

    accuracy                           0.97        32
   macro avg       0.65      0.67      0.66        32
weighted avg       0.94      0.97      0.95        32
```

Figure A.4.8 Results from BERT large for Trade Secret

```
print("accuracy score is",100*(accuracy_score(df_data["labels"],predictions)))
```

```
accuracy score is 66.66666666666666
```

```
from sklearn.metrics import classification_report
print(classification_report(df_data["labels"],predictions))
```

```
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         3
           1       0.67      1.00      0.80         6

    accuracy                           0.67         9
   macro avg       0.33      0.50      0.40         9
weighted avg       0.44      0.67      0.53         9
```

Figure A.4.9 Results from BERT large for Fourth Amendment

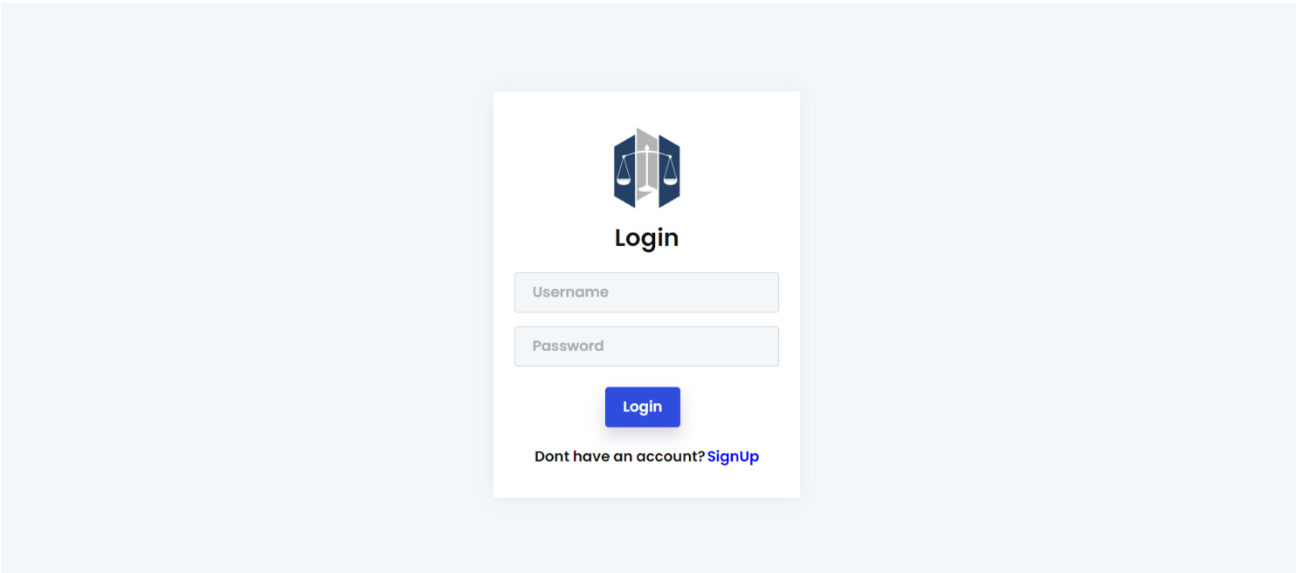

Figure A.4.10 Login Page

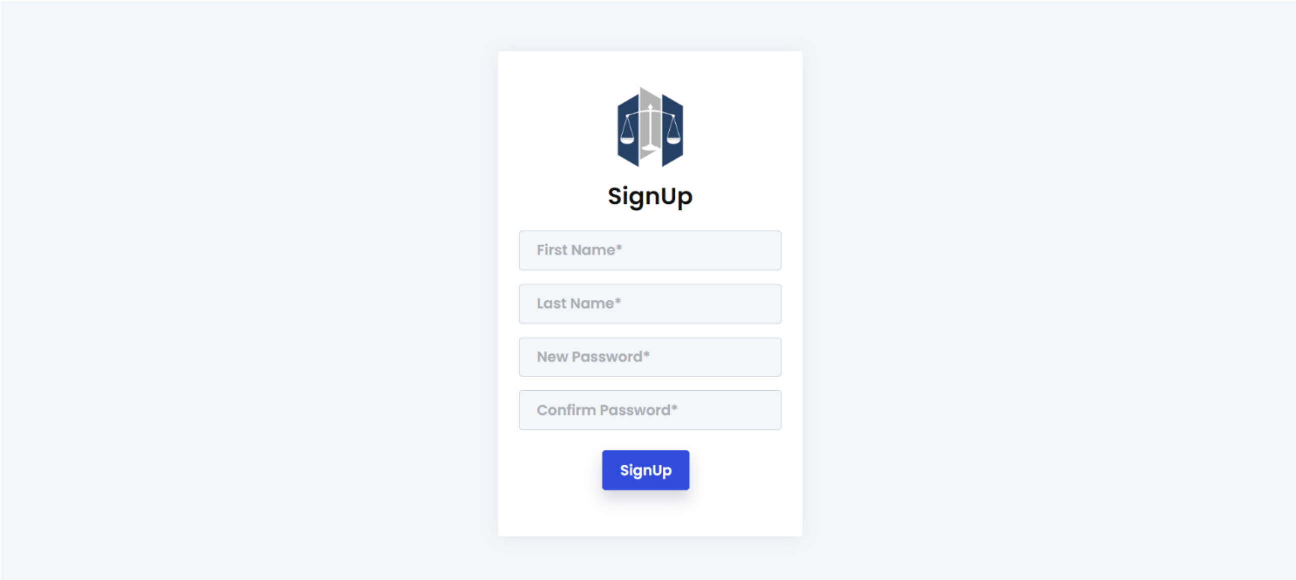

Figure A.4.11 SignUp Page

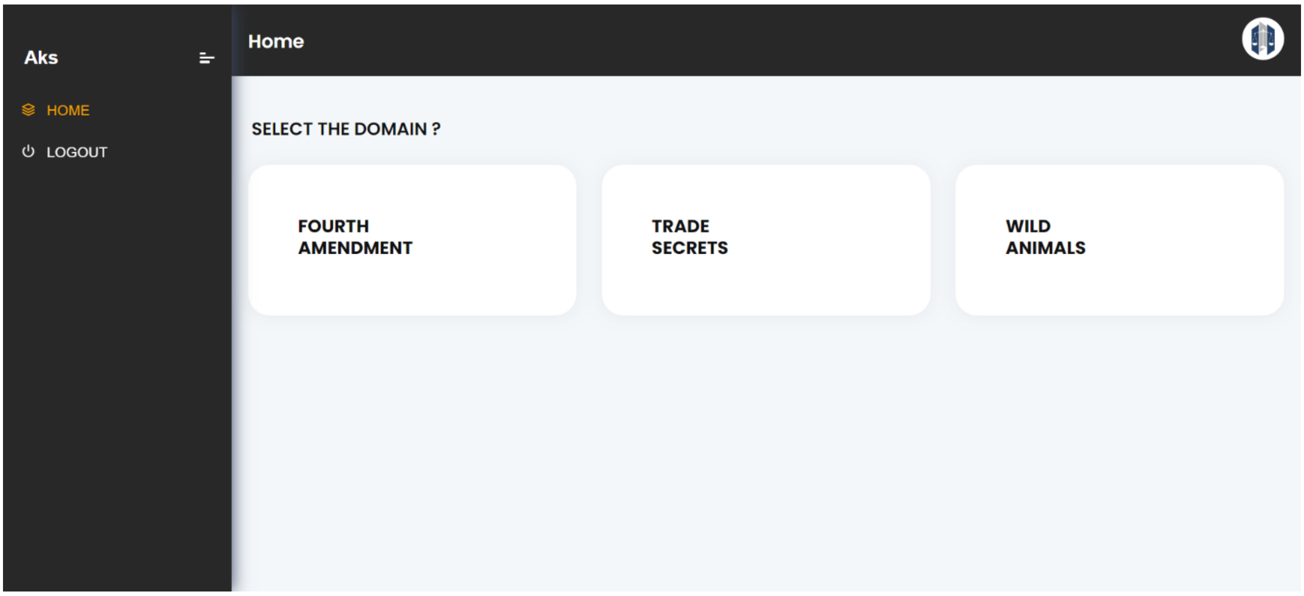

Figure A.4.12 Legal Case Domain Page

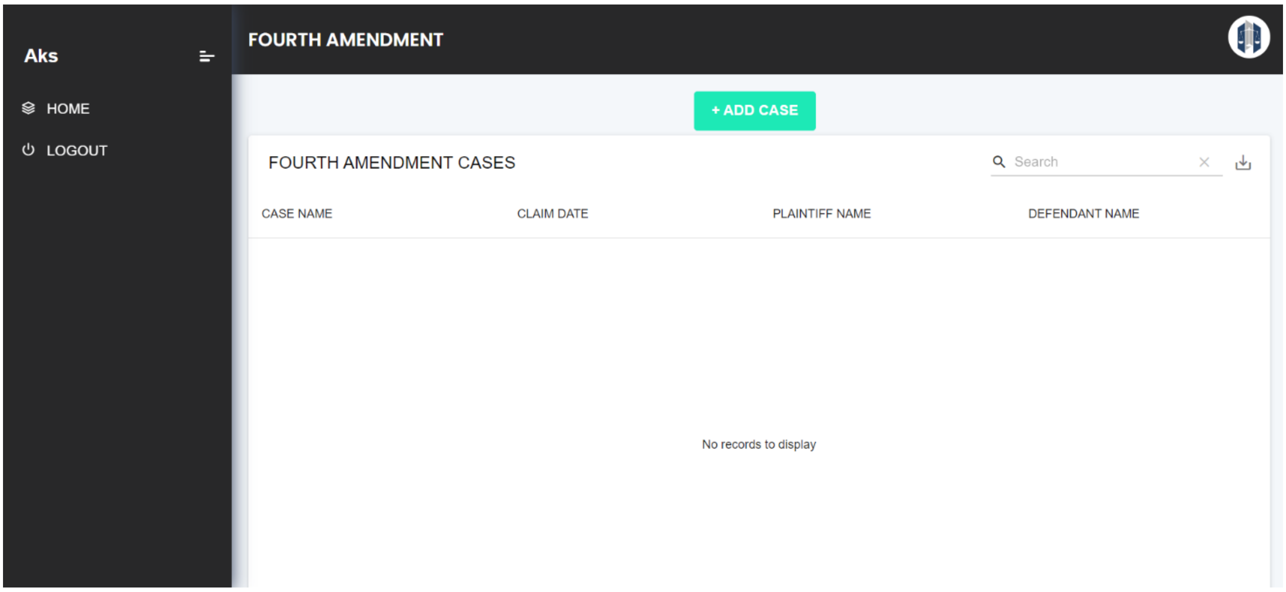

Figure A.4.13 Legal Case Home Page

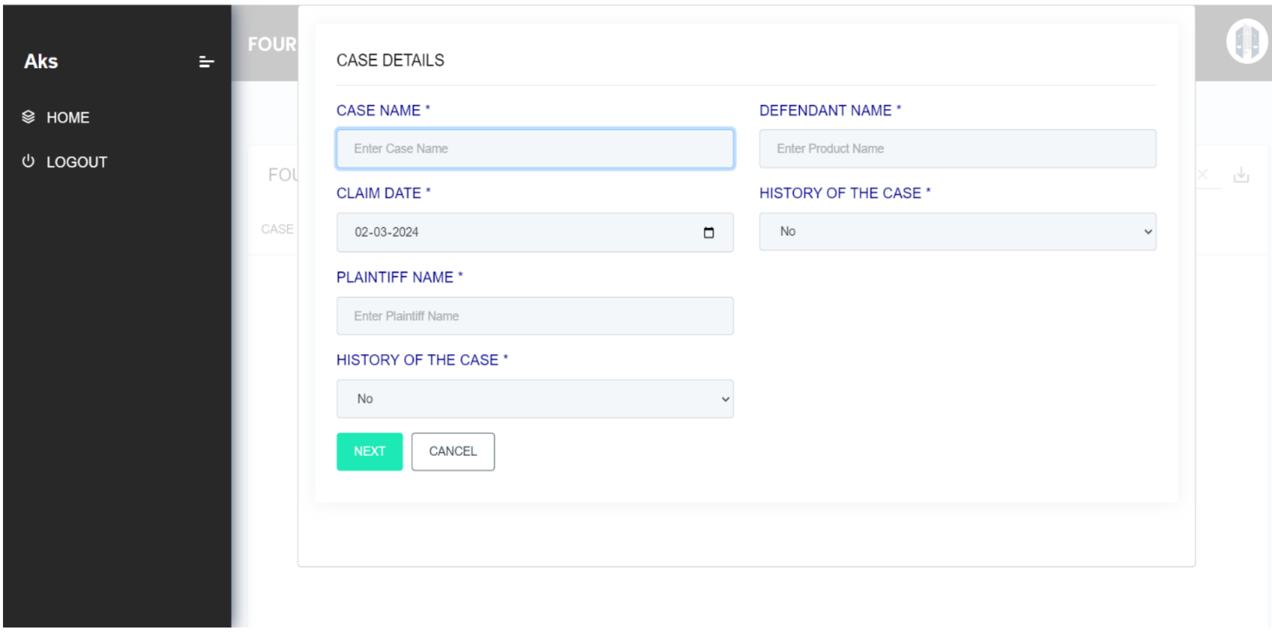

Figure A.4.14 Case Details Input Session

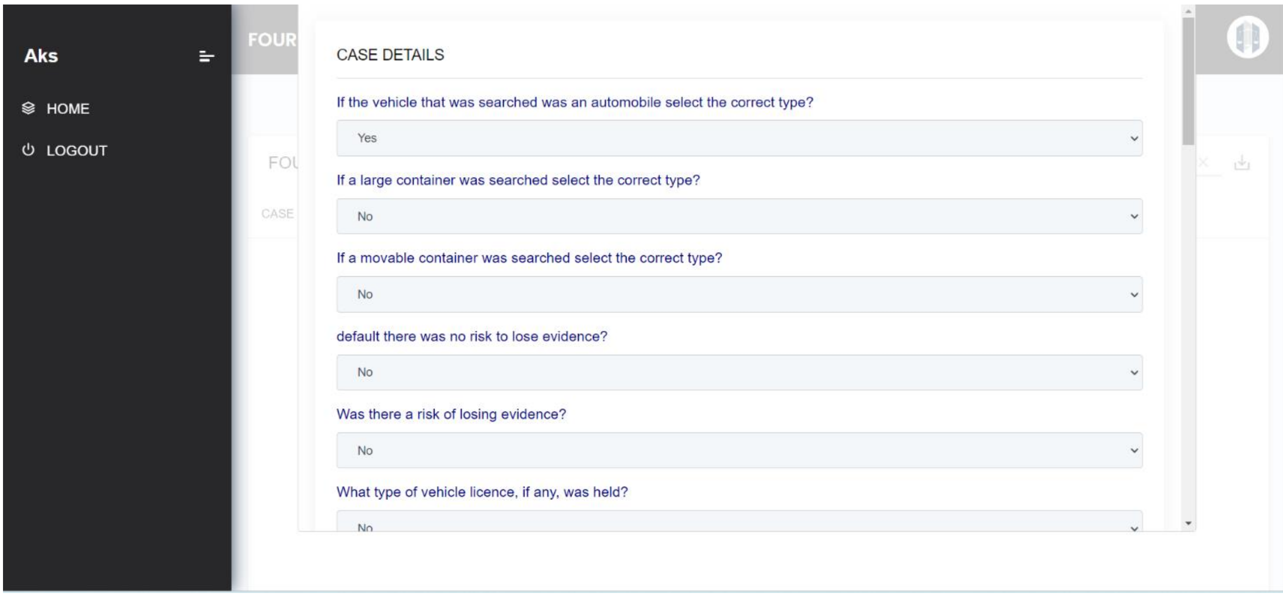

Figure A.4.15 Question and Answers Page

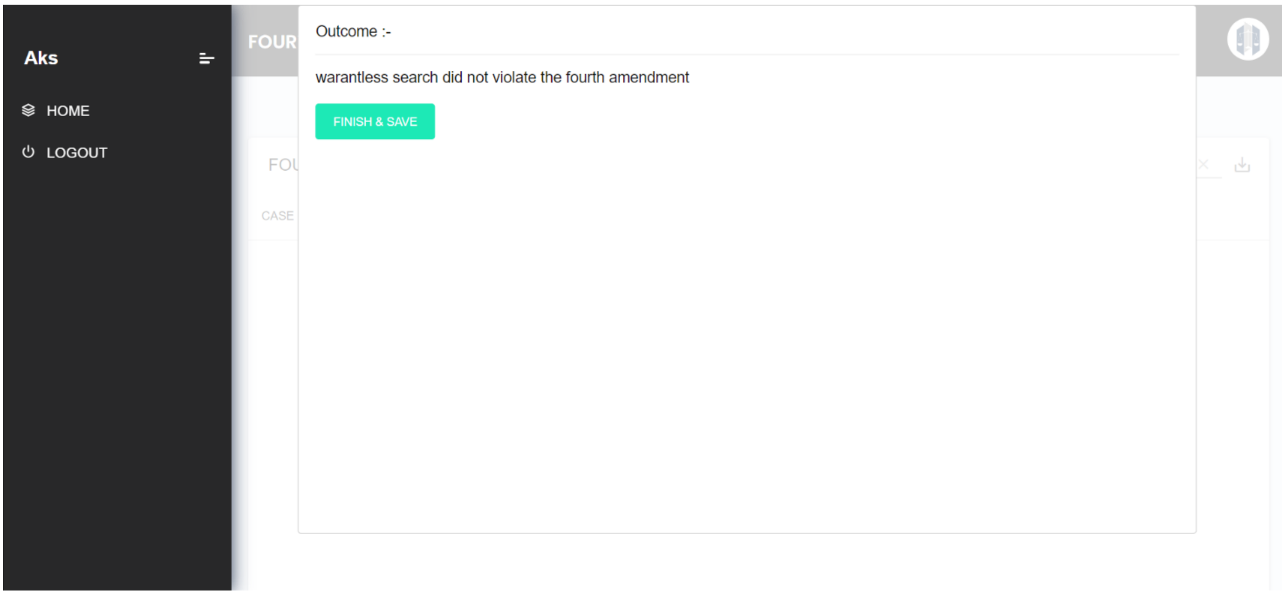

Figure A.4.16 Outcome Page

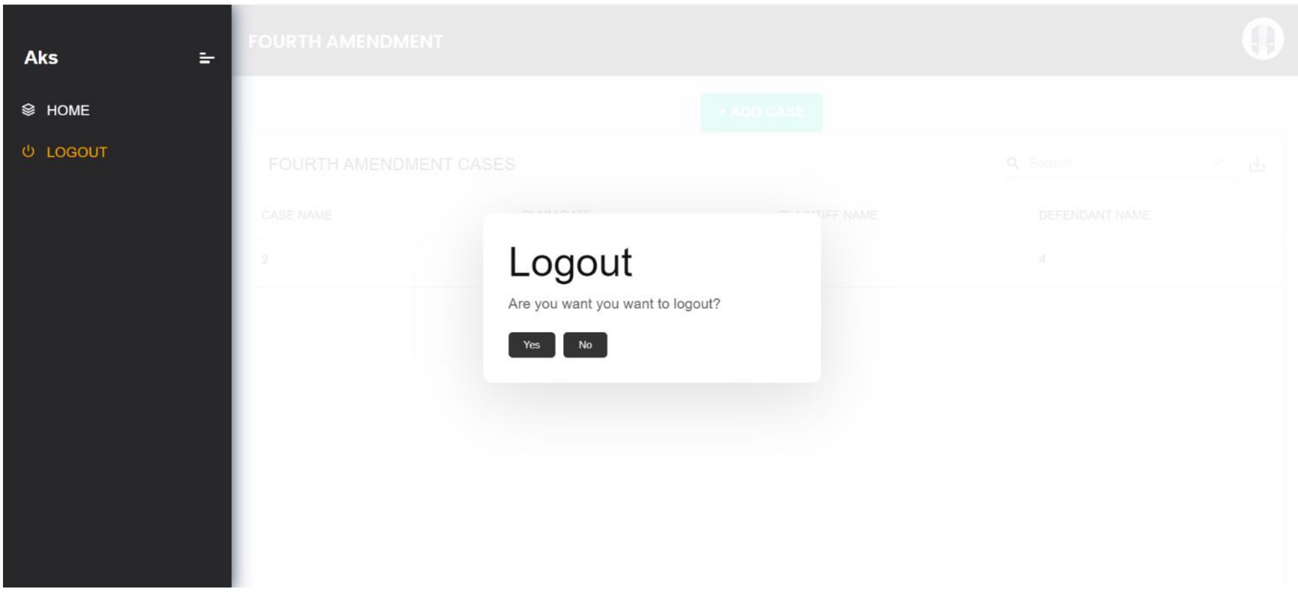

Figure A.4.17 Logout Page

A.5 PLAGIARISM REPORT

An Intelligent AI Chatbot For Legal Case Prediction Using BERT

Akshidha Unni
Dept of CSE,
Panimalar Engineering College
Chennai, India
akshidha2002@gmail.com

Deepthi L S
Dept of CSE,
Panimalar Engineering College
Chennai, India
deepthiloganathan411@gmail.com

Monisha D
Dept of CSE,
Panimalar Engineering College
Chennai, India
mmonisha4248@gmail.com

Dr Kavitha Subramani
Dept of CSE,
Panimalar Engineering College
Chennai, India
kavitha.bhskr@panimalar.ac.in

***Abstract*— Law and society are the most vitally connected accomplishments of mankind. The judicial system emerged with the emergence of human civilization. A legal case prediction chatbot which incorporates the most advanced BERT (Bidirectional Encoder Representations from Transformers)-based natural language processing models claims in quickening up legal research and bolster decision-making. The chatbot assists legal practitioners in forecasting case outcomes, locating important precedents, and providing insights into potential legal strategies through analyzing and comprehending the intricate details of legal texts. This novel use of BERT in the legal arena seeks to give legal professionals and stakeholders with an efficient and accurate tool for predicting case outcomes, hence promoting informed decision-making in the legal landscape.**

Keywords*— *Artifical Intelligence, Chatbot, BERT, Legal Case, Natural Language Processing.

I. Introduction

"Law and society are inextricably linked in our human experience. The rules we follow evolved alongside civilization. Understanding the legal system plays a crucial role in the development of communities. Making sure everyone follows the rules helps to shape our civilization. However, as our environment evolves, so does our understanding of laws. Unfortunately, many schools do not provide adequate legal education, leading to a lack of understanding of the legal systemThe administration of a lawsuit may be difficult, especially for those who are unfamiliar with with the process. Incorporation of artificial intelligence (AI) into the legal field has resulted in notable advancements in recent years that have augmented the environment in legal services. The advent of the Bidirectional Encoder Representations from Transformers (BERT) model, a significant advancement in the processing of natural languages (NLP), is one noteworthy step in this approach. Leveraging the power of BERT, we propose an intelligent AI chatbot designed to revolutionize legal case prediction.The legal profession is inherently information-intensive, with practitioners traversing massive amounts of legal texts, precedents, and case law in order to deliver effective advice and predict case outcomes. The complexity and dynamic nature of legal language provide significant problems to typical information retrieval systems. The limits of keyword-based searches and inflexible rule-based techniques demand novel solutions that can understand the complexities of legal speech. Our intelligent AI chatbot aims to bridge this gap by incorporating BERT, a cutting-edge language representation model, into the core of legal case prediction. By amalgamating cutting-edge AI technology with the intricacies of legal reasoning, our intelligent AI chatbot aspires to be a valuable companion for legal professionals, enhancing their capabilities, reducing research time, and improving the overall efficiency of legal case analysis and prediction. In the subsequent sections, we will look at the technical architecture, training approaches, and ethical considerations that have guided the development of this efficient AI-powered legal assistant.

II. Literature Survey

McConnell, Zhu, Pandya, and Aguiar [1] utilize ensemble learning methods, such as Adaboost and tree-based models, to predict legal case outcomes. Their study reveals superior accuracy in forecasting motion to strike outcomes in vehicular cases compared to tort cases, except for SVM. However, a limitation arises from courts occasionally granting motions to strike in part. Addressing this challenge could enhance prediction accuracy in legal decision-making. Song, Gao, He, and Schilder [2] present an overview of pre-trained Language Models (PLMs) in legal assistance, highlighting their strengths and limitations. They propose text selection as a promising direction to enhance PLMs' effectiveness in legal tasks, emphasizing the importance of optimizing input data. This research provides valuable insights for leveraging PLMs in legal domains and suggests avenues for future exploration. Amato, Fonisto, Giacalone, and Sansone [3] develop an intelligent AI for question answering using SBERT, with the fine-tuned quora-distilbert-multilingual model showing superior performance. However, challenges, primarily due to dataset size limitations, are identified. Future research could focus on addressing these challenges to enhance the effectiveness of SBERT-based question answering systems. Another study by Ge, Huang, Shen, Li, and Hu [4] reveals that Multi-level Matching Networks may greatly enhance recommendation accuracy integrating fine-grained fact-article correspondences. However, challenges arise due to the semi-structured nature of judgment documents, hindering automatic extraction of crucial elements like facts and cited articles. Overcoming these challenges is essential to fully leverage the benefits of this approach in recommendation systems. Said, Khudoberganov, Sharopov, and Abduvaliev analyze [5] international legal frameworks and practices regarding AI integration in courtrooms, offering recommendations based on primary and secondary sources. They advocate for ethical guidelines, transparency in algorithms, sound data management, human oversight, professional education, and international cooperation. Concerns about fairness, bias, accuracy, and accountability underscore the need for critical evaluation and ethical considerations in AI integration within legal systems.

Hassani and Silva [6] conduct a review of ChatGPT's role in data science, highlighting its potential to address skill gaps, generate synthetic data, and drive innovation. However, concerns include inaccuracies, biased training data, and limitations in handling complex inquiries, contrasting with Google's search algorithms' reliability. Future research should focus on mitigating these challenges to enhance ChatGPT's reliability and utility in data science. In [7], Franca, Boaro, dos Santos, et.al utilize XGBoost in ensemble learning, demonstrating its ability to balance sensitivity and specificity for accurate predictions. However, a drawback lies in its inability to handle high cardinality categorical variables effectively. Addressing this limitation could further enhance its applicability in predictive modelling tasks. BERT variations for multilingual legal document classification was explored by Niklaus, Chalkidis, and Stürmer [8], finding that innovative models like Long BERT and Hierarchical BERT outperform standard BERT, particularly in Macro F1 Score. While performance is consistent across German and French subsets, challenges persist in the Italian subset, suggesting the need for further investigation. Giacalone and Salehi [9] explore the integration of fair division theory and algorithmic-based dispute resolution in civil law, highlighting the potential of game-theory algorithms. While these methods offer efficiency and fairness enhancements, challenges in harmonizing them within diverse legal landscapes are identified. Future research could focus on developing adaptive approaches to accommodate legal diversity while ensuring fairness in dispute resolution. In their research Masha Medvedeva, Michel Vols and Martijn Wieling [10] employed super vector machines that has shown promise in predicting legal outcomes with 75% accuracy using simple input data. However, a significant drop in performance occurs with a 10-year data gap, likely due to reduced training data availability. Future research should focus on addressing data gaps to enhance SVM predictive accuracy in legal decision-making.

Queudot, Charton, and Meurs [11] utilize diverse NLP models to analyze discussions without specialized annotations, showcasing their versatility across different datasets. However, their study reveals a tendency for the model to produce conservative, uninteresting responses. Future work should focus on refining algorithms to encourage more engaging and diverse outputs in discussion analysis. Zheng, Liu, and Sun [12] introduce KD-BERT, a knowledge distillation-based legal decision prediction model known for its faster inference speed compared to other BERT models. However, the model's extensive 115M encoder parameters pose challenges for practical application in decision prediction algorithms. Future research could focus on optimizing model complexity while maintaining predictive accuracy in legal contexts. Limsopatham [13] investigates BERT's adaptation for legal text classification, highlighting its proficiency in various NLP tasks. While BERT shows promise, challenges like high pre-training costs and handling long documents impact its performance. Future research should focus on mitigating these challenges to enhance BERT's utility in legal text classification. Wehnert, Sudhi, Dureja, Kutty, Shahania, and De Luca [14] explore combining Sentence-BERT and LEGAL-BERT with TF-IDF vectors, employing BERT Score for similarity scoring in statute retrieval tasks. Their research involves data enrichment, augmentation, and hyperparameter optimization, leading to improved recall and winning outcomes. The lack of clarity in BERT-based methods, the further complexity of combining Sentence-BERT with TF-IDF, and the disparity in BERTScore's performance are among the issues, though. Despite these challenges, their strategies demonstrate significant advancements in statute law retrieval tasks.

Biresaw and Saste [15] explore the impact of Artificial Intelligence (AI) on legal research, particularly through Legal AI tools. They highlight challenges in addressing legal issues requiring human empathy and creativity while meeting client expectations. Concerns include limited contextual understanding, ethical considerations, and potential job displacement. Balancing AI benefits with human qualities is crucial for navigating these challenges in the legal profession. In their study on critiquing mechanisms for Dynamic Conversation-based Recommendation Systems (DCRS), Cai, Jin, and Chen [16] identify effective techniques to improve user interaction with recommendation chatbots across diverse needs. While personalized recommendations enhance user engagement, they caution that over-alignment with current preferences could narrow users' exploration space. Future research should focus on adaptive critiquing strategies to balance personalization and recommendation diversity in DCRS. CNN with PCA for legal data analysis was implemented by Shang [17], achieving superior results compared to benchmarks across four datasets. However, the CNN model requires further fine-tuning to adapt to changing legal needs effectively. This research highlights the promising potential of deep learning techniques in legal data analysis while emphasizing the importance of ongoing refinement for optimal performance.

III. Dataset

Future The database design for training the model includes three different datasets with respect to three different categories of legal cases which are as follows:

- Wild Animals
- Trade Secret
- Fourth Amendment – Land Disputes

Each one of these datas has several columns which are features or questions with respect to the particular legal case category. Each of these columns has Yes or No as records.

IV. Preprocessing Techniques

A. *Handling the missing values*

Missing values can have an adverse effect on model performance, and properly treating them is critical for accurate predictions. Several strategies are used to manage missing data, including imputation and elimination. Imputation is the process of substituting missing numbers with approximated or calculated values based on data currently available. Mean or median imputation is a common method for numerical characteristics, while mode imputation is used for categorical features. Alternatively, removal entails eliminating occurrences with missing values from the dataset. While straightforward, this method is only appropriate when the number of missing values is low and eliminating instances has no substantial influence on the total dataset. We had no missing values in our data hence there was no neccessity to treat them.

B. *Label encoding target variable*

Label encoding of the target variable is a preprocessing technique used frequently in machine learning when dealing with categorical target variables. This procedure converts categorical labels into numerical representations, enabling machine learning algorithms more efficient. Each unique category is allocated a unique number, converting the target variable into a format that algorithms can easily grasp. Label encoding simplifies the modelling process by translating categorical labels into numerical values, making them compatible with algorithms that require numerical input. However, it is important to note that this encoding assumes an ordinal relationship between the encoded values, which may not necessarily be correct. For this project we have encoded our target variables as 0,1, and 2.

C. *Feature Concatenation of Independent Variables*

Concatenating independent variables before introducing them to a BERT model is typical in natural language processing instances where the input data contains text and non-text elements. It is common for BERT models to need a single input format, so concatenation lets you combine many feature types into a single input sequence. This unified format simplifies the model's input representation while also providing a uniform framework for processing. BERT, which was built for natural language processing applications, processes text sequences by default. When working with datasets with a variety of attributes, such as numerical or categorical data, concatenation ensures a consistent input structure. This extensive input representation enables BERT to capture correlations between textual and non-textual elements, which improves the model's comprehension of context and predictive capabilities.

V. Proposed Method

Bidirectional Encoder Representation for Transformers

For processing of natural languages (NLP), BERT is a publicly accessible machine learning system. BERT serves as a means to help computers comprehend the underlying significance of unfamiliar phrases in phrase by obtaining background details from the adjoining material. The the Wikipedia website information was used to initiate training the BERT structure, which may be improved with the use of query and response data. Transformers is a deep learning model that is a base of BERT (Bidirectional Encoder Representations from Transformers). Each input element includes a final value confided in it, and the ratio of weights between them are created in real time tailored to their relationship.

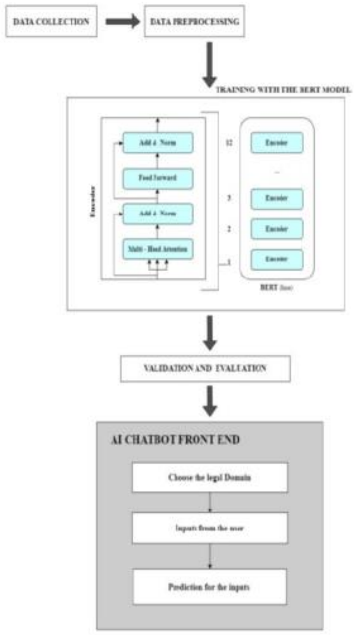

Fig. 1. Architecture diagram

Typically, language models were able to comprehend text input sequentially, coming not concurrently, but rather from the left to the right, or the right towards the left. This ability, which is facilitated by the release of transformers, is known as bidirectionality. BERT has previously been trained on a pair of separate but related NLP tasks—masked language modelling and future sentence prediction—using this bidirectional capacity. Masked Language Model (MLM) programming strives to cover-up a word that lie within a phrase, with the software forecasting which word was disguised by evaluating its context. Teaching the application to ascertain if two given sentences have a logical, coordinated connection is among the primary objectives of Next Sentence Prediction training.

. The MLM may be expressed as follows:

$$\mathcal{L}_{MLM} = -\sum_{i} (\log P(y_i | x_1, x_2, \ldots, x_n)) \quad (1)$$

where:

- $x_1, x_2, \ldots, x_n$ are input tokens.
- y_i is the ground truth token for masked position i.
- $P(y_i | x_1, x_2, \ldots, x_n)$ is the predicted probability for the correct token

The NSP can be given as follows:

$$\mathcal{L}_{NSP} = -(IsNext(S_1, S_2). \log P\,(IsNext = 1 | \, S_1, S_2) \\ + \left(1 - IsNext(S_1, S_2)\right). \log P(IsNext = 0 | S_1, S_2)) \quad (2)$$

where:

- $IsNext(S_1, S_2)$ is a binary indicator (1 if S_2 is the next sentence after S_1, 0 otherwise)
- $\log P\,(IsNext = 1 | \, S_1, S_2)$is the predicted probability fror the sentences being consecutive.
- $\log P\,(IsNext = 0 | \, S_1, S_2)$is the predicted probability for the sentences not being consecutive.

Overall, the BERT model can be given as:

$$\mathcal{L}_{BERT} = \mathcal{L}_{MLM} + \lambda . \mathcal{L}_{NSP} \quad (3)$$

where:

- $\mathcal{L}_{BERT}$ is the combined loss.
- λ is a hyperparameter that controls the trade-off between the MLM and NSP objectives.

The main objective of any NLP approach is to get to know the language of humans way it sounds when said authentically. In the overall setting of BERT, which frequently involves spotting a word in a blank. In order to accomplish this, models typically must learn on an extensive number of specialised, labelled training data. This will require large human data tagging In order to achieve this, models generally require to be programmed on an excessive amount of tagged, specific training data. by the language teams. On the other hand, BERT was trained purely on an unlabelled, basic text corpusIt continues to learn unassisted from unidentified content, optimising even when used in real-life situations. Its pre-training serves as the foundation for future "knowledge" development.

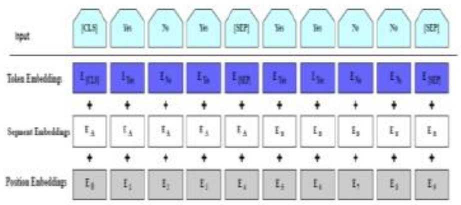

Fig. 2. Training the Inputs

BERT have the ability to recognise a wide range of patterns in language and structural patterns since it has been pre-trained on a considerable sum of unlabelled text material. During the pre-training phase, the model is presented with different kinds of datasets to get a varied understanding of the language. The model learns to predict missing words in sentences, producing comprehensive contextual embeddings that capture semantic relationships in the text. BERT's bidirectional training and contextualised embeddings make it particularly useful in a variety of downstream NLP applications. When fine-tuned on task-specific labelled datasets, BERT uses its pre-trained knowledge to perform exceptionally well on sentiment analysis, text classification, and named entity identification.

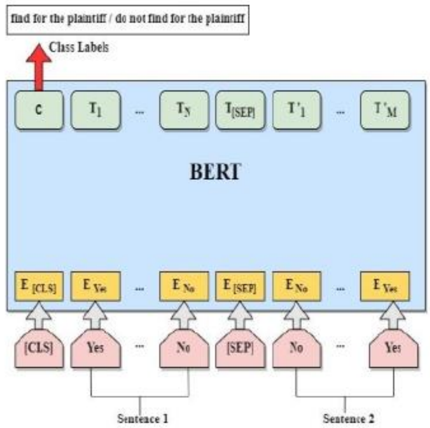

Fig. 3. Single Sentence Classification

VI. Performance Metrics

A. Accuracy

Accuracy is a commonly used performance metric in the field of machine learning that rates how precisely a model can be in its predictions. It focuses on the ratio of the number of precise forecasting made by the model to the overall number of guesses. This statistic shows how well the model turns out when its distribution of the distinct component is balanced. The following formula can be used to detect accuracy:

$$\textbf{Accuracy} = \frac{\textbf{Number of correct predictions}}{\textbf{Total number of predictions}} \quad (4)$$

B. Precision

In the application of machine learning, precision is a performance metric used to assess how well a categorization framework's optimistic predictions have come true. When assessing positive outcomes, a model with an elevated amount preciseness score makes fewer erroneous positive predictions and makes a smaller number of errors overall. One might evaluate precision by applying the following formula:

$$\textbf{Precision} = \frac{\textbf{True Positive}}{\textbf{True Positive} + \textbf{True Negative}} \quad (5)$$

C. Recall

Recall is an index of performance that estimates the percentage of instances that were favourable that the model accurately classified as favourable, in addition to accuracy. In addition to accurately identifying a large number of positive examples, an analytical framework with an impressive level of recall can provide few impartial a deficit predictions. A possible approach to describe a model's recall is as follows:

$$\text{Recall} = \frac{\textbf{True Positive}}{\textbf{True Positive + False Negative}} \tag{6}$$

D. F1 Score

It is typically employed in models that go over the effects of algorithms and factor in data into two distinct categories. It's been referred to as the average recall and exactness harmony. In most instances, it comprises a value between a value of zero to one, where 1 is the greatest possible F1 score. The F1 score has been determined implementing the following formula:

$$\textbf{F1 Score} = \frac{\textbf{2 x Precision x Recall}}{\textbf{Precision + Recall}} \tag{7}$$

VII. Results and Discussion

The table displayed below outlines the accuracy achieved by utilising multiple classifiers for classification.

TABLE I. ACCURACY SCORES

Legal Case Type	BERT $_{Base}$	BERT $_{Large}$
Wild Animals	80.00	60.00
Trade Secret	90.62	84.37
Fourth Amendment	88.88	66.66

The table showcases the accuracy, recall, and f1 scores for three distinct legal case domains.

TABLE II. CLASSIFICATION SCORES FOR BERT $_{BASE}$

Legal Case Type	Precision	Recall	F1 Score
Wild Animals	0.60	0.98	0.75
Trade Secret	0.86	0.99	0.92
Fourth Amendment	0.67	0.98	0.80

TABLE III. CLASSIFICATION SCORES FOR BERT $_{LARGE}$

Legal Case Type	Precision	Recall	F1 Score
Wild Animals	0.67	1.00	0.75
Trade Secret	0.85	0.97	0.97
Fourth Amendment	0.73	0.99	0.80

For all the three datasets, the BERT $_{Base}$ model has better accuracy than the BERT $_{Large}$ model while the BERT $_{Large}$ model has better F1 score, precision and recall than the base model. But both the types of BERT models is able to perform better than the linear models.

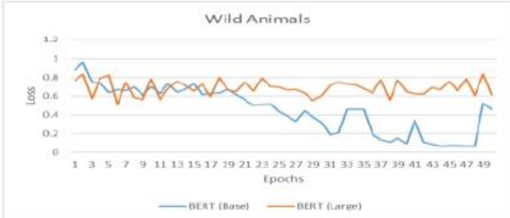

Fig. 4. Graph for loss curve of WildAnimals dataset

Fig. 5. Graph for loss curve of Trade Secret dataset

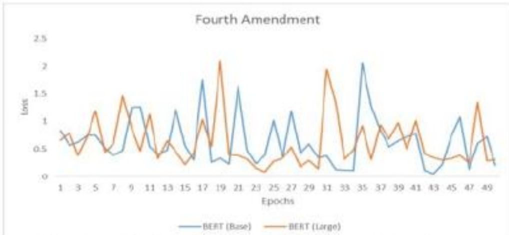

Fig. 6. Graph for loss curve of Fourth Amendment dataset

It can be clearly seen from the graphs that for Wild Animals dataset eventhough the model starts with a high error it decreases to about 0.5 at the end of the training, as for trade secrets and fourth amendments the error reaches almost 0 at the end.

Conclusion

In conclusion, the implementation of an intelligent AI chatbot for legal case prediction based on the BERT model constitutes a significant step forward in the integration of artificial intelligence and legal services. The incorporation of BERT, with its bidirectional training and contextualised word embeddings, gives the chatbot an unprecedented ability to comprehend the nuances of legal language and anticipate case outcomes with complexity and accuracy that was before unreachable. By delivering a user-friendly interface, the chatbot extends its potential to legal practitioners, researchers, and the general public, enabling legal case analysis more accessible and efficient. Alongside predictions, the chatbot functions as an exhaustive legal research assistant, offering applicable precedents, statutes, and case law, decreasing research time and helping to enhance legal practice.

REFERENCES

[1] Devin J. McConnell, James Zhu, Sachin Pandya, Derek Aguiar (2021). Case-level Prediction of Motion Outcomes in Civil Litigation. ACM, Pages 99–108.

[2] Dezhao Song, Sally Gao, Baosheng He, Frank Schilder (2022). On the Effectiveness of Pre-Trained Language Models for Legal Natural Language Processing: An Empirical Study. IEEE,vol.10,pp 75835-75858.

[3] Flora Amato, Mattia Fonisto, Marco Giacalone, d Carlo Sansone (2023). An Intelligent Conversational Agent for the Legal Domain. MDPI.

[4] J. Ge, Y. Huang, X. Shen, C. Li and W. Hu (2021). Learning Fine-Grained Fact-Article Correspondence in Legal Cases. IEEE vol.29, pp.3694-3706.

[5] Gulyamov Said, Khudoberganov Azamat , Sharopov Ravshan, Abduvaliev Bokhadir (2023). Adapting Legal Systems to the Development of Artificial Intelligence: Solving the Global Problem of AI in Judicial Processes. IRSHAD Journals,vol.1.

[6] Hossein Hassani, Emmanuel Sirmal Silva (2023). The Role of ChatGPT in Data Science: How AI-Assisted Conversational Interfaces Are Revolutionizing the Field. MDPI.

[7] JoaoV.F.Franca, Jose M.C.Boaro, Pedro T.C dos Santos et.al (2020). Legal Judgment Prediction in the Context of Energy Market using Gradient Boosting. IEEE.

[8] Joel Niklaus, Ilias Chalkidis, Matthias Stürmer (2021). Swiss-Judgment-Prediction: A Multilingual Legal Judgment Prediction Benchmark. arXiv.

[9] Marco Giacalone , Sajedeh Salehi (2021). CREA: An Introduction to Conflict Resolution with Equitative Algorithms. SSRN.

[10] Masha Medvedeva, Michel Vols, Martijn Wieling (2020). Using machine learning to predict decisions of the European Court of Human Rights. Springer.

[11] Marc Queudot, Eric Charton, Marie-Jean Meurs (2020). Improving Access to Justice with Legal Chatbots. MDPI pp.356-375.

[12] Min Zheng, Bo Liu, Le Sun (2022). Study of Deep Learning-Based Legal Judgment Prediction in Internet of Things Era. Hindawi.

[13] Nut Limsopatham. (2021). Effectively Leveraging BERT for Legal Document Classification. Year.

[14] Sabine Wehnert, Viju Sudhi, Shipra Dureja, Libin Kutty, Saijal Shahania, Ernesto W. De Luca (2021). Legal Norm Retrieval with Variations of the BERT Model Combined with TF-IDF Vectorization. ACM pp.285-294.

[15] Samuel Maireg Biresaw, Abhijit Umesh Saste (2022). The Impacts of Artificial Intelligence on Research in the Legal Profession. IJLS vol.5,pp.53-65.

[16] Wanling Cai , Yucheng Jin , and Li Chen (2022). Task-Oriented User Evaluation on Critiquing-Based Recommendation Chatbots. IEEE,vol.52 pp.354-366.

[17] Xuerui Shang (2022). A Computational Intelligence Model for Legal Prediction and Decision Support. Hindawi.

[18] Vijit Malik, Rishabh Sanjay, Shubham Kumar Nigam, Kripabandhu Ghosh, Shouvik Kumar Guha, Arnab Bhattacharya, Ashutosh Modi (2021). ILDC for CJPE: Indian Legal Documents Corpus for Court Judgment Prediction and Explanation, ACL Anthology.

BERT report

ORIGINALITY REPORT

3%	3%	2%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Higher Education Commission Pakistan Student Paper	1%
2	Submitted to Liverpool John Moores University Student Paper	1%
3	Zhen Ye, Jie Wang, Tao Sun, Jinxin Zhang, Wei Li. "Cross-Domain Few-Shot Learning Based on Graph Convolution Contrast for Hyperspectral Image Classification", IEEE Transactions on Geoscience and Remote Sensing, 2024 Publication	<1%
4	repository.tudelft.nl Internet Source	<1%
5	aclanthology.org Internet Source	<1%
6	pure.rug.nl Internet Source	<1%

7	"Natural Language Processing and Chinese Computing", Springer Science and Business Media LLC, 2023 Publication	<1%
8	www2.mdpi.com Internet Source	<1%
9	Sabine Wehnert, Viju Sudhi, Shipra Dureja, Libin Kutty, Saijal Shahania, Ernesto W. De Luca. "Legal norm retrieval with variations of the bert model combined with TF-IDF vectorization", Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law, 2021 Publication	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		

REFERENCES

1. Devin J. McConnell, James Zhu, Sachin Pandya, Derek Aguiar (2021). Case-level Prediction of Motion Outcomes in Civil Litigation. ACM, Pages 99–108.

2. Dezhao Song, Sally Gao, Baosheng He, Frank Schilder (2022). On the Effectiveness of Pre-Trained Language Models for Legal Natural Language Processing: An Empirical Study. IEEE,vol.10,pp 75835-75858.

3. Flora Amato, Mattia Fonisto, Marco Giacalone, d Carlo Sansone (2023). An Intelligent Conversational Agent for the Legal Domain. MDPI.

4. J. Ge, Y. Huang, X. Shen, C. Li and W. Hu (2021). Learning Fine-Grained Fact-Article Correspondence in Legal Cases. IEEE vol.29, pp.3694-3706.

5. Gulyamov Said, Khudoberganov Azamat , Sharopov Ravshan, Abduvaliev Bokhadir (2023). Adapting Legal Systems to the Development of Artificial Intelligence: Solving the Global Problem of AI in Judicial Processes. IRSHAD Journals,vol.1.

6. Hossein Hassani, Emmanuel Sirmal Silva (2023). The Role of ChatGPT in Data Science: How AI-Assisted Conversational Interfaces Are Revolutionizing the Field. MDPI.

7. JoaoV.F.Franca, Jose M.C.Boaro, Pedro T.C dos Santos et.al (2020). Legal Judgment Prediction in the Context of Energy Market using Gradient Boosting. IEEE.

8. Joel Niklaus, Ilias Chalkidis, Matthias Stürmer (2021). Swiss-Judgment-Prediction: A Multilingual Legal Judgment Prediction Benchmark. arXiv.

9. Marco Giacalone , Sajedeh Salehi (2021). CREA: An Introduction to Conflict Resolution with Equitative Algorithms. SSRN.

10. Masha Medvedeva, Michel Vols, Martijn Wieling (2020). Using machine learning to predict decisions of the European Court of Human Rights. Springer.

11. Marc Queudot, Eric Charton, Marie-Jean Meurs (2020). Improving Access to Justice with Legal Chatbots. MDPI pp.356-375.

12. Min Zheng, Bo Liu, Le Sun (2022). Study of Deep Learning-Based Legal Judgment Prediction in Internet of Things Era. Hindawi.

13. Nut Limsopatham, (2021). Effectively Leveraging BERT for Legal Document Classification. Year.

14. Sabine Wehnert, Viju Sudhi, Shipra Dureja, Libin Kutty, Saijal Shahania, Ernesto W. De Luca (2021). Legal Norm Retrieval with Variations of the BERT Model Combined with TF-IDF Vectorization. ACM pp.285-294.

15. Samuel Maireg Biresaw, Abhijit Umesh Saste (2022). The Impacts of Artificial Intelligence on Research in the Legal Profession. IJLS vol.5,pp.53-65.

16. Wanling Cai , Yucheng Jin , and Li Chen (2022). Task-Oriented User Evaluation on Critiquing-Based Recommendation Chatbots. IEEE,vol.52 pp.354-366.

17. Xuerui Shang (2022). A Computational Intelligence Model for Legal Prediction and Decision Support. Hindawi.

18. Vijit Malik, Rishabh Sanjay, Shubham Kumar Nigam, Kripabandhu Ghosh, Shouvik Kumar Guha, Arnab Bhattacharya, Ashutosh Modi (2021), ILDC for CJPE: Indian Legal Documents Corpus for Court Judgment Prediction and Explanation, ACL Anthology.