Machine Learning Dataset for Criminal Law Suggestions Using Case Studies Within The Context of Bangladesh

by

Ahnaf Arif Tasfin
21101156
Wasif Islam
21101199

A Final-thesis report submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
February 2025

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Ahnaf Arif Tasfin	**Wasif Islam**
21101156	21101199

Approval

The thesis titled "Machine Learning Dataset for Criminal Law Suggestions Using Case Studies Within The Context of Bangladesh" Submitted by

1. Ahnaf Arif Tasfin (21101156)
2. Wasif Islam (21101199)

Of Fall, 2024 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on January 31, 2025.

Examining Committee:

Supervisor:
(Member)

Moin Mostakim

Senior Lecturer
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Md. Golam Rabiul Alam

Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi

Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

With the growth of Language Modeling and the upcoming natural language processing-assisted tools which aim for text generation, can someday render bureaucracy meaningless while also decreasing human workload. To bring about that day, we need datasets which are able to train those models. Especially in the case of Bangladesh where there are very few datasets based on Bangladesh's legal case studies. In this paper, we have created a Multi-Label and Binary classification dataset through data augmentation using verified case studies from the Manupatra database with the criminal law subject within the context of Bangladesh and have tested them with a few models such as DistilBERT, BERT, GPT-2, GPT-3, XLNet and Mamba to classify the acts involved and court in a case study for a 2000 data split and 3000 data split. So far, we have collected 3000 criminal case studies for our augmented dataset. Experimental results showed that out of all the models, Mamba performed the best while GPT-2 came up with the worst results. DistilBERT showed almost similar results to BERT and XLNet despite their computational differences during the benchmarking process of our augmented dataset.

Keywords: Machine Learning; Natural language processing; Bangladesh; Manupatra; Case study; Predict; Bidirectional Encoder Representations; Neural Networks

Table of Contents

List of Figures

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

$BERT$ Bidirectional Encoder Representations from Transformers

$BIGRU$ Bidirectional Gated Recurrent Unit

GRU Gated Recurrent Unit

NLP Natural Language Processing

NN Neural Network

$UNDP$ United Nations Development Programme

Chapter 1

Introduction

1.1 Thoughts behind the Dataset

There has been research on law predictions and suggestive models such as legal topic classification [1], court opinion generation [9] and analysis [2], legal information extraction [6], and entity recognition [5] Yet there is little to none research for such models within the context of Bangladesh's laws and structure. It is understandable as Bangladesh has started its journey towards digitalization just a few years ago and the complexities of laws warrant caution while dealing with them [17]. Nonetheless, Natural Language Processing and its advancements have cleared the path for such research if datasets required for such research exist. We believe now is the time to slowly open the door towards this goal for Bangladesh by creating a augmented dataset using the verified criminal case studies from the Manupatra database. Such models trained by the dataset may assist legal practitioners by cutting down the time needed to understand which acts would be in play for different cases, help with generating judgements for an understanding of the case, judgement summarization, outcome prediction, descriptive analysis, sentiment analysis and legal research. This work contributes a new criminal legal acts augmented dataset of verified criminal case studies from the Supreme Court of Bangladesh which were gathered from the Manupatra database, provided by the BRACU Ayesha Abed Library. Manupatra, is one of the largest online legal research databases, the coverage of the database includes but is not limited to Bangladesh Supreme Court, Pre 1970, Statutes and Ordinances. The database provides more than 10,000 case studies with the subject criminal law from the Supreme Court of Bangladesh which we will be using for our augmented dataset. During the review of previous workings of legal text analysis, it is easy to see that legal text are mostly kept in unstructured format, any automation process must require the dataset to be properly preprocessed and feature engineered [14]. For our augmented dataset, we have structured 22 features in total from each criminal case study which include No., case id, case name, court, decision date, judges, appellant counsel, respondent counsel, subject, relevant section, acts involved, case note and judgment(1-10). We have restricted ourselves to just the criminal law subject for now but future studies can add upon the augmented dataset using the same format and test it using various models. For now we have collected three thousand case studies manually in the hopes that there will be little to no data loss which might have otherwise been caused because of automation. Our work is going to present a methodology for analysis of legal acts which can be depicted

using sections extracted from case note and judgment and discuss the possibilities of data augmentation to further provide uses of the raw data. We have decided to test the augmented dataset after preprocessing to create a benchmark and see the effectiveness of our data augmentation, first for 2000 case studies and then for 3000 case studies focusing mainly on classification.

1.2 Thoughts behind the Models

The models we will be using for the benchmark are DistilBERT, BERT, GPT-2, GPT-3, XLNet and Mamba. Bidirectional Encoder Representations from Transformers, also known as BERT is a pre-trained language model that can capture contextual information from both sides in a sentence and it has been applied and tested for many NLP tasks which include text classification and named entity recognition [11]. BERT does have a word limitation which will hinder our performance as we have long-text inputs. DistilBERT is short for Distilled BERT which is a smaller and faster version of BERT that has 97% of BERT's language comprehensive capacity while being 60% faster and 40% smaller. The model achieves this by using a process called information distillation where a smaller model learns to mimic a larger model, which in this case is BERT. We will be using this to compare the computational power required to understand the dataset. Generative Pre-trained Transformer 2 or GPT-2 focuses on generating text. It uses a unidirectional transformer decoder to predict the next word in a sequence based on preceding words. We will be using this model to compare the performance of our dataset on models not well suited for Multi-Label classification. GPT-3 excels in a wide range of NLP tasks which include text completion, translation, summarization and more. Just as GPT-2, GPT-3 is also not as well suited for multi-label classification tasks due to its unidirectional nature. We will use GPT-3 to compare it's performance on our dataset and evaluate the results compared to GPT-2. XLNet is a generalized autoregressive pretraining method for language understanding tasks. It uses a permutation-based training objective, which allows it to capture bidirectional context like BERT but without the limitations of the masked language modeling approach. XLNet also incorporates ideas from Transformer-XL, enabling it to handle longer sequences of text efficiently by maintaining state information between segments. This makes it ideal for our classification task with long texts. Mamba is a model designed specifically for efficient handling of long-text sequences. It uses bi-directional transformer architecture allowing it to understand contextual information from both directions within a sentence. Mamba also incorporates memory-augmentated attention and hierarchical processing to manage longer sequences better. This makes Mamba the most suitable for tasks such as multi-label classification. By benchmarking using Mamba, we want to compare its performance on our dataset with other models like BERT and XLNet.

1.3 Problem Statement

Within the context of Bangladesh's legal case studies, there is little to no research on classification models for legal acts. According to 'UNDP', For Bangladesh, AI will be one of the major drivers in the journey to 2041[22]. We can already see the

changes as different industries move to a more AI driven approach while the legal sector lags behind. The lack of suitable datasets in the legal sector have made it difficult and more time consuming to do research. Not only that, the complexities of legal data can be hard to handle if not properly verified. Our goal is to address these challenges by creating a criminal legal acts classification dataset through data augmentation using verified case studies from the Manupatra database.

We will have a set of criminal case studies:

$$D = \{D_1, D_2, \ldots, D_N\} \tag{1.1}$$

where each case study (D_i) consists of relevant information such as case notes, judgments, and other contextual details.

Each case study is associated with a set of legal acts:

$$A_i = \{A_{i1}, A_{i2}, \ldots, A_{iM}\} \tag{1.2}$$

where (A_{ij}) represents a legal act (e.g., sections of the penal code, specific statutes)

We can represent this as a function:

$$f(D_i) \approx A_{ij} \tag{1.3}$$

where (F) predicts the set of legal acts for a given case study.

For example, given a new case study (D_{new}) which is not in the training dataset, we will predict the set of legal acts after running it through a model:

$$Model.f(D_{new}) = A_{predicted1}, A_{predicted2}, \ldots \tag{1.4}$$

This will help open the door to machine learning research in the legal sector so that others can then use this dataset for research purposes by modifying it, adding upon it or by testing with different models to achieve better results.

1.4 Aims and Objectives

The head objective of our study is to use data augmentation to create a machine learning dataset for criminal act and court classification using verified criminal case studies extracted from Manupatra which are within the context of Bangladesh. This paper will also help determine the accuracy of the augmented dataset by testing models such as BERT, DistilBERT, GPT-2, GPT-3, XLNet and Mamba after data processing once for 2000 case studies and then for 3000 case studies.

Chapter 2

Literature Review

Now, we analyze a few of the research literature done in our area of study- In paper [14], The authors show research on legal text analysis so that they can identify the proper sections of laws which are applied in the cases. It is quite similar to our current study but their approach to collecting data is quite interesting as they do not have an available source to transfer the data in simple text or csv. So, the authors propose a web scraping technique where it will use the BeautifulSoup library which will use regular expressions to go through different HTML points and it can also build a parse tree that gives a clean method for going through and altering the corpus. They then extracted the text from their case descriptions and then identified their appropriate law sections from the given judgment and lastly mapped the case into sections. Their raw dataset had 1000 cases and 306 dimensions which had a large distribution of different law sections. The authors proposed to filter the dataset so that they can get rid of the outliers; they found that a setting of 50 thresholds of cases per law section, which is 5 percent of the appropriate cases, qualified 50 sections. They also found that setting a threshold of 2.5 percent law sections per case qualified 736 cases. They then combined the two filtered segments and prepared the last dataset. The author also paid extra attention to preprocessing the data since the text data were web-scraped which may have caused noises due to missing html tags, misspelled words and non-ASCII characters. They also removed stop words and cleared punctuation and converted everything into lower case. Their use of POS feature using the NLTK library to identify helped them return POS tagged set of pairs where each consisted a tokenized word and correlated POS tag along with applied chunking which were used to avoid tags such as preposition, adjective, conjunction, pronoun, adverb, list maker, noun, verb and foreign word and lastly they did stemming which removed morphological affixes words. After pipelining they compared various ML models like SVM, Naïve bayes, logistic regression, decision tree and Convolutional neural network. The authors took precaution to combat the difficulty of misplacing context in Bag of Word model which was because of getting rid of the word sequence in a sentence, accordingly they put in document frequency (DF) to not consider any word that comes up too much. The authors used word embedding for the CNN model so that the text would be in a more dense representation because of this the author chose to test the CNN classifier on it's own rather than with the other classifiers as word embedding models work differently, although they did see that other TfIdf with machine learning executed better with the dataset than word embedding included CNN. The author further mentioned that

the observed results showed certain law articles were easier to predict because of the existence of some special words, considering articles that did not have such words were proportionately harder to predict. Nonetheless, they showed that the SVM had the best performance with 97.29 percent accuracy and decision tree showed the worst performance with 73.24 percent accuracy. Other than that, Logistic regression reached an average accuracy of 88.45 percent, Naïve Bayes had 87.28 percent and lastly CNN with word embedding at 73.50 percent which is only 0.26 percent better than decision trees' worst result. According to the authors, the reason for SVM's better performance could be due to the great number of features present which were created using N-gram models. At the end, the authors mentioned that the reason they did not test further advanced ML models like long short term memory (LSTM) was because sequence modeling along with temporal dependency among words were not thought about. Which takes us to our next paper.

In paper [10], the authors have used two of the more advanced existing ML models and have modified them accordingly to predict legal judgment using an English legal judgment prediction dataset built by them, which consists of cases. They want to predict the court's end result, basically predicting the judgment block, while we are trying to predict the acts which were involved using our own dataset of cases. They boast the size of their dataset which has around 11.5k cases and also provide access to the raw text data which we also want to do. They mostly adhered to three tasks, which include binary classification, multi-label classification and case importance detection. They further claim to have done the first study based on data anonymization to see if the legal prediction models are one sided to certain words and demographics, while also as an extra product, they proposed a hierarchical version of BERT which solves BERT's length issue and we are keen on trying it out in our own study. They then divided the cases in a manner we find interesting as they used the case years. The authors' training and development sets contained the cases from the year 1959 all the way to 2013, and the test set contained cases from 2014 through 2018. They made sure their training and development set were balanced so that their data and their models were not biased in any particular case. On their tests, neural models outperformed an SVM which included bag-of-words. Neural models are where they used the most interesting approach and it is also the part we are going to focus most on. The first model they tried out was a BiGRU-Att model which is just using a BIGRU model with self-attention where the facts of a case are strongly matched into word sequence. The attention part is better explained in paper [3], which we will talk about a bit later. According to the author, the model works using words plotted with embeddings and then goes through a pile of BIGRUs, the embedded word is then calculated as the addition of the following context-aware embeddings which is then weighed by self-attention scores. Lastly, the case embedding then goes through the output layer for binary violation with the use of sigmoid for their tests and multi-label violation with softmax or zero activation for their case importance regression. The second model which they tested with is a slightly modified Hierarchical Attention Network, short for HAN, this model is better explained in paper [4]. The authors modified it so that BIGRU including self-attention can go through the words of each fact which would then create fact embeddings and then BIGRU on secon level including self attention goes through embeddings of fact, creating embedding which was single case and is processed through a output layer like in BIGRU-ATT. The third model they

used was the Label-Wise Attention Network, short for LWAN. According to them, LWAN calls for L attentions, single for every label. This creates L case embeddings per case, this helps each one to specifically predict their parallel label and since it is classified as a multi label model, they used it for multi label violation. The last two models that they used are BERT and HIER-BERT. For BERT they added a linear layer above BERT including a softmax, sigmoid or zero turn on, for multi label violation, binary violation and case importance. The authors were one of the first to highlight an important limitation of BERT, which severely hinders its ability to process long documents which is a norm for legal text processing. BERT has a maximum word limit to go through texts up to 512 wordpieces but their case descriptions can be up to 2.6k words so the data would need to be truncated which reduces the problem and we believe we will also face the same problem. To combat this, they put forward a hierarchical version of BERT which they call HIER-BERT, how this works is that the BERT-BASE model goes through the words of each fact and then produces embeddings. Lastly, fact embeddings are read though a self attention mechanism and produces a single case embedding which is processed through a similar output layer, just like in the prior mentioned HAN model. For their binary violation results they found, HIER-BERT performs the best when the data is Non-Anonymized, where it has a F1 score of 82.0 and HAN performs second best with a F1 score of 80.5 which is slightly better than BIGRU-ATT with a F1 score of 79.5 but when it is Anonymized, HAN takes the lead with an F1 score of 80.2 which is just slightly better than HIER-BERT at 80.1. The Overall worst score belongs to the BERT model which only sits at a F1 score of 17.0 and 25.4 for non-anonymized and anonymized data respectively, the poor performance of the model is because of the aforementioned truncation of case descriptions which HIER-BERT does not face. For the Multi-label Violation results we can see that HIER-BERT performs better overall with a F1 score of 60.0, than models such as BIGRU-ATT and LWAN which have 56.2 and 57.6 respectively. The authors then hypothesize that HAN's factual embeddings process related features of importance from every fact which allows it's GRU on second level to work on factual embeddings which are sequential that are at the moment used by the fact level mechanism with attention which provided a more brief outlook of the whole case and the exact same can be said to HIER-BERT which is reliant on BERT's factual embeddings. On the other hand, BIGRU-ATT works on a long single sequence of strongly matched facts which makes it harder for BIGRU to merge knowledge from assorted and faraway facts. From the given analysis, it explains HAN and HIER-BERT have the best performance overall.

Next, let's look at paper [6] where the authors explain further use of hierarchical RNN models by using them for obligation and prohibition extraction from contracts. They show that a mechanism which has self-attention increases the effectiveness of a BILSTM classifier by letting it focus on suggestive tokens while further introducing a hierarchical BILSTM model which turns every single sentence to an embedding and then continues processing them to classify each sentence. They used heatmap visualization for the attention scores of BILSTM-ATT and that is also how it was done on paper [10] for the HIER-BERT. They conducted research with a dataset consisting of 6,385 training, 1,595 development, and 1,420 test sections which had main body articles of 100 English service agreements selected randomly. The authors used a sentence splitter to preprocess the sections and it produced 31,545 training,

8,036 development and 5,563 test sentences. For their methods, the first model they considered was BILSTM which goes through a single sentence at one time but the sentence is shown by the weighted addition of the unseen states when self attention is added of the BILSTM which is their second method called BILSTM-ATT. As an extension they also used what they call X-BILSTM-ATT where the BILSTM chain is sustained with the tokenized embeddings which include the previous tokens as well. Lastly they used H-BILSTM-ATT which is the hierarchical BILSTM classifier where all the sentences of an entire section is considered and every single sentence is converted to a sentence embedding which is then fed to a second BILSTM. From this we learn that all the data first passes through their respective model and then the data passes through their own models again to create a hierarchical structure and since the data is analyzed twice, the accuracy is always higher. As expected, the authors' experiment revealed that the hierarchical H-BILSTM-ATT outperformed the other three flat models and even the training time was significantly lower than the other models. They concluded that the reason for the three flat methods obtaining especially lower F1 scores in consideration to the H-BILSTM-ATT in nested clauses is because the flat models have no view of the previous sentence. Which is another very important reason for us to use HIER-BERT over the flat BERT model.
In the paper [1], the authors look into the difficulty of binary text classification in the context of legal docket entries. The authors presented text classification problems and found out that the normal ML proceedings use bag- of-words feature which performs badly. Their investigation presented, traditional models such as SVMs had many shortcomings. So they built a simple propositional logic-based classifier that outperformed these traditional models. In this paper, the authors identified a compound effect called "Order re: Summary judgment" (OSJ). it is a legal term that implies that judgment was made by the court without a complete trial. They collected 5,596 docket entries which were hand-labeled by legal experts. Among the docket entries, 1,848 of them were in the OSJ category. They used standard linear SVM for the problem with word-based features as inputs and the model achieved 79.44 percent in F1 score. Later to improve the F1 score they picked the best 100 features with the use of average information gain metric and started the SVM it achieved an F1 score of 83.08 percent. Then with human assistance, they selected features that were critical in tagging a docket entry and trained the SVM which achieved 86.77 percent in F1. To improve the results further they constructed a formula

$$C(D) = \bigvee_{i=1}^{N(D)} \left(\bigwedge_{j=1}^{M_i} L(f_{ij}) \right) \tag{2.1}$$

Here the docket entry is represented by D, followed by N(D) which is represented as the number of sentences, the number of labeled features in the ith sentence is represented by Mi, lastly L() is used for plotting from a feature to its label and lastly C(D) represents the classification function where the true is implied in the docket entry of OSJ occasion. The authors used human selected to 100 features and ran the model and it achieved 85.71 percent in recall 93.45 percent in precision and 89.67 in F1. They concluded that an SVM which is enhanced by hand-picked features underperformed against their logic based classifier.
In paper [3], the authors presented how to analyze legal text and detect contractual obligations and prohibitions. they used BILSTM classifiers and improved it. They

improved the performance of BILTSM classifier by focusing on indicative words. They introduce a hierarchical BILSTM. The dataset they used contains 6,385 training, 1,595 development, and 1,420 test sections from random 100 English service agreements. They were preprocessed by a sentence splitter which produced 31,545 training, 8036 development, and 5,563 test sentences. The models the authors worked on are BILSTM, BILSTM-ATT, X-BILSTM-ATT, and H-BILSTM-ATT. The BILSTM model achieved 90 percent in precision, 88 percent in recall, 88 percent in F1 score, 94 percent in AUC score, the BILSTM-ATT model achieved 90 percent in precision, 88 percent in recall, 89 percent in F1 score, 96 percent in AUC score, the X-BILSTM-ATT model achieved 90 percent in precision, 88 percent in recall, 89 percent in F1 score, 94 percent in AUC score, The H-BILSTM-ATT model achieved 95 percent in precision, 95 percent in recall, 95 percent in F1 score, and 98 percent in AUC score. The H-BILSTM-ATT outperformed the other three models because it considers a sentence at an instance and important details about a sentence can be put further together in its own sentence embedding and engage with the sentence embedding of faraway sentences.

In paper [9], the authors worked on court view generation and for this task, they proposed a sequence-to-sequence model which was label-conditioned with attention. The authors constructed their dataset using the published version of legal documents in China Judgements Online.they trained a model with the descriptions of the charges that can generate rationales for court decisions. Adam algorithm was to used to optimize this model and it uses beam search to generate and select the best possible rational sequence. For the baseline, they used RAND, BM25, MOSES+, NN-S2S, and RAS+. For evaluation, they did automatic evaluation and human judgment. BLEU-4 score and Rouge Scores were adopted for automatic evaluation. Their model outperformed the other baseline and achieved 45.8 in B-4, 70.9 in R-1, 52.5 in R-2, and 67.7 in R-L for automatic evaluation. In human judgment their model achieved 4.93 in fluent, 4.54 in accuracy, and .72 in adoption.

In paper [5], the authors presented a Wikipedia-based approach to develop resources for the legal domain. They took the information available in Wikipedia and connected it with ontologies. They established mapping for the Wordnet and Wikipedia-based YAGO ontology and LKIF ontology. Their corpus was divided 80 percent for training 10 percent for tuning and 10 percent for testing. They used curriculum learning(CL) for Named Entity Recognition and Classification(NERC). For comparison the used Support Vector Machine(SVM), Stanford CRF Classifier, and MLP. Here the neural network classifier outperformed both SVM and Stanford and achieved 95 percent in accuracy, 77 percent is precision, 64 percent in recall, and 68 percent in F1. The authors also presented that when using fewer instances the MLP classifier outperforms other models.

In paper [8], the author presented a large-scale Chinese legal dataset which can be used for judgment prediction named CAIL2018. It had more than 2.6 million criminal cases. The dataset is divided into two parts: fact explanation and correlated judgment result. The result of each case is divided into three terms that is, relevant law articles, charges, and prison terms. To evaluate the task of Legal Judgment Prediction the authors selected three baselines TFIDF+SVM, FastText, and CNN. the authors randomly selected 1,710,856 cases for training and 965,219 cases for testing. For training, they used Adam for optimization. Among the models, the TFIDF+SVM model got the best result average in three different predictions. For

charges prediction, the model achieved 94 percent, 73.9 percent, and 21.4 percent in accuracy, macro-precision, and macro-recall. For relevant articles prediction, it got 92.9 percent, 71.8 percent, and 52.4 percent in accuracy, macro-precision, and macro-recall respectively. Lastly, For terms of penalty prediction, it got 75.4 percent for accuracy, 75.4 percent for macro-precision, and 46.1 percent for macro-recall.
In paper [19], the authors presented a new framework for case outcome prediction named PILOT (Predicting Legal Case Outcome). They also created a up to date dataset which was named ECHR22023 and the data was taken from the European Court of Human Rights (ECHR) database. The model handles legal case prediction through multilabel classification. The step the model follows is, Precedent case fetching, case encoding with fusion of proof, and outcome prediction with Temporal pattern mining. For baseline, the authors chose BERT, HIER-BERT, BERT-LWAN, EPM-base, BERT+CL+kNN, BERT+TemporalAttention, LWDROV2, ChatGPT 5-shots. The authors used four metrics for legal case results they are, micro-F1, micro-Jaccard, micro-PR-AUC, and micro-ROC-AUC. PILOT model outperformed every other model and achieved 71.5 percent, with an uncertainty of ± 0.8 percent in F1, 55.7 percent, with an uncertainty of ± 1.0 percent in Jaccard, 54.3 percent, with an uncertainty of ± 1.4 percent in PR-AUC, and 83.1 percent, with an uncertainty of ± 0.7 percent in ROC-AUC.
In paper [12], The authors proposed a multi-channel alternative neural network model named MANN. It trains using past judgment entries and executes the built in LJP task in a merged framework. The authors took the first dataset from CAIL2018 filtered it and made RCAIL. The authors also collected judgment documents and criminal cases from China judgment online and PKU fabao and constructed three datasets which were named CJO-s, CJO-L, and PKU. The dataset is divided into 3 subtask law article, charge prediction, prediction, and prison term prediction. To run the dataset the authors used ADAM for optimization. The authors choose six baselines to evaluate MANN they are, TF-IDF+SVM, CNN, GRU, Bi-GRU, HAN, and TOPJUDGE. For evaluation metrics, they chose macro Area Under Curve(AUC), macro Precision, macro Recall, and macro F1-score. After training all four datasets MANN model outperformed the other six models. MANN enhances F1 by 2.99 percent, 4.55 percent on average for all the subtasks.
In paper[7], the authors introduced a model named Bidirectional Encoder Representation (BERT). The model was created to pretrain deep bidirectional representation using unlabeled text. For fine-tuning, all of the pre-training data are fine tuned through labeled data. The authors used WordPiece embeddings with 30,000 token vocabulary. The authors pre-trained BERT with two tasks which were unattended and they are, Masked LM, and Next Sentence Prediction(NSP). For fine-tuning the authors plugged in the duty-fixed inputs and outputs into BERT which is then fine-tuned, end-to-end. BERT achieved amazing results on several benchmark NLP task which include GLUE, SQuAD and SWAG.
in paper[13], the The authors used a unified text-to-text transformer approach for the T5 LLM model. They used this model for Legal Judgement Prediction (LJP), they predicted the judgment using legal case facts. The T5 model uses a single Transformer for all sub-tasks. The dependency relation between these sub-tasks uses an autoregressive decoder and the order of these sub-tasks can be adjusted by minimizing error propagation. The author used a dataset from China Judgment Online, which covers many legal articles and they selected 225,843 criminal judgment

documents. They randomly selected 12,810 cases for the test set, 122,2634 cases for the validation set, and the rest were saved in the use of training. When they ran the T5 model they set 128 as their batch size and epoch 30 for all models. For comparing baseline they used LSTM, CNN, FactLaw, TopJudge, and LBERT. The T5 model did significantly better than all other models. It also outperformed LBERT in macro f-1 metric by 8-30 percent. By using dependent T5 there results improved further.
In paper[21], the authors created a comprehensive dataset designed to facilitate the development of artificial intelligence models for legal text processing. The dataset consists of 258 146 cases. The AI models used on the dataset can summarize legal text, using historical data, models can predict potential outcomes, the model can give legal guidance and automated contract analysis They used End-to-End RoBERTa, RoBERTa Pipeline, GPT-3.5, GPT-4 models. The results for these models shows that while transformer models like RoBERTa can classify legal text with high accuracy, generative models like GPT-3.5 and GPT-4 are more effective for generating meaningful legal text with higher similarity to reference material.
In the paper[15], the authors gave an overview of the field of Legal Judgment Prediction (LJP), focusing on datasets, evaluation metrics, models, and the challenges faced in this domain. The authors used 31 LJP datasets across six languages, giving their construction processes and categorizing them based on three attributes they are language, jurisdiction, and prediction task. Among the 31 LJP datasets, notable ones are CAIL2018, ECHR, ECHR-CASES, and Swiss-Judgment-Prediction. The authors used 14 evaluation metrics tailored for different LJP tasks. They used many transformer-based models such as BERT and its variants, BIGRU, HAN, etc. They compared and discussed the most recent benchmark datasets and experimental results of different methods. Based on their observations, they propose new recommendations for future datasets.
In the paper[16], the authors addressed the challenges of pretraining large language models (LLMs) on datasets that may contain biased, obscene, copyrighted, or private information. They created the "Pile of Law," a 256GB open-source dataset comprising English-language legal and administrative texts. It contains diverse legal documents, including court opinions, contracts, administrative rules, and legislative records. The paper examines how governments and legal systems have managed the balance between transparency and the protection of sensitive information. The authors demonstrate how machine learning models can be trained to identify and filter sensitive content, creating a way for more context-aware data preprocessing techniques. By integrating legal standards into the data curation process and offering a rich dataset for experimentation, this paper contributes to the development of more ethical and effective language models in the legal domain.
In paper[18], the authors investigated how variations in dataset characteristics affect the performance of machine learning models in legal applications. The authors utilized three publicly available legal datasets: the European Court of Human Rights (ECHR) Dataset, the United States Federal Court Decisions (USFCD) Dataset, and the Contractual Obligation Dataset. The authors changed the dataset characteristics, such as size, train/test splits, and labeling accuracy, to observe their effects on model performance. The authors used a deep learning classifier based on a neural network architecture that is suitable for natural language processing tasks.
In paper[20], the authors explored the challenges and solutions for summarizing legal documents in multiple languages, especially when there is a lack of labeled

datasets. They used both extractive summarization and abstractive summarization. They experimented with the Australian Legal Case Reports dataset, evaluating their model's effectiveness in summarizing legal case reports in several languages. They used BERT and T5 as their models. For the evaluation metrics, they used ROUGE (Recall-Oriented Understudy for Gisting Evaluation), BLEU (Bilingual Evaluation Understudy), METEOR (Metric for Evaluation of Translation with Explicit ORdering) and BERTScore. Their approach outperformed state-of-the-art extractive summarization techniques on the Australian Legal Case Reports dataset and set a new baseline for abstractive summarization.

Chapter 3

Dataset

3.1 Augmented Data Structure

This is a demonstration of the augmented data structure using one of the verified case studies extracted from the Manupatra database with the subject 'Criminal Law' with 22 features which includes, case id, case name, court, decision date, judges, appellant counsel, respondent counsel, subject, relevant section, acts involved, case note and judgement (1-10).

Field	**Value**
case_id	BD_SC_CR_1268_1992
case_name	Sahajahan and Others Vs. The State
court	Supreme Court of Bangladesh (High Court Division)
decision_date	19.06.2012
judges	Mohammad Marzi-ul-Huq, Md. Ruhul Quddus
appellant_counsel	No One Appears
respondent_counsel	Mrs. Syeda Rabia Begum, A.A.G.
subject	Criminal Law
relevant_section	CODE OF CRIMINAL PROCEDURE, 1898...
acts_involved	Arms Act, 1878, Sections: 19(a); 19A Code of Criminal Procedure, 1898 (CrPC), Sections: 339C, 339C(4), 339D, 439; Explosive Substances Act, 1908, Sections: 3, 4;
case_note	Criminal - Validity of Order - Sections 339C, 339C(4) and 339D of Criminal Procedure Code, (Cr.P.C.) - Rule at instance of accused-Petitioners was issued on application calling in question legality of order passed by Special Tribunal. This Rule at the instance of the accused-petitioners was issued on an application under section 439 of the Code of Criminal Procedure calling in question the legality of order dated 19.7.1992 passed by Special Tribunal...
judgement_1	This Rule at the instance of the accused-petitioners was issued on an application under section 439 of the Code of Criminal Procedure calling in question the legality of order dated 19.7.1992 passed by Special Tribunal...

Field	Value
judgement_2	...
judgement_3	...
judgement_4	...
judgement_5	...
judgement_6	...
judgement_7	...
judgement_8	...
judgement_9	...
judgement_10	...

In the Data, the acts involved had no certain one-off pattern that we could follow to split the unique acts, sections, and the year of its inception. This is important as later on we want to implement multi-label one hot encoding for our classification task. We had to manually go over 3000 case studies and construct a pattern for the acts involved. The constructed pattern looked something like this:

"Arms Act, 1878, Sections: 19(a);"

where each act is "act name, year of inception, section: string;" and multiple acts can be split using the ';' to form individual acts. We proceeded to clean the relevent section column using the same method.

3.1.1 Court Division

The augmented dataset contains 3000 unique criminal case studies. Those case studies are with in 2 unique courts which are the Supreme Court of Bangladesh (High Court Division) and Supreme Court of Bangladesh (Appellate Division). There are in total 2223 High Court and 779 Appellate division case studies as shown in figure 4.3. That makes it a split of 74.06% High Court division and 25.94% Appellate division which is shown in 3.1.

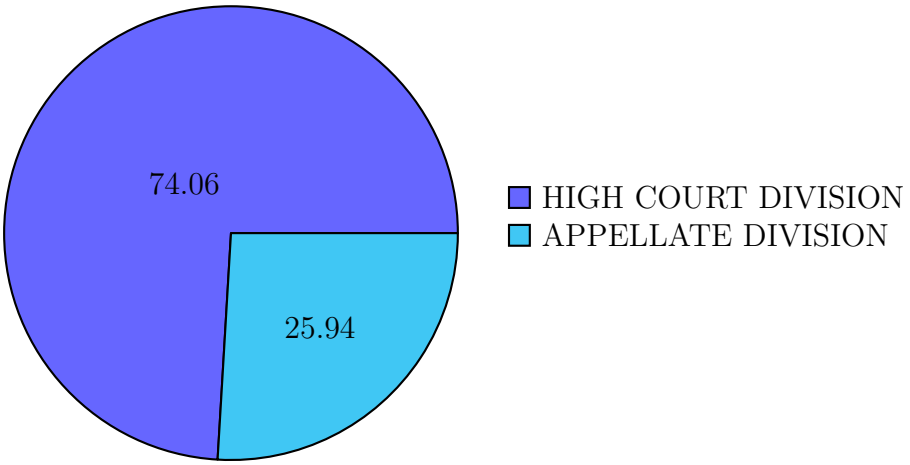

Figure 3.1: Percentile Pie Chart of Court Counts

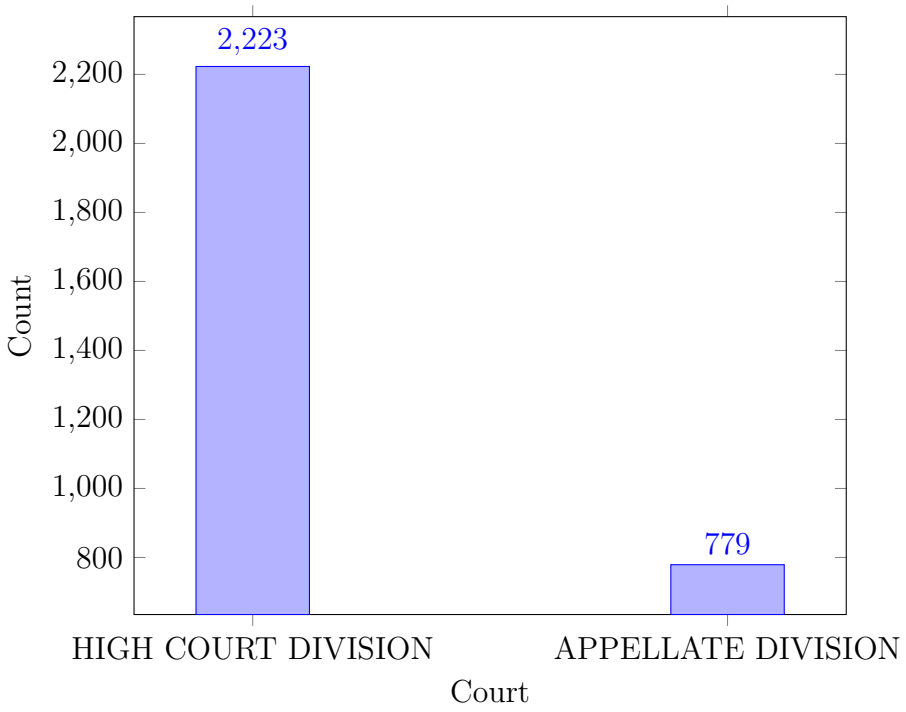

Figure 3.2: Bar Chart of Court Counts

3.1.2 Decision Dates

The decision dates range from 1970 to 2024. There are in total 2228 case studies that has decision dates that range from 2011 to 2024, 423 cases studies in the range from 2001 to 2010, 303 case studies in the range from 1991 to 2000 and 48 case studies in the range from 1970 to 1990 and their appearances are shown in figure 3.4. That makes it a split of 74.22% with in the range from 2011 to 2024, 14.09% from 2001 to 2010, 10.09% from 1991 to 2000 and 1.60% from 1970 to 1990 as shown in figure 3.3

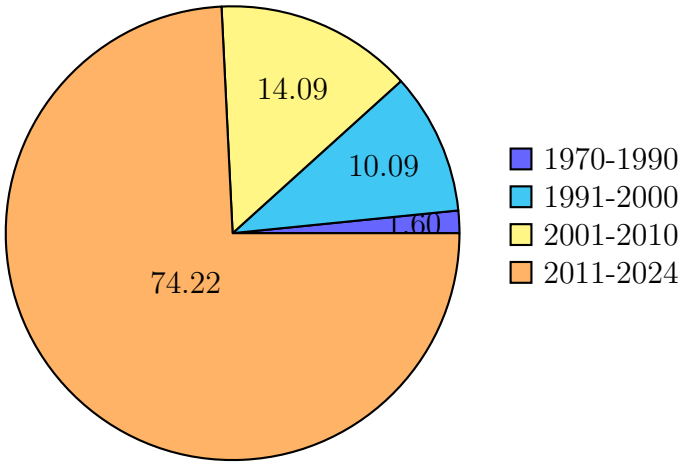

Figure 3.3: Percentile Pie Chart of Decision Dates by Year Range

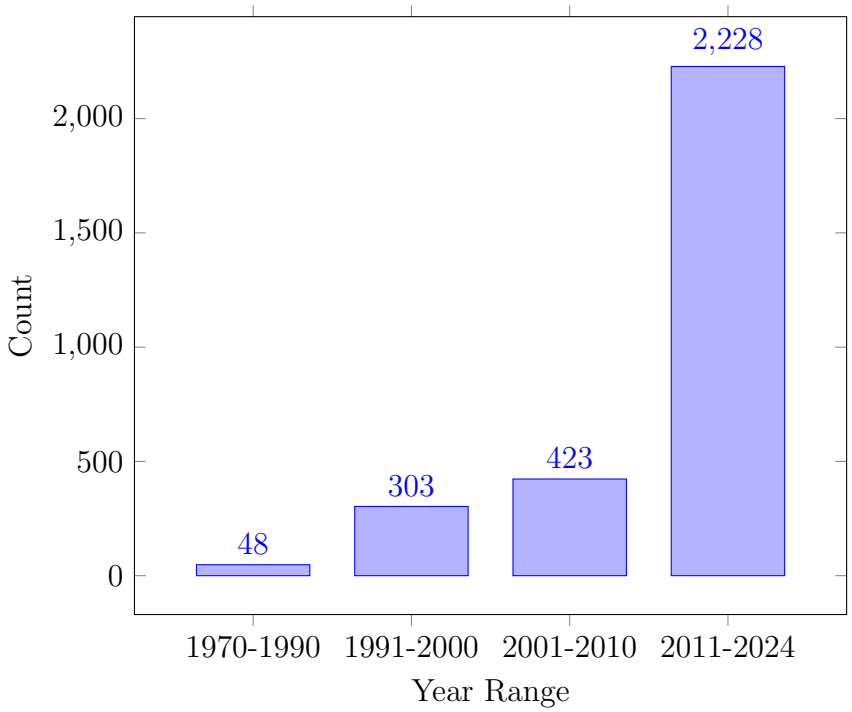

Figure 3.4: Bar Chart of Decision Dates by Year Range

3.1.3 Acts Involved

There are over 80 unique act names and over 3000 unique acts including sections. The most prominent act names include Penal code which occur 8190 times, Code of Criminal Procedure occur 6289 times, Negotiable Instruments Act occur 1032 times, Evidence Act occur 1000, Special Powers Act occur 511 times and so on. Some of their occurrences are visualized in figure 3.5.

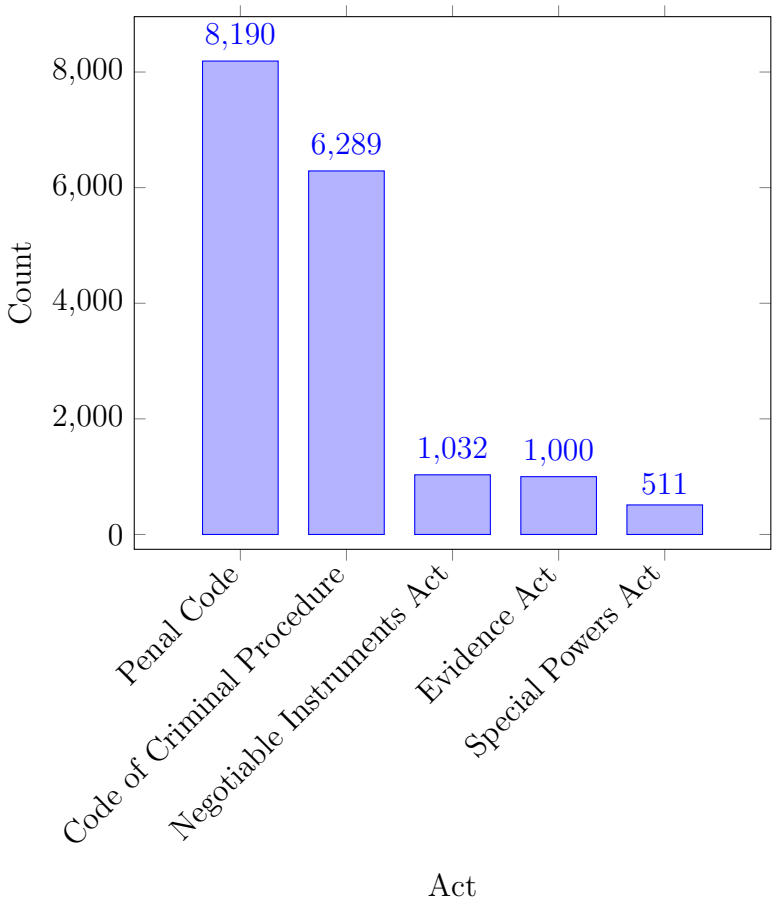

Figure 3.5: Bar Chart of Act Counts

Chapter 4

Methodology

After reviewing multiple papers, we feel that the best approach would be to use the augmented data after pre-processing to examine the effectiveness while also benchmarking the dataset based on classification using models such as BERT and DistilBERT to compare the computational power needed, GPT-2 and GPT-3 for testing a model not traditional for classification and comparing their results, XLNet and Mamba to find results for 2000 case studies and 3000 case studies and later compare. These are few of the well known pre-trained models we want to use as all chosen models are great for processing large documents which is the norm for legal case studies.

4.1 Input Data

As previously mentioned the dataset contains 22 columns which include case id, case name, court, decision date, judges, appellant counsel, respondent counsel, subject, relevant section, acts involved, case note and judgement (1-10).We want to split the dataset for Binary Classification and Multi-Label classification where we classify the judgment into 2 courts, High Court and Appellate Court for Binary classification and classify the judgement into labels based on acts involved for Multi-Label classification, both based on One-hot-encoded data. The graph for it is given in figure 4.1

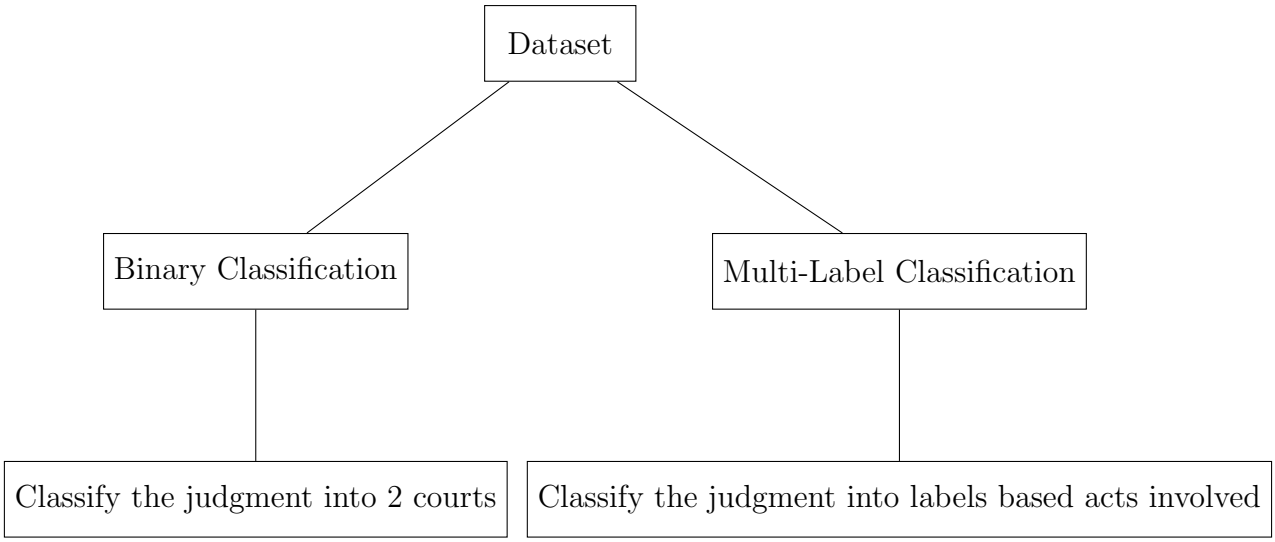

Figure 4.1: Input Data

4.2 Implementation and Results

In this part, we will use transformer models to see their results and briefly understand how they work. For our dataset, we have applied 6 transformer based models, which are BERT, DistilBERT, GPT-2, GPT-3, XLNet and Mamba. The results of the multi-label classification were observed through their hamming score and the results of the binary classification were observed through their accuracy for 2000 case studies and 3000 casestudies. Lastly, we did a comparative analysis between all the models used.

4.3 Preprocessing

To classify the acts involved we implemented multi-label one-hot encoding, where the concatenated judgement is the text data and the labels are unique acts involved. For our court classification, we decided to do a simple binary classification using one-hot encoding where the concatenated judgement is again used as text data and the court names as classes. For the models, we set the proper perameters and trained them accordingly. The full dataset had 3000 data, it was devided in to train data which had 2100 data, validation and test data which had 300 data each and full test data which had 600 data and for 2000 data, we devided it into train data which had 1600 data, validation and test data which had 200 data each and full test data which had 400 data.

4.3.1 Feature Engineering

We wanted to implement multi-label one hot encoding where the concatenated judgements would be the text data and the unique acts will be labels. Unfortunately, there are over 80 unique criminal acts for only 3000 collected case studies. It would have been a complex task to classify the acts with such limitations, so after some testing, we decided to only keep the unique acts that occurred over 150 times which

reduced the labels to only 10. The 10 labels and their appearance percentage are as follows, Code of Criminal Procedure appeared 28.65%, Penal Code appeared 27.1%, Constitution Of The People's Republic Of Bangladesh appeared 4.27%, Evidence Act appeared 3.31%, Negotiable Instruments Act appeared 3.54%, Special Powers Act appeared 1.68%, Criminal Law Amendment Act appeared 1.48%, Prevention Of Corruption Act appeared 1.49%, International Crimes (tribunals) Act appeared 0.59%, Arms Act appeared 0.59%. The visualization is shown in figure 4.2.

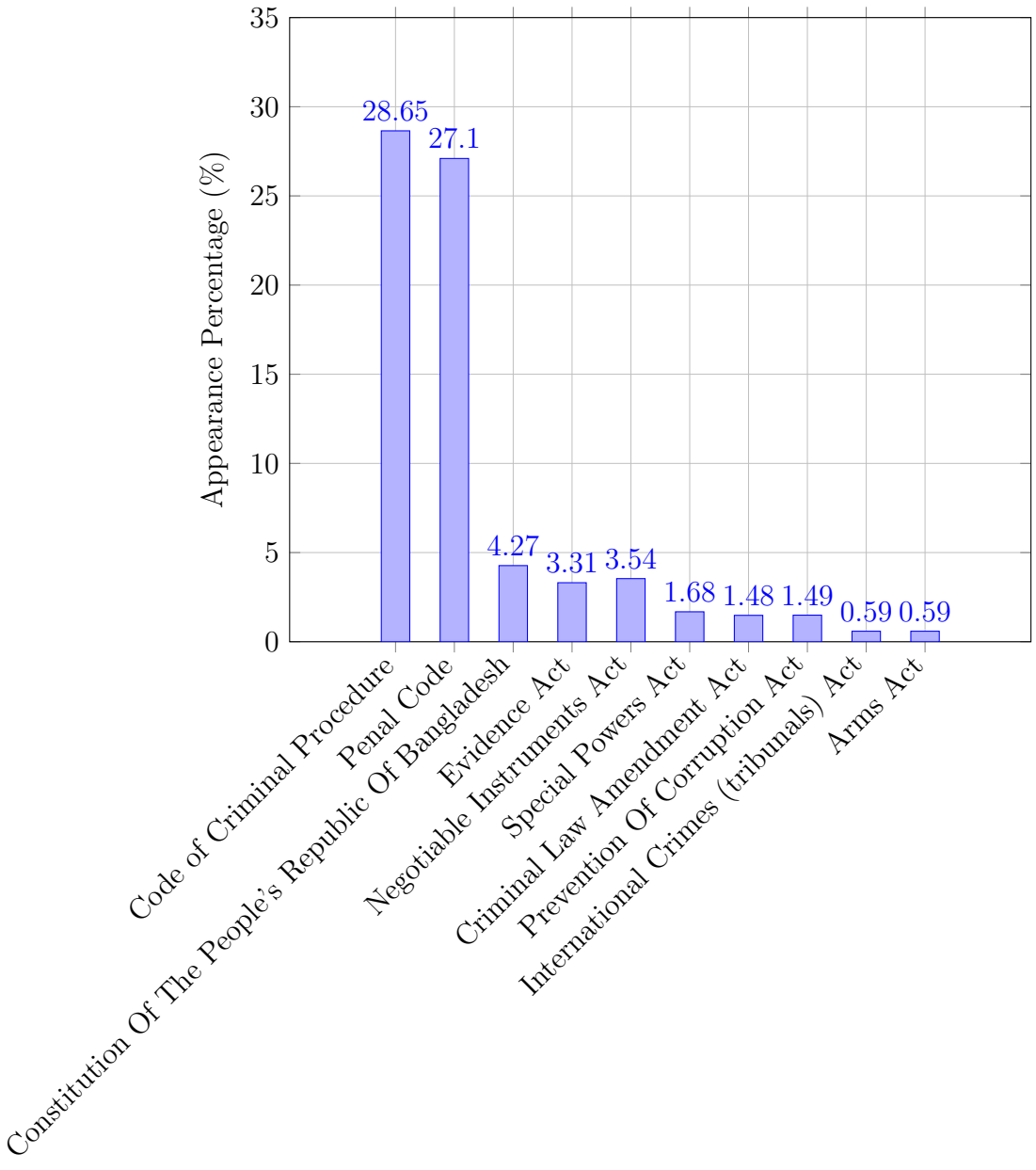

Figure 4.2: Appearance Percentage of Selected Legal Labels (3000 case studies)

We also labeled the 2 unique court names for one-hot encoding where the text data are the judgements. The 2 labels and their appearance percentage are as follows, SUPREME COURT OF BANGLADESH (HIGH COURT DIVISION) appeared

61.05% and SUPREME COURT OF BANGLADESH (APPELLATE DIVISION) appeared 38.95%. The visualization is shown in figure 4.3.

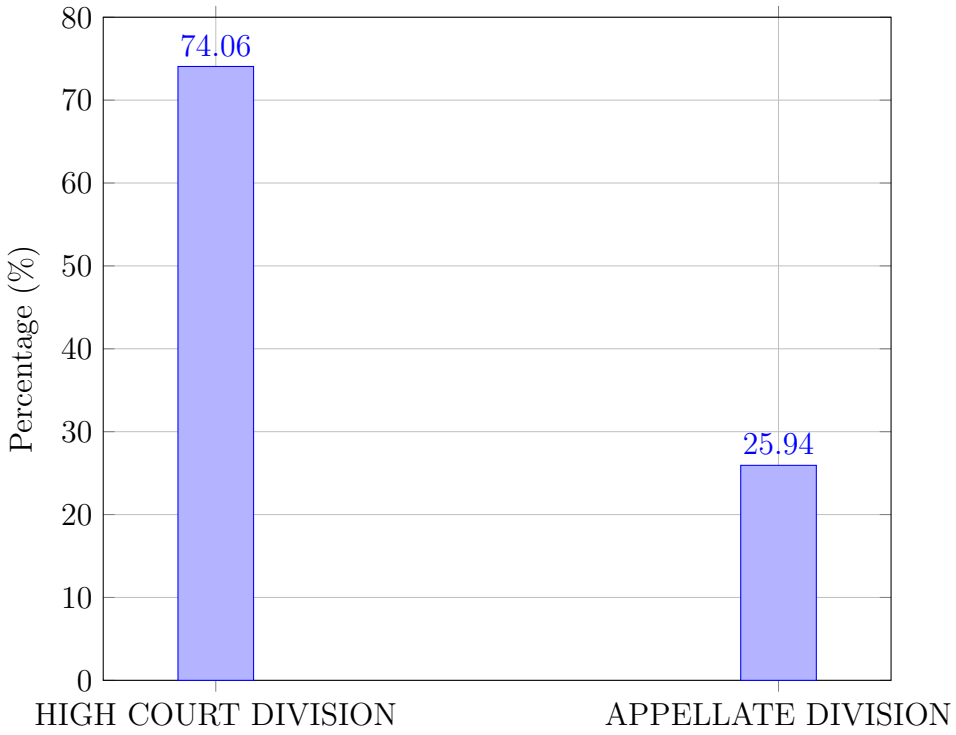

Figure 4.3: Percentile Bar Chart of Court Counts (3000 case studies)

For 2000 case studies there are over 60 unique criminal acts. We decided to do the same and only keep the unique acts that occurred over 150 times which reduced the labels to only 9. The 9 labels and their appearance percentage are as follows, Code of Criminal Procedure appeared 30.14%, Penal Code appeared 28.97%, Constitution Of The People's Republic Of Bangladesh appeared 5.00%, Evidence Act appeared 3.48%, Negotiable Instruments Act appeared 2.59%, Special Powers Act appeared 1.97%, Criminal Law Amendment Act appeared 1.92%, Prevention Of Corruption Act appeared 1.63%, International Crimes (tribunals) Act appeared 0.87%. The visualization is shown in figure 4.4

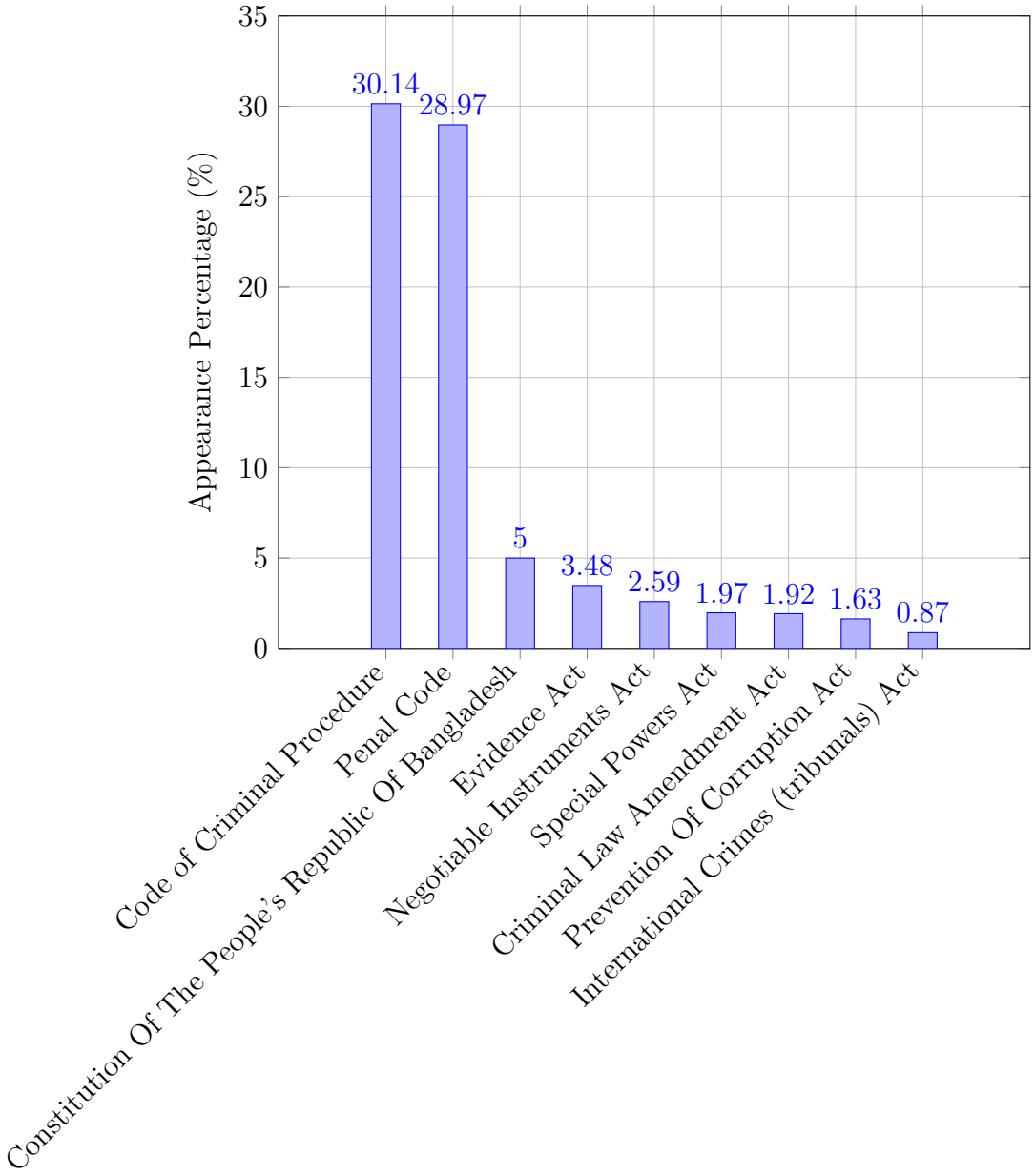

Figure 4.4: Appearance Percentage of Selected Legal Labels (2000 case studies)

We also labeled the 2 unique court names for one-hot encoding where the text data are the judgements. The 2 labels and their appearance percentage are as follows, SUPREME COURT OF BANGLADESH (HIGH COURT DIVISION) appeared 61.05% and SUPREME COURT OF BANGLADESH (APPELLATE DIVISION) appeared 38.95%. The visualization is shown in figure 4.5

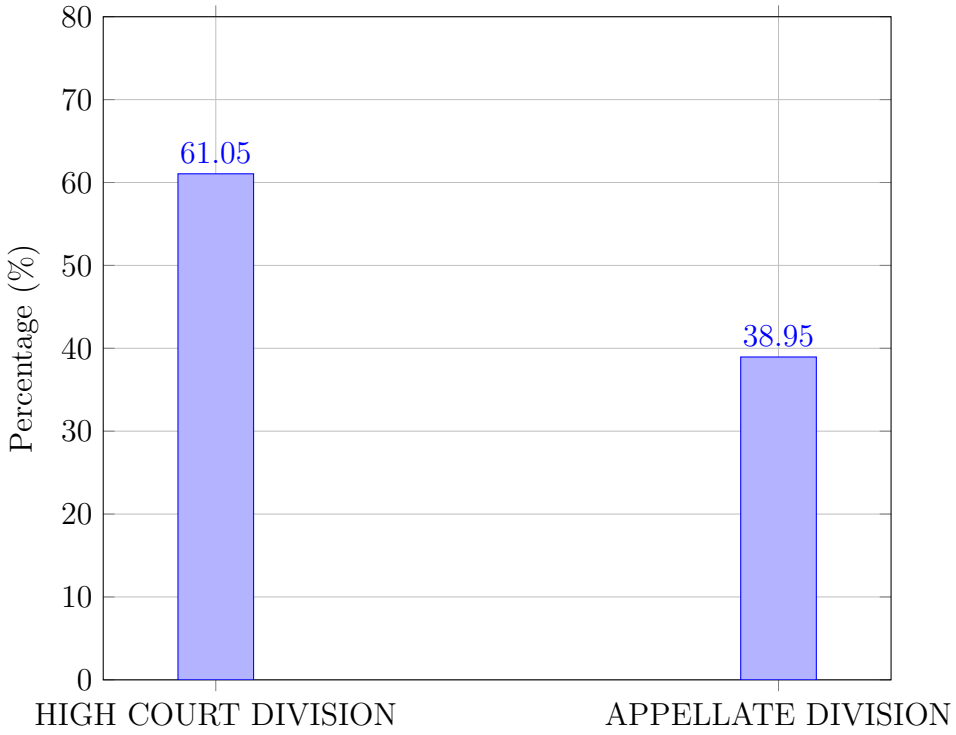

Figure 4.5: Percentile Bar Chart of Court Counts (2000 case studies)

4.3.2 Text Cleaning

We implemented basic text cleaning processes such as removing punctuation, special characters and numbers from the judgement text. We also converetd the judgement text to lowercase to make sure it is uniform and lastly, removed stop words such as ”and”, ”is”, ”the”, etc.

4.4 Model Information

4.4.1 BERT

BERT is an innovative model that has had significant impact on the field of NLP. It goes through text bidirectionally meaning it takes the context from both left to right and right to left simultaneously. This lets it to easily understand the meaning of words based on their surrounding context more effectively. In our implementation, for the multi-label one-hot encoded classification we used the BERT tokenizer to tokenize the data. We set the learning rate to 1×10^{-5} and trained the model for 20 epoch for the 2000 data split and 25 epoch for the 3000 data split after various experiments. We used BCEWithLogitsloss as our loss function and Adam as our optimizer. We then implemented hyper perameter tuning and produced the hamming score for 2000 case studies and 3000 case studies as shown in table 4.1

Model	**Hamming score**
BERT (2000 Data)	0.7321

BERT (3000 Data)	0.8234

Table 4.1: BERT Results (Multi-Label)

In our implementation, for the binary classification, we also used the BERT tokenizer to tokenize the data. We set the learning rate to 1×10^{-5} and trained the model for 20 epoch for the 2000 data split and 25 epoch for the 3000 data split after various experiments. after various experiments. We used BCE as our loss function and Adam as our optimizer.We then implemented hyper perameter tuning and produced the Precision, Recall, F1-Score and Accuracy as shown in table 4.2.

Model	**Precision**	**Recall**	**F1-score**	**Accuracy**
BERT (2000 Data)	0.9923	0.9837	0.9878	0.9941
BERT (3000 Data)	0.9945	0.9982	0.9944	0.9967

Table 4.2: BERT Results (Binary Classification)

4.4.2 DistilBERT

DistilBERT is short for Distilled BERT which is a smaller and faster version of BERT that has 97% of BERT's language comprehensive capacity while being 60% faster and 40% smaller. The model achieves this by using a process called information distillation where a smaller model learns to mimic a larger model, which in this case is BERT. We will be using this to compare the computational requirements. In our implementation, for the multi-label one-hot encoded classification we used the DistilBERT tokenizer to tokenize the data. We set the learning rate to 1×10^{-5} and trained the model for 20 epoch for the 2000 data split and 25 epoch for the 3000 data split after various experiments. We used BCEWithLogitsloss as our loss function and Adam as our optimizer. We then implemented hyper perameter tuning and produced the hamming score for 2000 case studies and 3000 case studies as shown in table 4.3

Model	**Hamming score**
DistilBERT (2000 Data)	0.7276
DistilBERT (3000 Data)	0.8049

Table 4.3: DistilBERT Results (Multi-Label)

In our implementation, for the binary classification, we also used the Distil BERT tokenizer to tokenize the data. We set the learning rate to 1×10^{-5} and trained the model for for 20 epoch for the 2000 data split and 25 epoch for the 3000 data split after various experiments. after various experiments. We used BCE as our loss function and Adam as our optimizer.We then implemented hyper perameter tuning and produced the Precision, Recall, F1-Score and Accuracy as shown in table 4.5.

Model	**Precision**	**Recall**	**F1-score**	**Accuracy**
DistilBERT (2000 Data)	0.9873	0.9825	0.9858	0.9853
DistilBERT (3000 Data)	0.9926	0.9964	0.9942	0.9875

Table 4.4: DistilBERT Results (Binary Classification)

4.4.3 GPT-2

GPT-2 focuses on generating text. It uses a unidirectional transformer decoder to predict the next word in a sequence based on preceding words. We will be using this model to compare the performance of our dataset on models not well suited for Multi-Label classification. In our implementation, for the multi-label one-hot encoded classification we used the GPT-2 tokenizer to tokenize the data. We set the learning rate to 1×10^{-5} and trained the model for 20 epoch for the 2000 data split and 25 epoch for the 3000 data split after various experiments. We used BCEWithLogitsloss as our loss function and Adam as our optimizer. We then implemented hyper perameter tuning and produced the hamming score for 2000 case studies and 3000 case studies as shown in table 4.5

Model	**Hamming score**
GPT-2 (2000 Data)	0.6236
GPT-2 (3000 Data)	0.6029

Table 4.5: GPT-2 Results (Multi-Label)

4.4.4 GPT-3

GPT-3 excels in a wide range of NLP tasks which include text completion, translation, summarization and more. Just as GPT-2, GPT-3 is also not as well suited for multi-label classification tasks due to its unidirectional nature. We will use GPT-3 to compare it's performance on our dataset and evaluate the results compared to GPT-2. In our implementation, for the multi-label one-hot encoded classification we used the GPT-3 tokenizer to tokenize the data. We set the learning rate to 1×10^{-5} and trained the model for 20 epoch for the 2000 data split and 25 epoch for the 3000 data split after various experiments. We used BCEWithLogitsloss as our loss function and Adam as our optimizer. We then implemented hyper perameter tuning and produced the hamming score for 2000 case studies and 3000 case studies as shown in table 4.6

Model	**Hamming score**
GPT-3 (2000 Data)	0.6527
GPT-3 (3000 Data)	0.6673

Table 4.6: GPT-3 Results (Multi-Label)

4.4.5 XLNet

XLNet is a generalized autoregressive pretraining method for language understanding tasks. It uses a permutation-based training objective, which allows it to capture bidirectional context like BERT but without the limitations of the masked language modeling approach. XLNet also incorporates ideas from Transformer-XL, enabling it to handle longer sequences of text efficiently by maintaining state information between segments. This makes it ideal for our classification task with long texts. In our implementation, for the multi-label one-hot encoded classification we used the XLNet tokenizer to tokenize the data. We set the learning rate to 1×10^{-5} and trained the model for 20 epoch for the 2000 data split and 25 epoch for the 3000 data

split after various experiments. We used BCEWithLogitsloss as our loss function and Adam as our optimizer. We then implemented hyper perameter tuning and produced the hamming score for 2000 case studies and 3000 case studies as shown in table 4.7

Model	**Hamming score**
XLNet (2000 Data)	0.7146
XLNet (3000 Data)	0.8222

Table 4.7: XLNet Results (Multi-Label)

4.4.6 Mamba

Mamba is a model designed specifically for efficient handling of long-text sequences. It uses bi-directional transformer architecture allowing it to understand contextual information from both directions within a sentence. Mamba also incorporates memory-augmentated attention and hierarchical processing to manage longer sequences better. This makes Mamba the most suitable for tasks such as multi-label classification. By benchmarking using Mamba, we want to compare its performance on our dataset with other models like BERT and XLNet. In our implementation, for the multi-label one-hot encoded classification we used the Mamba tokenizer to tokenize the data. We set the learning rate to 1×10^{-5} and trained the model for 20 epoch for the 2000 data split and 25 epoch for the 3000 data split after various experiments. We used BCEWithLogitsloss as our loss function and Adam as our optimizer. We then implemented hyper perameter tuning and produced the hamming score for 2000 case studies and 3000 case studies as shown in table 4.8

Model	**Hamming score**
Mamba (2000 Data)	0.7983
Mamba (3000 Data)	0.8452

Table 4.8: Mamba Results (Multi-Label)

4.5 Result from experiments

The 6 transformer models we used on the 2000 data split for multi-label classification, among them BERT, DistilBERT, XLNet showed similar hamming scores with 0.7321, 0.7276 and 0.7146 respectively. On the other hand, GPT-2 and GPT-3 suffered with hamming scores of 0.6236 and 0.6527 respectively. GPT-3 performed slightly better than GPT-2 but out of all the models, Mamba performed the best with a hamming score of 0.7983 as shown in table 4.9

Model	**Hamming score**
BERT (2000 Data)	0.7321
DistilBERT (2000 Data)	0.7276
GPT-2 (2000 Data)	0.6236
GPT-3 (2000 Data)	0.6527
XLNet (2000 Data)	0.7146

Model	Hamming score
Mamba (2000 Data)	0.7983

Table 4.9: Model Results(Multi-Label) (2000 Data)

For the 2000 data split binary classification, Both BERT and DistilBERT showed significant accuracy. This was also expected as there were only 2 classes and 2000 data with large sums of text data which was enough for the models. BERT performed slightly better than DistilBERT as shown in the following table 4.10.

Model	Precision	Recall	F1-score	Accuracy
BERT (2000 Data)	0.9923	0.9837	0.9878	0.9941
DistilBERT (2000 Data)	0.9873	0.9825	0.9858	0.9853

Table 4.10: Model Results (Binary Classification) (2000 Data)

The 6 transformer models we used on the 3000 data split for multi-label classification, among them BERT, DistilBERT, XLNet again showed similar hamming scores but had an average of a 0.10 bump in their results, with 0.8234, 0.8049 and 0.8222 respectively. On the other hand, GPT-2 performed worse with the current split and GPT-3 suffered again with hamming scores of 0.6029 and 0.6673 respectively. GPT-3 performed slightly better than GPT-2 but out of all the models, Mamba performed the best with a hamming score of 0.8452 as shown in table 4.11

Model	Hamming score
BERT (3000 Data)	0.8234
DistilBERT (3000 Data)	0.8049
GPT-2 (3000 Data)	0.6029
GPT-3 (3000 Data)	0.6673
XLNet (3000 Data)	0.8222
Mamba (3000 Data)	0.8452

Table 4.11: Model Results(Multi-Label) (3000 Data)

For the 3000 data split binary classification, Both BERT and DistilBERT showed significant accuracy once more. This was again expected beacuse of the same reason as previously, especially with 1000 more data. BERT performed slightly better than DistilBERT as shown in the following table 4.12.

Model	Precision	Recall	F1-score	Accuracy
BERT (3000 Data)	0.9945	0.9982	0.9944	0.9967
DistilBERT (3000 Data)	0.9926	0.9964	0.9942	0.9875

Table 4.12: Model Results (Binary Classification) (3000 Data)

From our experiments, we can see that there has been a noticeable change in perfromance of the models, where each models besides GPT-2 and GPT-3 have an average of a 10% bump in their hamming scores when there is more data. Mamba performed the best in both data splits as it was specifically made for large form text data and is a good fit for multi-label one hot encoding. BERT was not as

far behind and DistilBERT showed almost similar results to BERT, giving us an idea of the computational power required to run our dataset. XLNet performed almost similar to BERT. One of the most surprising results was GPT-2 performing worse with the 3000 data split than with the 2000 data split. It could be because of it's maximum input length and tokenization. We can come to the conclusion that adding more data to the augmentated dataset following our created patterns would improve model performance and future legal machine learning work within the context of Bangladesh will have a baseline benchmark.

Chapter 5

Conclusion

To conclude, our research tries to lower the bar for entry in legal text analysis within the context of Bangladesh. Although existing research has explored various legal predictions and suggestive models in other countries, the unique complexities of Bangladeshi laws need to be explored for better solutions. As Bangladesh moves forward on its digitalization journey, we hope that research in ML and NLP in legal sectors do not lag behind. Our contribution lies in the creation of a criminal legal acts prediction dataset using verified case studies within the Supreme Court of Bangladesh from Manupatra where there are over 10,000 case studies in the chosen subject that we have divided into 22 features total and collected 3000 case studies for our current augmented dataset. Due to our limited time we were not able to increase the size of our dataset further and test further to improve our model performances. We hope to see better results with more data in the future. In summary, our research strives to empower legal practitioners, reduce manual effort and expedite legal analysis. As we unlock the potential of NLP and ML, we invite further exploration and collaboration towards digitalization in Bangladesh's legal sector.

Bibliography

[1] R. Nallapati and C. D. Manning, "Legal docket classification: Where machine learning stumbles," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2008, pp. 438–446.

[2] W. Y. Wang, E. Mayfield, S. Naidu, and J. Dittmar, "Historical analysis of legal opinions with a sparse mixed-effects latent variable model," in *ACL*, 2012, pp. 740–749.

[3] K. Xu, J. Ba, R. Kiros, *et al.*, "Show, attend and tell: Neural image caption generation with visual attention," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2015, pp. 2048–2057.

[4] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, San Diego, California: Association for Computational Linguistics, 2016, pp. 1480–1489. DOI: 10.18653/v1/N16-1174. [Online]. Available: https://aclanthology.org/N16-1174/.

[5] C. Cardellino, M. Teruel, L. Alonso Alemany, and S. Villata, "Legal nerc with ontologies, wikipedia and curriculum learning," in *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2017, pp. 254–259.

[6] I. Chalkidis, I. Androutsopoulos, and A. Michos, "Obligation and prohibition extraction using hierarchical rnns," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018, pp. 254–259.

[7] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018. [Online]. Available: http://arxiv.org/abs/1810.04805.

[8] C. Xiao, H. Zhong, Z. Guo, *et al.*, "CAIL2018: A large-scale legal dataset for judgment prediction," *CoRR*, vol. abs/1807.02478, 2018. [Online]. Available: http://arxiv.org/abs/1807.02478.

[9] H. Ye, X. Jiang, Z. Luo, and W. Chao, "Interpretable charge predictions for criminal cases: Learning to generate court views from fact descriptions," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018, pp. 1854–1864.

[10] I. Chalkidis, I. Androutsopoulos, and N. Aletras, "Neural legal judgment prediction in english," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, 2019, pp. 4317–4323.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2019.

[12] S. Li, H. Zhang, L. Ye, X. Guo, and B. Fang, "Mann: A multichannel attentive neural network for legal judgment prediction," *IEEE Access*, vol. 7, pp. 151 144–151 155, 2019. DOI: 10.1109/ACCESS.2019.2945771.

[13] Y. Huang, X. Shen, C. Li, J. Ge, and B. Luo, *Dependency learning for legal judgment prediction with a unified text-to-text transformer*, 2021. arXiv: 2112.06370 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2112.06370.

[14] S. Sengupta and V. Dave, "Predicting applicable law sections from judicial case reports using legislative text analysis with machine learning," *Journal of Computational Social Science*, 2021. DOI: 10.1007/s42001-021-00135-7.

[15] J. Cui, X. Shen, F. Nie, Z. Wang, J. Wang, and Y. Chen, *A survey on legal judgment prediction: Datasets, metrics, models and challenges*, 2022. arXiv: 2204.04859 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2204.04859.

[16] P. Henderson, M. S. Krass, L. Zheng, *et al.*, *Pile of law: Learning responsible data filtering from the law and a 256gb open-source legal dataset*, 2022. arXiv: 2207.00220 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2207.00220.

[17] M. Hossain, *Digital Transformation and Economic Development in Bangladesh: Rethinking Digitalization Strategies for Leapfrogging*. Springer Nature, 2022.

[18] H. Westermann, J. Savelka, V. R. Walker, K. D. Ashley, and K. Benyekhlef, *Data-centric machine learning in the legal domain*, 2022. arXiv: 2201.06653 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2201.06653.

[19] L. Cao, Z. Wang, C. Xiao, and J. Sun, *Pilot: Legal case outcome prediction with case law*, 2024. arXiv: 2401.15770 [cs.CL].

[20] G. Moro, N. Piscaglia, L. Ragazzi, *et al.*, "Multi-language transfer learning for low-resource legal case summarization," *Artificial Intelligence and Law*, vol. 32, pp. 1111–1139, 2024. DOI: 10.1007/s10506-023-09373-8. [Online]. Available: https://doi.org/10.1007/s10506-023-09373-8.

[21] A. Östling, H. Sargeant, H. Xie, *et al.*, *The cambridge law corpus: A dataset for legal ai research*, 2024. arXiv: 2309.12269 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2309.12269.

[22] *Digital bangladesh to innovative bangladesh: The road to 2041*, https://www.undp.org/bangladesh/blog/digital-bangladesh-innovative-bangladesh-road-2041, December 25, 2021.