Classifying Legal texts/documents using BERT

Problem Statement

Create a system to classify various types of legal documents (e.g., contracts, wills, deeds) and extract specific clauses or information from them.

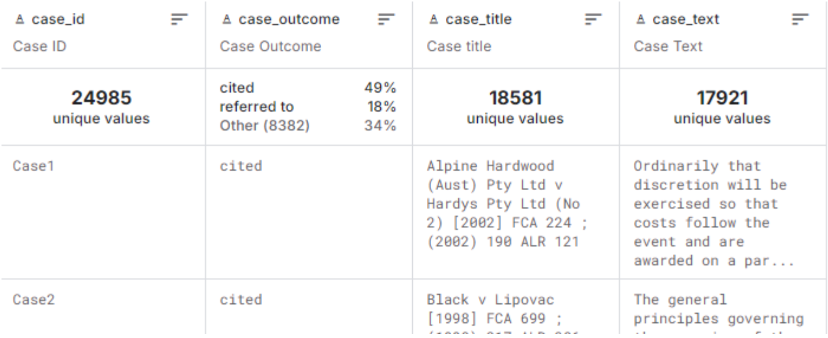
Proposed Solution

Dataset

To begin with, I searched online for some datasets of different legal documents but was unable to find them. Therefore, I decided to a choose a dataset which has already parsed legal text and their categories.

The dataset is obtained from Kaggle: https://www.kaggle.com/datasets/amohankumar/legal-text-classification-dataset/data.

The dataset contains a total of 25000 legal cases in the form of text documents. Each document has been annotated with catchphrases, citations sentences, citation catchphrases, and citation classes. Citation classes indicate the type of treatment given to the cases cited by the present case. The Legal Citation Text Classification dataset is provided in CSV format. The dataset has four columns, namely Case ID, Case Outcome, Case Title, and Case Text. The Case ID column contains a unique identifier for each legal case, the Case Outcome column indicates the outcome of the case, the Case Title column contains the title of the legal case, and the Case Text column contains the text of the legal case.



case_id	case_outcome	case_title	case_text
Case ID	Case Outcome	Case title	Case Text
24985 unique values	cited 49% referred to 18% Other (8382) 34%	18581 unique values	17921 unique values
Case1	cited	Alpine Hardwood (Aust) Pty Ltd v Hardys Pty Ltd (No 2) [2002] FCA 224 ; (2002) 190 ALR 121	Ordinarily that discretion will be exercised so that costs follow the event and are awarded on a par...
Case2	cited	Black v Lipovac [1998] FCA 699 ;	The general principles governing

There are 10 categories of case outcomes namely:

- ‘affirmed',
- 'applied'
- 'approved'
- 'cited'
- 'considered'
- 'discussed'
- 'distinguished'
- 'followed'
- 'referred to’
- 'related

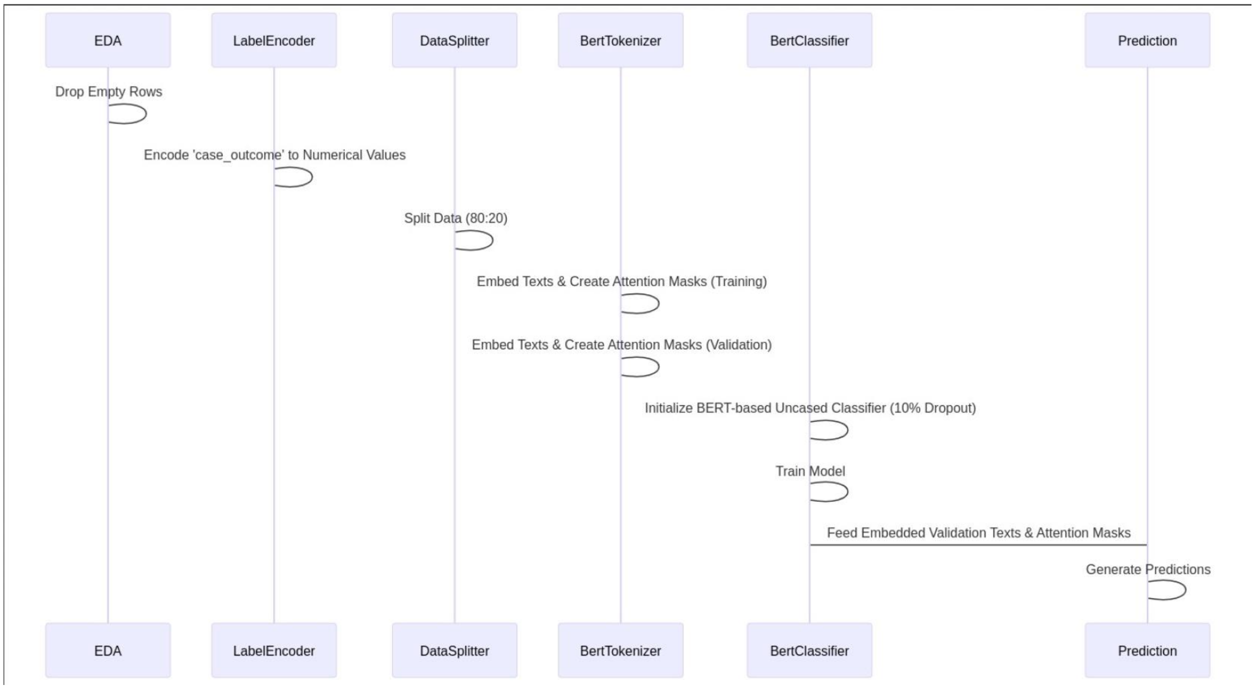
Model Selection & Steps

I have used a BERT model for classifying texts into 10 categories which is a case of multi class text classification. The model is trained for 20 epochs on Google Colab as I did not have the compute power to train more epochs as the session was getting disconnected continuously. The model training configuration is as below:

```
{
  "bert_model_name": "bert-base-uncased",
  "num_classes": 10,
  "max_length": 128,
  "batch_size": 8,
  "num_epochs": 100,
  "learning_rate": 2e-5
 }
```

- Performed Exploratory Data Analysis (EDA) on the raw tabular data of legal texts to drop any empty rows.
- Used a Label Encoder from the scikit-learn library to encode the 'case_outcome' column into numerical values.

- Split the text and labels into 80:20 for training and validation sets, respectively.
- Utilized the BertTokenizer from the Transformer library to embed the texts and create attention masks for the training and validation data.
- Employed a BERT-based uncased classifier model with a 10% dropout rate.
- Fed the embedded validation texts and attention masks into the trained model to generate predictions.


EDA
LabelEncoder
DataSplitter
BertTokenizer
BertClassifier
Prediction
Drop Empty Rows
Encode 'case_outcome' to Numerical Values
Split Data (80:20)
Embed Texts & Create Attention Masks (Training)
Embed Texts & Create Attention Masks (Validation)
Initialize BERT-based Uncased Classifier (10% Dropout)
Train Model
Feed Embedded Validation Texts & Attention Masks
Generate Predictions
EDA
LabelEncoder
DataSplitter
BertTokenizer
BertClassifier
Prediction

Results

For measuring the accuracy of the system, I have used F1-score metric. The F1-score after 20 epochs on validation set (There are total of 25k samples and after dropping empty rows we have 24895 samples out of which 20% are chosen randomly for validation set) is given below:

Epochs	Num Validation Samples	F1-Score
20	4979	0.5743

The accuracy of the system is low due to the compute power constraints and regular disconnecting of session of Google Colab. We can get a better accuracy provided there are no disruptions during training.

Future Scope & Solutions

1. I have not used any NLP algorithm to extract the relevant information from the legal text as such relevant information is either not present or is not uniform in the dataset. In future, if I am given a relevant dataset with information to extract, we could use Regex and NER (Named Entity Recognition) to extract such information

2. Also, there can be multiple solutions with advent of GenAI. We could use Langchain and load the csv file and ask questions to it using LLMs. Another solution could be to fine tune an LLM with this dataset and prompt it to categorize the legal document/text.

3. Development of a model API using Flask / Fastapi of the trained model for downstream application consumption.

Code Usage

Please refer on how to train and test the model with the selected legal text classification data in this repository: https://github.com/naga24/Legal-Text-Classifier-BERT.

Contact

Nagarjun <nagarjun.gururaj@gmail.com>