Rhetorical Roles Prediction of Legal Texts

Parikshith Honnegowda
University of California,
Santa Cruz
phonnego@ucsc.edu

Vijay Chilaka
University of California,
Santa Cruz
vchilaka@ucsc.edu

Xin Zhang
University of California,
Santa Cruz
xzhan445@ucsc.edu

Abstract

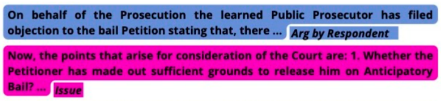
The number of legal cases in every country is increasing in an exponential way, and that too when countries like India are considered with high population this is increasing very drastically. So to overcome some of the problems in legal documents there is a need for developing methods to process and structure the unstructured legal documents. So in this work we are proposing a method to predict a rhetorical roles of a sentence in a legal judgement, which is one of the important task in structuring the unstructured legal documents. This method will further help other tasks that come under legal documents such as search of legal information and summarization. Here we develop a model using neural networks to predict the rhetorical roles. And we have experimented this using multi-layer perceptron, BERT and Long short-term memory neural network models. And the experiments show that the BERT with LSTM model performs better.

1 Introduction

Legal practitioners perform several daily duties linked to natural language processing in the legal field. These tasks could be replaced by machine learning algorithms, however this hasn't happened much yet due to a lack of annotated data. Although there are incredibly vast text bases in the legal field, they are not preprocessed and structured in a way that allows for seamless integration with machine learning methods. In densely populated nations like India, the number of open court cases has increased dramatically. Due to the subjectivity involved in the legal process, it may be difficult to automate the entire judicial pipeline completely; nevertheless, many intermediate tasks can be automated to augment legal practitioners so it can help expedite the system.

Legal documents can be pre-processed with the help of techniques such as natural language processing to organize the data, structure it in such a way and have labels for the structured data so it will be easy for search, retrieval and use it for summarization. This is an important step which will act as a building block for developing Legal Artificial Intelligence solutions. There are mainly three sub-tasks in the area of legal documents: Rhetorical Roles (RR), Legal Named Entity Recognition (L-NER), Court Judgment Prediction with Explanation (CJPE).

In this paper we work on Rhetorical Roles (RR) which deals with predicting the rhetorical roles for the segmented sentences of legal document. The legal documents consists of unstructured sentences and in the given dataset there is also segmented sentences for which we will build a model to predict a rhetorical roles. So this is a task of multi class classification of sentences in a legal document. Each text segment is given a label, such as a Preamble, Fact, Argument by Respondent, Ruling by Lower Court, Argument by Petitioner, Issue, Analysis, Statute, Precedent Relied, Precedent Not Relied, Ratio of the decision, Ruling by Present Court and None. Consider the below figure which depicts an example of rhetorical roles tagged to sentences.



On behalf of the Prosecution the learned Public Prosecutor has filed objection to the bail Petition stating that, there ... *Arg by Respondent*

Now, the points that arise for consideration of the Court are: 1. Whether the Petitioner has made out sufficient grounds to release him on Anticipatory Bail? ... *Issue*

Figure 1: Sentences tagged with rhetorical roles

2 Related work

Multi-class Text Classification on Unbalanced, Sparse and Noisy Data, in this work they have used Multi layer perceptron model and Doc2Vec vectorizer on unbalanced noisy data and have achieved a best micro f-score of .089 (1). Another work is Corpus for Automatic Structuring of Legal Documents,have used SciBERT-HSLN model which can capture longer range dependencies between sentences in a document with F1 score of 78% (2). And Semantic Segmentation of Legal Documents via Rhetorical Roles, have developed a multitask learning-based deep learning model with document rhetorical role label shift as an auxiliary task for segmenting a legal document achieved an f1 score of 70% (3).

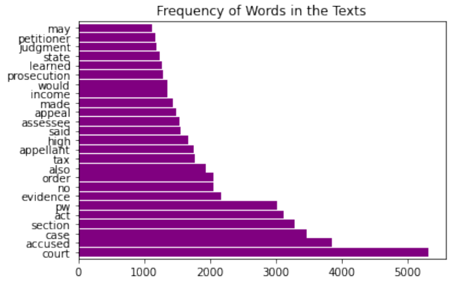


Figure 3: Frequency of Words from all the text in Dataset

3 Dataset

The Dataset (8) provided for this task mainly consists of judgments from the courts and there are totally 247 such judgments given which are segmented in terms of sentences and are labeled a rhetorical role for each sentence. The dataset given is in the format of JSON and it consists of unique ID for each judgment and each judgment ID is a dictionary which consists of judgment text, the annotated array which consists of unique ID for each segmented sentence, start, end, the segmented text and the corresponding label. And the total number of such sentence segments in the train data are totally 28986 in number and the total rhetorical roles used in this dataset are 13 in number. The main two features we have considered for training our model are text and label fields. Below figure 2 depicts the frequency of roles and the figure 3 depicts the frequency of words in the dataset.

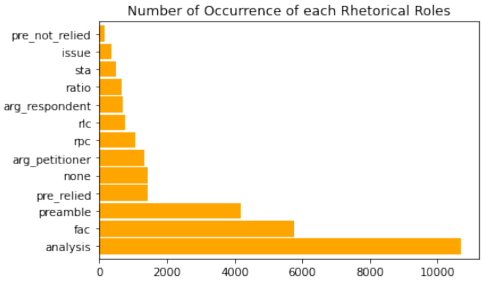


Figure 2: Frequency of Rhetorical Roles in the Dataset

The below section depicts the description of each rhetorical role labels in the given dataset.
Rhetorical Roles

- Preamble - A typical judgement would start with the court name, the details of parties, lawyers and judges' names, Headnotes.
- Facts (FAC) - Describe the sequence of actions that resulted in the lawsuit being filed and how the case developed throughout the legal system.
- Ruling by Lower Court (RLC) - Judgments given by the lower courts (Trial Court, High Court) based on which the present appeal was made (to the Supreme Court or high court).
- Issues (ISSUE) - Some judgements mention the key points on which the verdict needs to be delivered.
- Argument by Petitioner (ARG_PETITIONER) - This refers to precedent cases that petitioner attorneys argue.
- Argument by Respondent (ARG_RESPONDENT) - Arguments by respondents lawyers.
- Analysis (ANALYSIS) - Courts discussion on the evidence,facts presented,prior cases and statutes.
- Statute (STA) - Text in which the court considers established laws, found in a variety of places, including acts, sections, articles, etc.
- Precedent Relied (PRE_RELIED) - Sentences in which the court reviews earlier case materials, arguments, and rulings that it used as a basis for its findings.
- Precedent Not Relied (PRE_NOT_RELIED) - Sentences in which the court examines earlier case materials, arguments, and rulings that weren't considered in its decision-making.
- Ratio of the decision (RATIO) - The main justification for applying any legal theory to the legal issue is stated.
- Ruling by Present Court (RPC) - Final judgment, conclusion, and ruling of the court resulting from the logical/natural conclusion of the reasoning
- None - If a sentence does not belong to any of the above categories.

4 Methodology

4.1 Proposed Approach Architecture

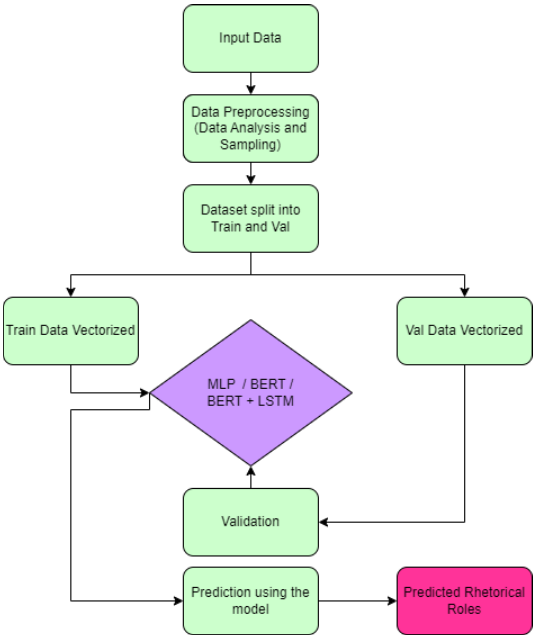

Input Data
Data Preprocessing (Data Analysis and Sampling)
Dataset split into Train and Val
Train Data Vectorized
Val Data Vectorized
MLP / BERT / BERT + LSTM
Validation
Prediction using the model
Predicted Rhetorical Roles

Figure 4: Proposed Model Approach

We propose a model architecture which includes data preprocessing where we will remove unwanted punctuation's, numbers, symbols, normalize the data, etc. Next we will split the data into train and validation for purpose of training and validation. After this we choose models such as Multi-layer Perceptron, BERT and BERT+LSTM to train the model on the data and evaluate the performance of the model.Then we validate the model based on test data and choose a best model. After each model we predict the roles and form a report with confusion matrix.

4.2 Data Preprocessing

Data processing is done on the dataset before proceeding further to the model development because to check and convert the dataset format into the required format for training. Have done iteration over the whole dataset and converted the dataset into dataframe for further processing. Some of the data preprocessing techniques used are as follows:

- Conversion of all the test into lower case
- Removal of unwanted punctuation as they won't provide any value to data and they introduce noise in the dataset.
- Normalizing the some punctuation's which might have redundant punctuation's after words in the text.
- Removal of extra spaces and normalizing it in the dataset by using regular expressions.
- Normalizing certain words when two words are seperated with a punctuation as '-' so that we get two words by separating.
- Tokenization of words in each sentences.
- Removal of stop words in each sentence as these words are common and add no much value to training in this dataset.

4.3 Data Analysis

The given dataset consists of 28986 segmented sentences in total but the distribution of sentences are more towards three to four roles. So this is one of the main concern as the dataset is imbalanced and has more weightage towards three classes which will make the model to learn towards those labels. To over come this we have under sampled the dataset on the top three most labels to 1600 samples so it balances the dataset. The below figure depicts the frequency distribution of sentences after sampling.

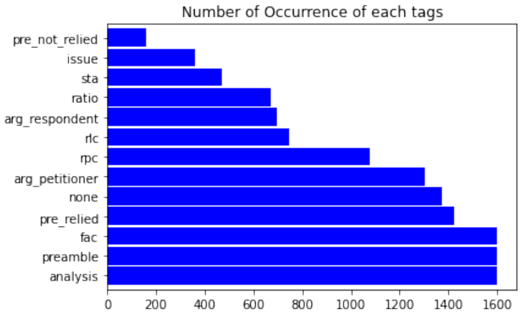


Figure 5: Frequency distribution of data after sampling

4.4 Feature Extraction

As we know all the models in machine learning needs the data in numerical format to do the processing we can't provide the direct text form from the dataset for training the model. In order to see that the input for the machine learning model is in numeric form or vector representation we use feature extractions methods such as one hot encoding, bag of words,etc. The main aim of this is dimensionality reduction where the reduction of random features in number is considered so to obtain principal features. The fewer features will contain the same information as the non reduced features which are in form of text.

- **Bag of Words:** In this technique we first do the data preprocessing and then we make a list of unique words in the corpus of text which is known as vocabulary. And this vocab is used to represent the text for input into vector form represented using text to index format or one hot encoding.

- **One Hot Encoding:** The one hot encoding is the technique of converting categorical data referring to label values which we have in our dataset to a list of binary representation. And this method is used to convert all the data required into vectors for the input to the machine learning model.

4.5 Models

The models we are using here are neural networks and pre trianed BERT models, mainly used for classification tasks, unsupervised learning, etc. These neural network models mainly work by making the prediction by combining the weights with features and applying a activation function at the output layer. The three models we are using here are as follows:

1. Multi Layer Perceptron (MLP) Model
2. BERT Model
3. BERT + LSTM Model

1. Multi Layer Perceptron Model : Multi Layer Perceptron is the one which is fully connected multi layer neural network. It is similar to the single layer perceptron but has more number of hidden layers with different weights. As the single layer perceptron has only feed-forward network, the multi layer perceptron has one more stage a back propagation stage which is based on the error, it back propagates the error and finds the derivative with respect to each weight and update the model so to minimize the error.

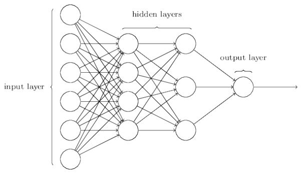


Figure 6: Multi Layer Perceptron

2. BERT Model : BERT Transformer (4) is an attention mechanism that learns the contextual relationships between words (or sub words) in a text. Transformer's basic design consists of two independent mechanisms: an encoder that reads the text input and a decoder that generates prediction. The Transformer encoder reads the entire sequence of words at once, in contrast to directional models, which read the text input sequentially (from right to left or left to right).Word sequences are changed with a [MASK] token for 15 percent of the words in each sequence before being fed into the BERT. Based on the context offered by the other, non-masked words in the sequence, the model then makes an attempt to forecast the original value of the masked words.

3. BERT + LSTM Model Long Short Term Memory Networks, most commonly referred to as "LSTMs," are a unique class of RNN that can recognize long-term dependencies. The capability of LSTM networks to combat the RNN's vanishing gradients or long-term dependence made us choose LSTM over RNN. Additionally, LSTM gives us the opportunity to make the layer bidirectional (6).

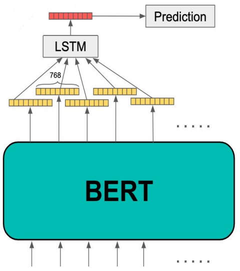


Figure 7: Multi Layer Perceptron (7)

5 Experiments

This section depicts the actual experiments carried out on the given dataset. Here we have used totally three models such as MLP, BERT and BERT+LSTM on different hyper-parameters. The models are mainly evaluated based on predictions on which we calculate precision, recall and F1-score. The formulas for calculation of precision, recall and F1-scores are as follows:

$$Precision = \frac{TP}{TP+FP}$$

$$Recall = \frac{TP}{TP+FN}$$

$$F1 = \frac{2*Precision*Recall}{Precision+Recall}$$

The first experiment we did was using the Multi-Layer Perceptron model where we have used the hyper-parameters as depicted in the following table. The loss function used is Binary Cross Entropy. The optimizer used is Adam. We took the sampled

Hyperparameter	Value
`Epoch`	5
`Number of Hidden Layers`	2
`Batch Size`	64
`Dropout`	0.2 & 0.3
`Activation function`	Relu

Table 1: Multi layer perceptron model summary

data to perform the training and achieved the following scores on the MLP model.

Epochs	Train Accuracy	Validation Accuracy	F1-score
5	0.833	0.341	0.31

Table 2: Table of Accuracy for MLP

The confusion matrix for the MLP model is as follows:

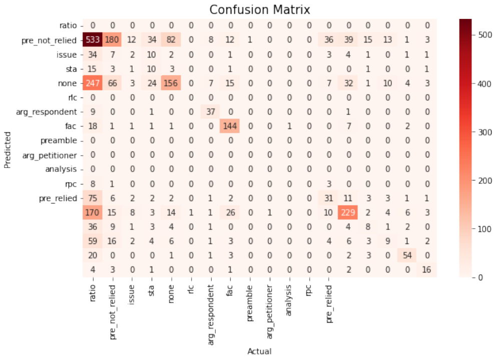


Figure 8: Multi Layer Perceptron confusion matrix

The next experiment we did was on BERT pretrained model and the parameters used for this model are show in the following table. And we performed this experiment on both not sampled and sampled data as there was imbalance in the given dataset.

Hyperparameter	Value
`Epoch`	7
`Embedding Size`	300
`Batch Size`	32 & 16
`Dropout`	0.3
`Learning Rate`	2e-5
`Loss function`	Cross Entropy

Table 3: BERT model summary

After carrying out the experiment on the above parameters we achieved an accuracy scores as follows for both not sampled and sampled data:

Data	Train Accuracy	Test Accuracy	F1-score
`Not Sampled`	0.850	0.608	0.61
`Sampled`	0.887	0.553	0.55

Table 4: Table of Accuracy for BERT

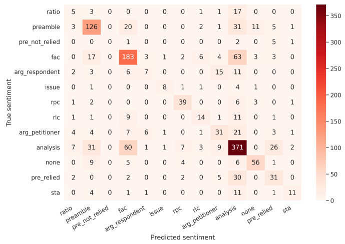


Figure 9: BERT with not sampled data confusion matrix

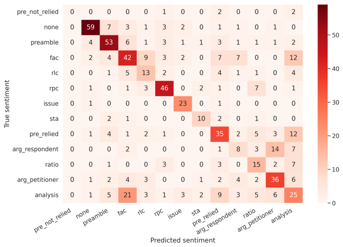


Figure 10: BERT with sampled data confusion matrix

The next experiment we did was adding a LSTM layer over a BERT pre trained model and the parameters used for this model are show in the following table. And we performed this experiment on both not sampled and sampled data as there was imbalance in the given dataset.

Hyperparameter	Value
`Epoch`	10 & 15
`Embedding Size`	768
`Batch Size`	32 & 16
`Dropout`	0.3
`Learning Rate`	2e-5
`Loss function`	Cross Entropy

Table 5: BERT+LSTM model summary

After carrying out the experiment on the above parameters we achieved an accuracy scores as follows for both not sampled and sampled data:

Data	Train Accuracy	Test Accuracy	F1-score
`Not Sampled`	0.852	0.602	0.60
`Sampled`	0.916	0.562	0.56

Table 6: Table of Accuracy for BERT+LSTM

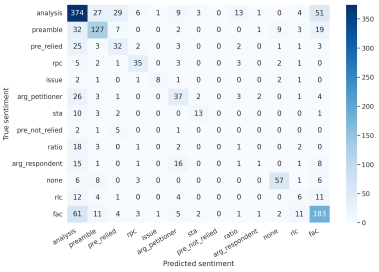


Figure 11: BERT+LSTM not sampled data confusion matrix

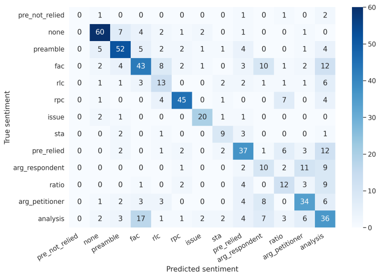


Figure 12: BERT+LSTM sampled data confusion matrix

6 Results

The main aim of this work is to build a model that predicts the rhetorical roles of segmented sentences from a legal document. So for this we had used the dataset which consisted of totally 28986 sentences. But the data was highly imbalanced and we used under sampling method to balance the dataset. And then performed experiment on three models such as MLP, BERT and BERT+LSTM where in we achieved an F1 scores for each model as follows:

Model	**F1-Score**
`MLP(Sampled Data)`	0.31
`BERT (Not Sampled)`	0.61
`BERT (Sampled)`	0.55
`BERT+LSTM (Not Sampled)`	0.60
`BERT+LSTM (Sampled)`	0.56

Table 7: Table of Accuracy for BERT+LSTM

We can see that there was very little improvement of 1-2% when an LSTM was added over the BERT, but LSTM actually helps in vanishing gradients issue and help in learning with context of next and previous words. From the above table we can conclude that the BERT+LSTM model over the sampled data performed the best over all the other models.

7 Error Analysis

In this part we can consider the confusion matrices that are depicted in the above model experiments. We can consider the confusion matrices of both BERT and BERT+LSTM models. And the confusion matrix is where each cell indicates how often a label on x-axis was classified as same label on y-axis. The diagonal cells corresponds to where the label was classified correctly hence the diagonal is the one where we can check for good results. If we consider the figures 9 and 11 which was actually on not sampled data we can see that mainly 3 classes are being predicted the most, which implies that they are predicted no matter which class a sample belong to. But when you consider the figures 10 and 12 we can see that the diagonal is distributed so here it predicts almost all the class labels based on the sample. Even though the accuracy was more on not sampled data the predictions were mainly towards couple of classes but on sampled data the confusion matrix depicts that it performed well and there are less outliers.

8 Team Work

The whole project was done as a team and everyone was part of every task but as per task division, the tasks done by each team member is as follows: The Data Preprocessing, data analysis & sampling, report and BERT model was handled by Parikshith Honnegowda. The Literature study and MLP model was handled by Xin Zhang. The BERT+LSTM, error analysis, report was handled by Vijay Chlaka.

References

[1] Matthias Damaschk, Tillmann Dönicke, and Florian Lux. 2019. Multiclass Text Classification on Unbalanced, Sparse and Noisy Data. In Proceedings of the First NLPL Workshop on Deep Learning for Natural Language Processing, pages 58–65, Turku, Finland. Linköping University Electronic Press..

[2] Prathamesh Kalamkar, Aman Tiwari, Astha Agarwal, Saurabh Karn, Smita Gupta, Vivek Raghavan, and Ashutosh Modi. 2022. Corpus for automatic structuring of legal documents. pages 4420–4429

[3] Vijit Malik, Rishabh Sanjay, Shouvik Kumar Guha, Shubham Kumar Niga, Angshuman Hazarika, Arnab Bhattacharya, and Ashutosh Modi. 2021. Semantic segmentation of legal documents via rhetorical roles.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. https://doi.org/10.48550/arXiv.1810.04805

[5] Paheli Bhattacharya, Shounak Paul, Kripabandhu Ghosh, Saptarshi Ghosh, and Adam Wyner. 2019. Identification of rhetorical roles of sentences in indian legal judgments.

[6] https://mccormickml.com/2019/07/22/BERT-fine-tuning/

[7] https://twitter.com/KirkDBorne/status/1154018344128995329/photo/1

[8] https://github.com/Legal-NLP-EkStep/rhetorical-role-baseline