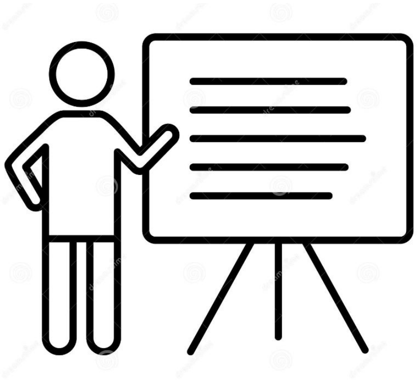
BTP - B23RS01

Legal Statute Identification

Under The Guidance of **Dr R. Shathanaa**

OUTLINE



- Introduction
- Motivation
- Problem statement
- Literature review
- Objective
- Proposed Approach
- Available Datasets
- Results
- RoadMap (Plan for two semester)
- References

INTRODUCTION

- Legal Judgment Prediction (LJP) examines the given input case document and recommends the judgment components such as applicable law sections, prison term, and penalty as delivered by the judge in the court.
- Law section is a key component and statute prediction plays a vital role in the legal decision in determining the remaining components.
- The legal process can be time-consuming and costly, and there is a growing need for ways to improve its efficiency and effectiveness.
- Section prediction through Deep learning algorithms can expedite decision-making, reduce the workload of legal professionals, and improve the efficiency and accessibility of the legal system for all.

MOTIVATION

- Pending cases are a major issue in the Indian legal system, with millions of cases backlogged in courts across the country.
- Factors such as a shortage of judges, increasing awareness of the rights of the common man and too much litigation from the government side contribute to the high number of pending cases.
- Section prediction plays a significant role in judgement prediction by correctly interpreting the applicable statutes based on the given fact description.
- It assists in the precise analysis of the case document and expedite the decision thereby reducing the count of pending cases in the Indian legal system and provide valuable insights to improve the system as a whole.

PROBLEM STATEMENT

- Predicting the applicable IPC sections based on the precedence by correctly interpreting the given fact description.
- Choosing appropriate Deep Learning Model that gives the best performance.

LITERATURE REVIEW

AUTHOR AND YEAR	TITLE	METHODOLOGY	EVALUATION PARAMETERS	CONS
Yao, Fanglong, Xian Sun, Hongfeng Yu, Yang Yang, Wenkai Zhang, and Kun Fu [2020]	Gated hierarchical multi-task learning network for judicial decision prediction [1]	Model Architecture: 1) GHE- uses sentence encoder to draw in-depth semantics 2) DAP- multi-task dependency learning module Predicts judgment by capturing the semantic information of the sentences and learning dependencies between the various judgment components.	• Accuracy • Precision • Recall • F1-score for the information extraction task.	Do not consider the commonalities and specificities between the tasks.

LITERATURE REVIEW

AUTHOR AND YEAR	TITLE	METHODOLOGY	EVALUATION PARAMETERS	PROS
Yao, Fanglong , Xian Sun, Hongfeng Yu, Wenkai Zhang, and Kun Fu [2021]	Commonalities -, specificities-, and dependencies-enhanced multi-task learning network for judicial decision prediction [2]	Model Architecture: 1) commonalities-enhanced feature extractor. 2) specificities-enhanced task network 3) dependencies-enhanced joint learning network	• Accuracy • Precision • Recall • F1-score	Addressed the commonalities and specificities between the tasks.

LITERATURE REVIEW

AUTHOR AND YEAR	TITLE	METHODOLOGY	EVALUATION PARAMETERS	CONS
Vijit Malik, Rishabh Sanjay, Shubham Kumar Nigam [2020]	ILDC for CJPE: Indian Legal Documents Corpus for Court Judgment Prediction and Explanation [3]	1) Developed a binary classification dataset with Indian supreme court case proceedings. 2) A combination of web scraping and manual annotation used to gather and annotate the court judgments 3) Use of DL techniques to predict the outcome of a case based on its facts and circumstances	• Accuracy • Precision • Recall • F1 - Score	Some documents does not contain decision in last few sentences or tokens.

LITERATURE REVIEW

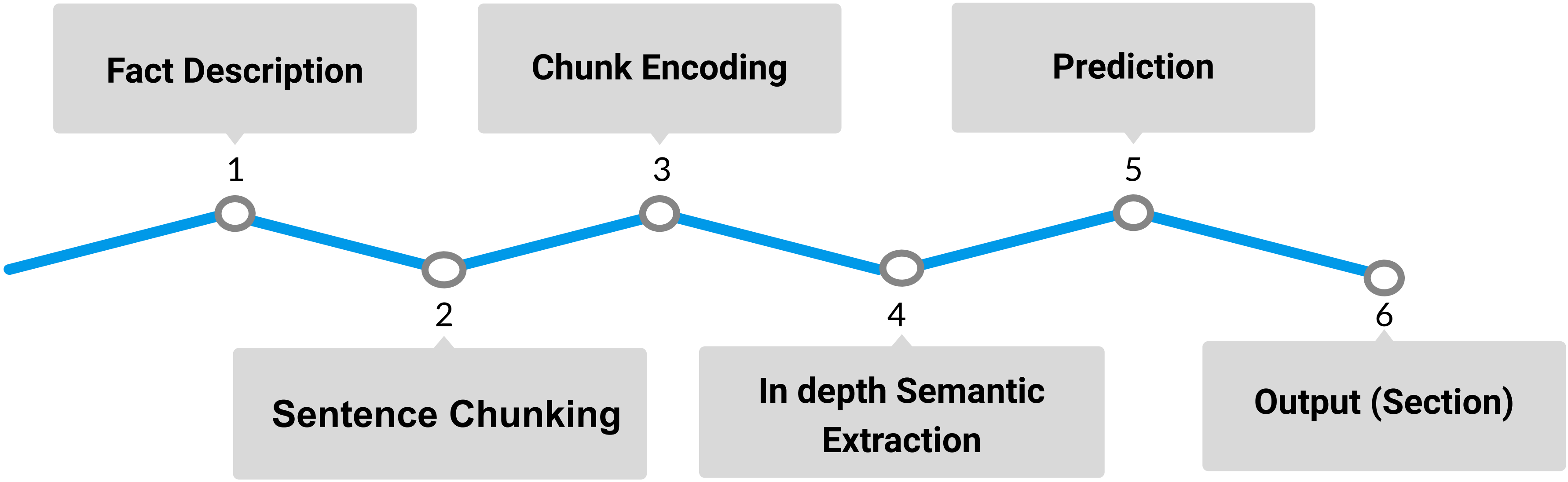
AUTHOR AND YEAR	TITLE	METHODOLOGY	EVALUATION PARAMETERS	DRAWBACKS
Shounak Paul, Pawan Goyal, Saptarshi Ghosh [2020]	A Heterogeneous Graph-based Approach for Automatic Legal Statute Identification from Indian Legal Documents [4]	1) Pre-processing 2) Graph creation (nodes) 3) Statute identification (Random walk) 4) Evaluation	• Accuracy • Precision • Recall • F1-score • Higher of all parameters higher the identification of statues.	Dataset contains only statutes with out prison term and penalty components.

OBJECTIVE

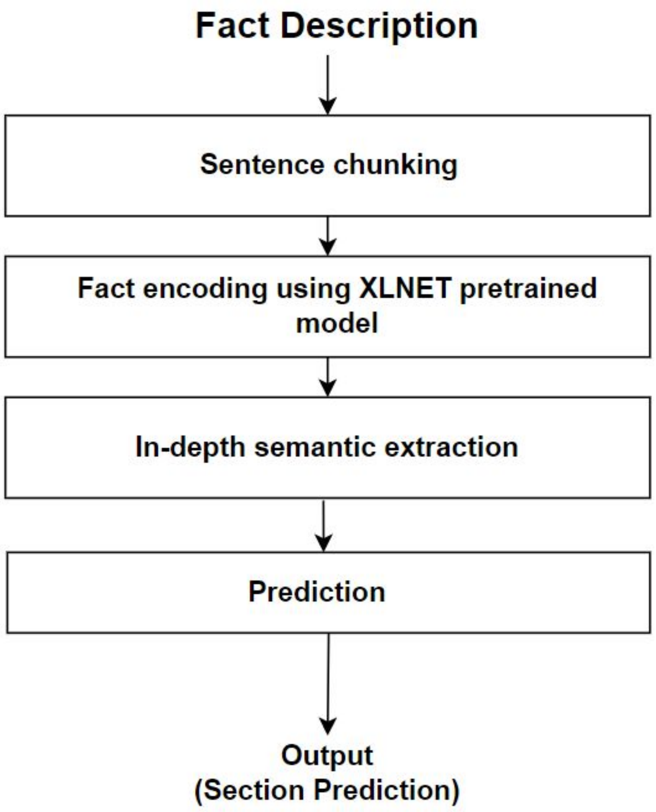
- Our main objective is to analyze a given set of legal facts and predict the applicable legal sections that applies to the case.

- The ultimate goal is to improve the efficiency and accuracy of legal judgement prediction by automating the process of section prediction using advanced NLP and DL techniques.

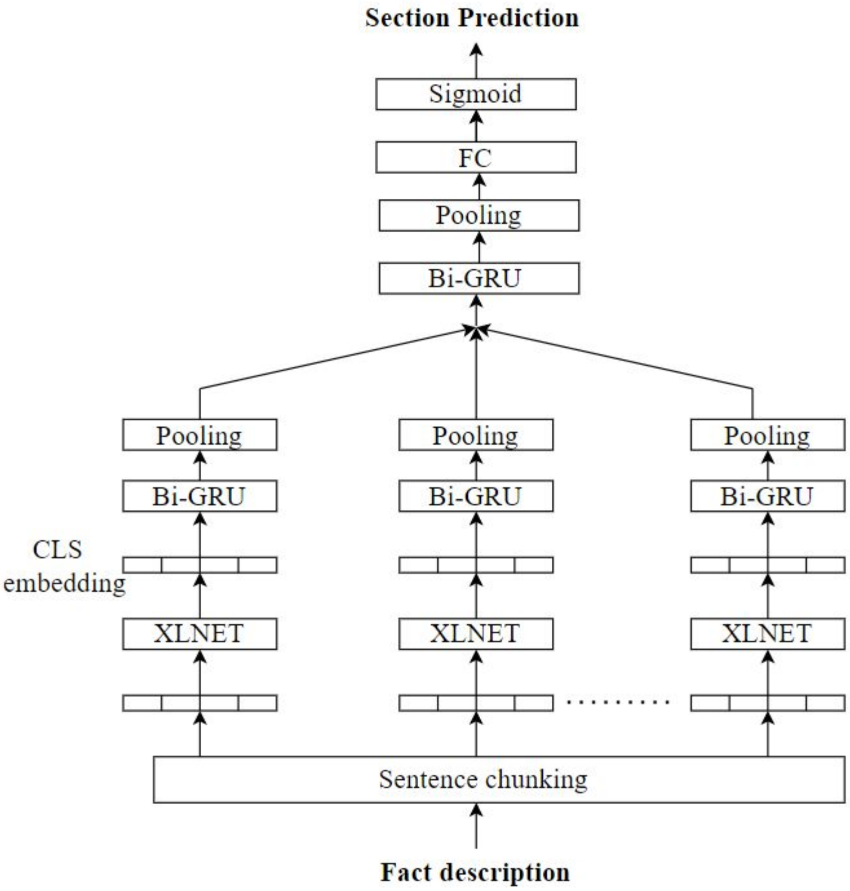
PROCEDURE


Fact Description
1
Sentence Chunking
2
Chunk Encoding
3
In depth Semantic Extraction
4
Prediction
5
Output (Section)
6

Overview Of Architecture



Detailed Architecture



Why XLNET and Bi-GRU[3]

Performance of BERT and XLNET

Model	Macro-Precision	Macro-Recall	Macro-F1	Accuracy
BERT + BiGRU + att.	71.31	70.98	71.14	71.26
XLNet + BiGRU + att.	**77.32**	**76.82**	**77.07**	**77.01**

Why XLNET and Bi-GRU[3]

Performance of XLNET with various gate mechanisms

Model	Macro-Precision	Macro-Recall	Macro-F1
XLNet+BiGRU+att.	**77.32**	**76.82**	**77.07**
XLNet+GRU+att.	74.55	74.32	74.43
XLNet+BiLSTM+att.	75.98	75.91	75.94
XLNet+LSTM+att.	74.95	74.37	74.66

DataSet Used for the ANALYSIS[3]

ILDC DataSet:

Corpus (Avg. Tokens)	Number of Documents		
	Train	Validation	Test
ILDC	32305	994	1517

ANALYSIS

Model	**Precision (3e)**	**Recall (3e)**	**F1 (3e)**	**Accuracy (3e)**
XLNet+BIGRU+pooling	0.66314	0.62175	0.64175	0.68778
XLNet+BiGRU+pooling+ softmax at dense layer	NA	0.5	NA	0.50197
XLNet+BiGRU(100)+pooling+BiGRU(100) + pooling	0.75479	0.73092	0.74266	0.73342

ANALYSIS

Model	Precision (3e)	Recall (3e)	F1 (3e)	Accuracy (3e)
XLNet+BiGRU(100)+pooling+BiGRU (50) + pooling	0.75202	0.72962	0.75065	0.73511
XLNet+BiGRU(100)+pooling+BiGRU (150) + pooling	0.75553	0.73827	0.74680	0.74133
XLNet+BiGRU(100)+pooling+BiGRU (100) + pooling with softmax in dense layer	NA	0.5	NA	0.50197

ANALYSIS

Model	Precision (3e)	Recall (3e)	F1 (3e)	Accuracy (3e)	Precision (20e)	Recall (20e)	F1 (20e)	Accuracy (20e)
XLNet+BiGRU	0.75075	0.73902	0.74484	0.73944	0.68800	0.68800	0.688001	0.68799
XLNet+BiGRU(100)+pooling+BiGRU(100) + pooling	0.75075	0.73902	0.74484	0.73944	0.71581	0.71491	0.71536	0.71627
XLNet+BiGRU(100)+pooling+2 BiGRU(100) + pooling	0.75023	0.73299	0.74151	0.7415	0.69403	0.68769	0.69084	0.68502

ANALYSIS

Model	Precision (3e)	Recall (3e)	F1 (3e)	Accuracy (3e)	Precision (20e)	Recall (20e)	F1 (20e)	Accuracy (20e)
XLNet+2BiGRU(100)+pooling+BiGRU(100) + pooling	0.75370	0.71759	0.73520	0.72293	0.65626	0.63658	0.64627	0.63511
XLNet+BiGRU(100)+dropout+pooling+BiGRU(100) + pooling	0.75415	0.73294	0.74339	0.73674	0.69249	0.68701	0.68974	0.68773
XLNet+BiGRU(100)+pooling+BiGRU(100) + pooling+dropout	0.74648	0.73305	0.73970	0.74089	0.72589	0.72551	0.72570	0.73164

RESULT

Model	Precision (3e)	Recall (3e)	F1 (3e)	Accuracy (3e)	Precision (20e)	Recall (20e)	F1 (20e)	Accuracy (20e)
XLNet+BiGRU(100)+pooling+BiGRU(100)+pooling+dropout+shuffle	0.76311	0.76262	0.76287	0.76440	0.72519	0.72001	0.72260	0.72288
XLNet+BiGRU(100)+dropout(0.2)+pooling+BiGRU(100)+pooling+dropout+shuffle	**0.77161**	**0.77057**	**0.77109**	**0.77055**	**0.750507**	**0.74757**	**0.74904**	**0.74379**

Final Model

(3e) : 3 Epochs Result
(20e): 20 Epochs Result

Final Model

Layer (type)	Output Shape	Param #
text (InputLayer)	[(None, None, 768)]	0
masking (Masking)	(None, None, 768)	0
bidirectional (Bidirectiona l)	(None, None, 200)	522000
max_pooling1d (MaxPooling1D)	(None, None, 200)	0
bidirectional_1 (Bidirectio nal)	(None, None, 200)	181200
max_pooling1d_1 (MaxPooling 1D)	(None, None, 200)	0
dropout (Dropout)	(None, None, 200)	0
dense (Dense)	(None, None, 30)	6030
dense_1 (Dense)	(None, None, 1)	31

Total params: 709,261
Trainable params: 709,261
Non-trainable params: 0

Sample Output

```
print(x_test0[9])
print(x_test0[9].shape)
```

```
[[ 0.10606287 -0.27953875 -0.09833915 ...  0.6276578   0.21352552
  -0.06789596]
 [ 0.1081848   0.00159596  0.11206068 ...  0.22839008  0.22549874
  -0.16895068]
 [ 0.2503333  -0.47446337 -0.23584625 ...  0.02303861  0.08077054
  -0.2940132 ]
 ...
 [ 0.3588356   0.38372254 -0.2853119  ... -0.603173    0.00332183
   0.45234802]
 [-0.2866297   0.7653041   0.13413383 ... -0.9453125  -0.2061241
   0.59378815]
 [ 0.02419375  0.19298854  0.13397974 ... -0.7381441  -0.292474
  -0.08548267]]
(20, 768)
```

[13]
```
x_test = np.expand_dims(x_test0[9], axis=0) # add a new axis for the timestep
y_pred = model.predict(x_test)
print(y_pred)
```

```
1/1 [==============================] - 5s 5s/step
[[[0.06047338]
  [0.06048959]
  [0.05860003]
  [0.06193072]
  [0.06120095]]]
```

[14]
```
count_zeros = np.sum(y_pred == 0)
count_ones = np.sum(y_pred == 1)
if count_zeros > count_ones:
    y_pred = 0
else:
    y_pred = 1

print(y_pred)
```

```
1
```

AVAILABLE DATASET[4]

DataSet Name	Indian Legal Statute Identification (ILSI)
Language	English
No. of documents / Facts	66, 090
No. of labels / statutes	Top 100 Frequent sections
Avg. no. of words per doc	1232
Avg. no. of labels per doc	3.78

ROADMAP

Phase 1 — Learning and Data Gathering (4 weeks)

Phase 2 — Building our Model (6 weeks)

Phase 3 — Testing (5 weeks)

Phase 4 — Optimising (3 weeks)

REFERENCES

1. Yao, Fanglong, Xian Sun, Hongfeng Yu, Yang Yang, Wenkai Zhang, and Kun Fu. "Gated hierarchical multi-task learning network for judicial decision prediction." *Neurocomputing* vol. 411 (2020): pp. 313-326.

2. Yao, Fanglong, Xian Sun, Hongfeng Yu, Wenkai Zhang, and Kun Fu. "Commonalities-, specificities-, and dependencies-enhanced multi-task learning network for judicial decision prediction." *Neurocomputing,* vol. 433 (2021): pp. 169-180.

REFERENCES

3. Malik, Vijit, Rishabh Sanjay, Shubham Kumar Nigam, Kripa Ghosh, Shouvik Kumar Guha, Arnab Bhattacharya, and Ashutosh Modi. "ILDC for CJPE: Indian legal documents corpus for court judgment prediction and explanation." *arXiv preprint arXiv:2105.13562* (2021).

4. Paul, Shounak, Pawan Goyal, and Saptarshi Ghosh. "LeSICiN: A heterogeneous graph-based approach for automatic legal statute identification from Indian legal documents." In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 10, pp. 11139-11146. 2022.

GROUP MEMBERS

Rahul Varma (S20200010212)

Kamal Sai (S20200010233)

Krishna Vamshi (S20200010138)