ResearchTeam_HCN at SemEval-2023 Task 6: A knowledge enhanced transformers based legal NLP system

Dhanachandra Ningthoujam[†], **Pinal Patel**[*],
Rajkamal Kareddula[*], **Ramanand Vangipuram**[†]
[*]RAAPID.ai
[†]HEALTHCARENLP SOFTECH LLP
[†]{dhanachandra.n,ramanandh.v}@healthcarenlp.com
[*]{pinal,rajkamal}@raapid.ai

Abstract

This paper presents our work on LegalEval (understanding legal text), one of the tasks in SemEval-2023. It comprises of three sub-tasks namely Rhetorical Roles (RR), Legal Named Entity Recognition (L-NER), and Court Judgement Prediction with Explanation (CJPE). We developed different deep-learning models for each sub-tasks. For RR, we developed a multi-task learning model with contextual sequential sentence classification as the main task and non-contextual single sentence prediction as the secondary task. Our model achieved an F1-score of 76.50% on the unseen test set, and we attained the 14th position on the leaderboard. For the L-NER problem, we have designed a hybrid model, consisting of a multi-stage knowledge transfer learning framework and a rule-based system. This model achieved an F1-score of 91.20% on the blind test set and attained the top position on the final leaderboard. Finally, for the CJPE task, we used a hierarchical approach and could get around 66.67% F1-score on judgment prediction and 45.83% F1-score on the explainability of the CJPE task, and we attained 8th position on the leaderboard for this sub-task.

1 Introduction

The court pending cases are rapidly raising in populous countries like India. Towards this challenge, systems that can automatically process, understand and organize legal documents need to be designed. Another major challenge of processing legal documents is that they are unstructured, very lengthy, contain a lot of legal jargon, etc. Thus, the existing NLP systems can not be directly applied. The SemEval-2023 (Modi et al., 2023) provides 3 different problems on the Indian legal documents namely: 1. Rhetorical Roles (RR) (Kalamkar et al., 2022b) (Malik et al., 2021a): Segmenting a legal document into semantically similar sentences. The RR task can be defined as the classification of sentences in a legal document into pre-defined classes. 2. Legal Named Entity Recognition (L-NER) (Kalamkar et al., 2022a): Legal entity extraction from the legal document. The task is to find out the legal entities like Court names, Petitioner names, Respondents, Judge names, and case numbers mentioned in a given legal document. 3. Court Judgement Prediction with Explanation (CJPE)(Malik et al., 2021b): Court judgment output prediction with explanations. Given a legal document, the task is to find out the final outcome of the case i.e whether the appeal raised by the petitioner is accepted or not. In case of acceptance, generate the explanation of the case. The datasets for all the above sub-tasks are available in the English language. In this work, we present our proposed solutions for the above three problems. We proposed a multi-task learning neural network model to solve the sub-task RR. Our model consists of two classification layers. The first classifier is the sequential sentences classification which is the primary task. Next, the second classifier is the single sentence classification which acts as the secondary task. For the legal NER task, we proposed a hybrid model consisting of a multi-stage sequence-to-sequence training framework based on the pre-trained large legal language model and a rule-based system to improve the performance of the neural network model. Finally, for the CJPE sub-task, we proposed two separate models for judgment prediction and explanation generation. The work of Lee et al. 2020 has demonstrated that fine-tuning or pre-training of language models on domain-specific data significantly improves the model performance on the downstream task. They have validated the same by pre-training the BERT model (Devlin et al., 2018) (BioBERT) on Bio-Medical text data which significantly outperformed the BERT model which was pre-trained on general text data. We also developed our own LLM based on RoBERTa (Liu et al., 2019). All of our proposed models are based on our legal RoRERTa

Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023), pages 1245–1253

model. For CJPE sub-task 1, we used a hierarchical approach to train the model. In the first stage, we used our Legal RoBERTa to fine-tune the model on the dataset and this fine-tuned model embedding is used to train a BiGRU model for the final predictions.

In recent years, there has been a lot of research and development work on legal NLP like Chinese Legal NER and Relationship extraction (Chen et al., 2020), Legal Judgment Prediction for Chinese (Zhao et al., 2022), Russian Court Decisions Data Analysis Using Distributed Computing and Machine Learning to Improve Lawmaking and Law Enforcement (Metsker et al., 2019). However, the application of NLP and ML solutions in the Indian legal system is low. The legal documents are very different from the general text, they are unstructured, very long, and contain a large amount of jargon. The existing NLP systems can not be directly applied for solving the above-mentioned problems. Our proposed solutions leverage a pre-trained large language model trained on Indian legal documents. Our proposed models achieve an F1-score of 76.50%, 91.20%, and 66.67% on the hidden test set of RR, L-NER, and CJPE respectively. Our main contributions are summarized below:

1. Development of a large language model for Indian legal judgment text based on the RoBERTa pre-training strategy.

2. Proposed a deep learning-based multi-task learning model with attention pooling mechanism for RR task.

3. Introduce a multi-stage learning neural network model to solve the legal NER problem.

4. Our legal NER model performs exceptionally well and occupies the first rank in the SemEval-23 legal NER task with an F1-score of 91.20% in the hidden test dataset.

5. Proposed a hierarchical model with domain-specific language model for CJPE classification task and achieved around 66.71% F1-score.

2 Background

The LegalEval task of SemEVal-23 consists of three sub-tasks.

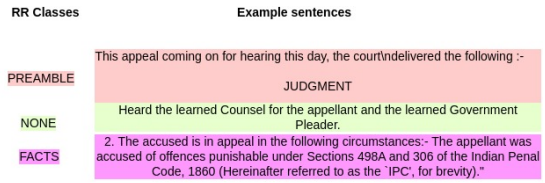

Figure 1: Example of sentences with RR labels

2.1 Rhetorical Roles (RR):

The task is to classify the sentences in a legal judgment into semantically similar classes. The SemEval-23 RR task's dataset consists of 13 class labels namely Preamble (PREAMBLE), Facts (FAC), Ruling by Lower Court (RLC), Issues (ISSUE), Argument by Petitioner (ARG_PETITIONER), Argument by Respondent (ARG_RESPONDENT), Analysis (ANALYSIS), Statute (STA), Precedent Relied (PRE_RELIED), Precedent Not Relied (PRE_NOT_RELIED), Ratio of the decision (RATIO), Ruling by Present Court (RPC) and None (NONE). The detailed definitions of each class and the dataset preparation strategies are provided in Kalamkar et al. 2022b. We also propose a baseline system using the deep learning architecture proposed in Brack et al. 2021. Malik et al. 2021a proposed a multi-task learning (MLT) framework with rhetorical role classification as the primary task and label shift prediction as the auxiliary task. The label shift prediction is a problem that the model has to predict whether the current sentence S_t has the same label as the previous sentence S_{i-1} in a legal document. Another work from Ghosh and Wyner 2019 proposed a BiLSTM-CRF model with the sent2vec model as a feature extraction layer. An example sample of the RR dataset is provided in Figure 1

2.2 Legal NER:

The Legal NER is the task of classifying all the tokens present in a sentence in a legal judgment document. The legal entities covered in SemEval-23 include 14 entity types, COURT, PETITIONER, RESPONDENT, JUDGE, LAWER, DATE, ORG, GPE, STATUTE, PROVISION, PRECEDENT, CASE_NUMBER, WITNESS, and OTHER_PERSON. Detailed definitions of each class and dataset preparation methods are provided in Kalamkar et al. 2022a. For this task, we proposed a hybrid model consisting of a multi-stage knowledge transfer framework and a rule-based system. Some of the previous work done on simi-

lar problem statements are discussed below: The work of Leitner et al. 2019 proposed an approach for Named Entity Recognition (NER) in the legal domain of the German language. The dataset consists of 67,000 sentences and 54,000 annotated entities approximately. They proposed a state-of-the-art model comprising of Conditional Random Fields (CRFs) and Bidirectional Long-Short Term Memory Networks (BiLSTMs). In another work, Trias et al. 2021 proposed an ensemble language model using a transformer neural network architecture combined with a finite state machine to extract names from the English language legal text. In the work of Au et al. 2022, the authors describe a publicly available legal NER dataset called ENER, based on legal company filings available from the US Securities and Exchange Commission's EDGAR dataset. They demonstrated that training a number of different NER algorithms on the general English CoNLL-2003 corpus but testing on their test collection confirmed significant degradation in accuracy, as measured by the F1-score. Chen et al. 2020 proposed a legal triplet extraction system for drug-related criminal judgment documents. The system extracts the entities and the semantic relationships jointly and benefits from the proposed legal lexicon feature and multi-task learning framework.

2.3 CJPE:

Given a legal judgment document, the task involves automatically predicting the case's outcome (binary: accepted or denied) and providing an explanation for the prediction. The explanations are in the form of relevant sentences in the document that contribute to the decision. For the classification task, the problem is a binary classification of predicting whether the outcome is accepted or denied. There are two types of datasets provided for this task single and multi. A single dataset is a dataset where each document contains only information or text related to one case. A multi-dataset is where one sample in the dataset may contain text from multiple cases. Here are a few references to the previous works done in this domain:

In the recent work of Strickson and De La Iglesia 2020, the authors have presented a prediction system that can be used at the judgment drafting stage, and the identification of the most important words and phrases within a judgment, based on which they automatically try to predict the outcome of a court case given only the case document. They have accomplished this by creating a labeled dataset of UK court judgments and the subsequent application of machine learning models. In another work, Kowsrihawat et al. 2018 proposed a prediction model for criminal cases from the Thai Supreme Court using End-to-End Deep Learning Neural Networks. Their model imitates a process of legal interpretation, whereby recurrent neural networks read the fact from an input case and compare them against relevant legal provisions with the attention mechanism. In the recent work of Alghazzawi et al. 2022, the authors proposed a pipeline where they initially worked on prioritizing and choosing features that scored the highest in the provided legal data set, only the most pertinent features were picked. After that, the LSTM+CNN model was utilized to forecast lawsuit verdicts. Another work in Xu et al. 2020, proposed a multi-task legal judgment prediction model which combines a sub-task of the seriousness of charges. By introducing this sub-task, their model was able to capture the attention weights of different terms of penalty corresponding to charges and give more attention to the correct terms of penalty in the fact descriptions.

3 System overview

In the recent years, because of the emergence of pre-trained large language models (LLM), the performance of the NLP models has drastically improved. Though there are many pre-train LLMs for general English and other domains, there are very few LLMs for Indian legal text (Paul et al., 2022). Towards this, we also developed our own Indian legal RoBERTa language model. The mask language modeling pre-training objective was adopted for pre-training the model. Our LLM is pre-train on around 1.1M documents which we scrapped from the Indian Kanoon website [1]. While collecting these documents, we considered the Indian court cases from the years 1960 to 2023. Also, we covered the cases of Major Indian courts like the Supreme Court, High Court, and district courts from different states and union territories of India. Moreover, we also covered 11 different types of cases of food adulteration, education, industrial disputes, tax, criminal, civil, automobile, land and property, industry and labor, constitutional and financial. The details of the actual dataset released

[1] https://indiankanoon.org/

by the LegaEval organizers are explained below in the data section. On top of the training datasets, we also generated a synthetic dataset to solve data scarcity as well as the class imbalance problem.

3.1 RR:

For this task, we proposed two baseline models and one main proposed model with an MLT framework. The first baseline model uses the sentence transformer model (Reimers and Gurevych, 2019) which encodes the sentences into fixed-dimension vector spaces. The SentenceTransformers model is initialized with the pre-trained model all-mpnet-base-v2. The pre-trained SentenceTransformers model weights are not allowed to update during the fine-tuning of the downstream task. Our proposed second baseline model is based on the transformers with attention pooling which was used in the work of Brack et al. 2021. In this approach, we have the option to choose any LLM, and the weights of the LLM are updated during the fine-tuning on the downstream task. The dataset for the RR task is highly imbalanced where some classes Precedent Not Relied constitute only 0.5%, Issues constitute only 1.26% whereas other classes like Analysis constitute 36.89%, Facts constitute 19.81%. To tackle this problem, our main proposed solutions use synthetic datasets for the low frequent classes. We used the back-translation technique to generate the new sentences for the low frequent classes to create a new dataset somewhat balanced. We translate a sentence in English in the RR dataset to Dutch, then translate it back to English from the Dutch sentence. The newly generated sentence is labeled the class as the original sentence. An example of back-translation is given below.

Source English sentence: We consider that these contentions are correct.

Translated Dutch sentence: Wij zijn van mening dat deze beweringen juist zijn.

Dutch to English Translation output: We believe that these claims are correct.

Thus, our final approach has the same model architecture as baseline 2 and is trained on the new dataset. Our multi-task learning framework consists of two classification heads one for the main multi-class sequence labeling task (RR) and the other as a single-sentence classification task. Each document D_i in the RR dataset having n number of sentences s_1, s_2, s_3 ... s_n. Then, each sentence s_i is passed through the transformer model to extract the

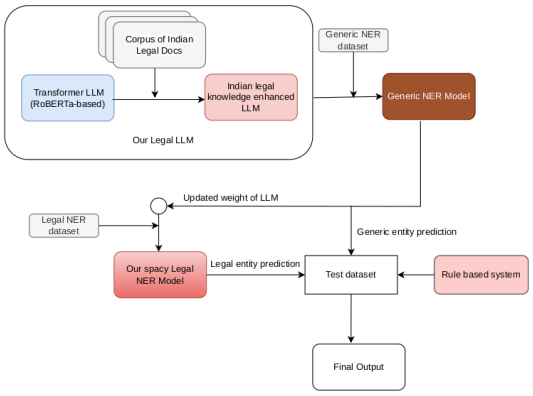

Figure 2: Our L-NER model architecture

contextual word vector $WV_i = LLM(Si)$. The SentenceTransformers (for baseline model 1) or Attention pooling mechanism (for baseline model 2 and our proposed model) extract sentence representation from the word vectors. These contextual word vectors are passed through a feedforward layer to predict the single sentence classification labels whereas, for the main sequential sentence classification layer, the concatenation of the contextual word vectors and single sentence classification output vectors are passed through a Bi-LSTM layer to extract the contextual sentence information and pass the output to the classification layer. Note: we don't use the CRF as a classification layer to reduce the computational complexity of training as well as to adapt RR problems into our proposed framework. During the training, we try to reduce the cross entropy losses from both the classifier. The RR baseline model 1, model 2, and our proposed model are represented as RR-SentenceTransformers, RR-ATTN-POOL, and RR-ATTN-POOL-SYNTH respectively in the following sections.

$$Total_{\text{loss}} = \omega * loss_{\alpha} + (1 - \omega) * loss_{\beta}$$

Where the $\text{loss}_{\text{alpha}}$ is the loss generated by the sequential sentence classifier, the $\text{loss}_{\text{beta}}$ is the loss generated by the single sentence classifier and the ω is a hyperparameter and the weight of the $\text{loss}_{\text{alpha}}$

3.2 L-NER:

We proposed a hybrid multi-stage knowledge transfer learning framework as shown in 2. The different components depicted in the above L-NER architecture are explained in the following subsections:

3.2.1 Legal LLM:

In order to capture the semantic information specific to legal documents, we developed an LLM

for Indian legal documents based on the RoBERTa (Liu et al., 2019) with an objective of masked language modeling. We utilized this LLM for training two different NER models:

3.2.2 Generic NER model:

To add a sense of generalizability, we have further fine-tuned it on the CoNLL-2003 (Sang and De Meulder, 2003) general data set which consists of four different entity types that are persons (PER), organizations (ORG), locations (LOC), and miscellaneous names (MISC). This generic NER model is also used in the post-processing step where we use some rules to correct the final predicted output from L-NER because as per our analysis, this generic NER model was able to detect some of the general entities that are missed by the final L-NER model. For Example, in the following sentence:

"CW3 Mr Vijay Mishra, Deputy Manager, HDFC Bank, Noida, UP has deposed that complainant had a current account with HDFC Bank in the year 2004"

The final L-NER model might miss out on entities like "Vijay Mishra" which should have been recognized as a "WITNESS" entity type. But in our observation, the generic NER model was doing good in detecting "Vijay Mishra" as a "PERSON" entity type. This behavior can be taken as an advantage in correcting the final L-NER predictions. Below we explain the relationship between the legal entity types and the entity types from the generic NER model.

If a new entity with the tag "PERSON" is detected by the generic NER model but was missed by the L-NER model, then based on context (or) keyword matching score we can assign one of "PETITIONER" (or) "RESPONDENT" (or) "LAWYER" (or) "JUDGE" (or) "WITNESS" (or) some "OTHER PERSON" legal entity type to it in the final output. For example, below are the keywords that we used to relate the detected "PERSON" entity type from the generic NER model to the "JUDGE" entity type.

"court", "judgment", "Justice", "Magistrate", "Honble", "Coram", "Shri", "Hon'ble"

Similarly, when a new "ORGANIZATION" entity type is detected by the generic NER model but was missed by the final L-NER model, we check if we have the keyword court in the detected entity text, if yes we assign the "COURT" legal entity type to it.

Lastly, when a new "LOCATION" entity type is detected by the generic NER model but was missed by the final L-NER model, we can assign it a "GPE" legal entity type as per our analysis.

3.2.3 Our Spacy Legal NER model:

After fine-tuning our legal RoBERTa-base model on the CoNLL-2003 dataset, in order to make the model aware of the semantic knowledge of the legal data provided in the SemEval-23, we have taken the updated weights from generic NER and further train a spacy transformers model using the L-NER dataset.

3.2.4 Rule based system:

Once we have final predictions from the Spacy Legal NER model, we then use some rules to correct the predictions, those rules are as follows. Through analysis, we have assigned a list of keywords to every entity type present in the dataset. We will be comparing the predictions from the final spacy legal NER model and the predictions from the generic NER model, we will be looking out for the cases where we are missing out on any entity types from the spacy model but which are being predicted by the generic NER model, when these kinds of cases happen, we will try to assign the missed entity type to spacy predictions. For example, given a sentence from the legal judgment: "In The High Court Of Kerala At ErnakulamCrl Mc No. 1622 of 2006()1. T.R.Ajayan, S/O. O.Raman,...PetitionerVs1. M.Ravindran,...Respondent2. Mrs. Nirmala Dinesh, W/O. Dinesh,For Petitioner:Sri.A.KumarFor Respondent:Smt.M.K.PushpalathaThe Hon'ble Mr. Justice P.R.RamanThe Hon'ble Mr. Justice V.K.Mohanan Dated :07/01/2008"

If the entity "T.R.Ajayan" has been missed by the spacy legal NER model, but was recognized as a "PERSON" entity type by the generic NER model, in this case, the detected "PERSON" entity type can be a "PETITIONER" (or) "RESPONDENT" (or) "LAWYER" (or) "JUDGE" (or) "WITNESS" (or) some "OTHER PERSON". In order to decide this entity type, we will be utilizing the context words (or) keywords assigned for each entity type. We will look around the missed entity in the input sentence and check which entity type's keywords match the surrounding context. Based on the keyword similarity score we will be assigning the missed entity type to the spacy predictions.

Along with our proposed main approach, we proposed two baseline models; The first baseline

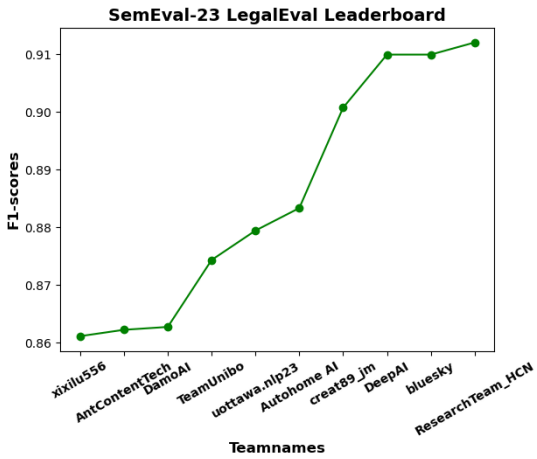

Figure 3: SemEval-23 LegalEval Leaderboard

model is simply fine-tuning our legal LLM using the huggingface transformers (Wolf et al., 2019) RobertaForTokenClassification script. We refer to this model as LegalLM-NER in the following sections. The second baseline model is an ensemble model that concatenates the contextual word embedding generated from our LLM and Kalamkar et al. 2022a model's prediction output and passes it to a classification layer. On top of the training dataset, we also utilized some syntactic datasets for training the 2nd baseline model. Similarly, we refer to this model as Ensembled-Legal-NER. To generate the L-NER synthetic dataset, we extract entities from 200 legal judgment documents obtained by scrapping from the Indian Kannoon website.

This proposed approach has shown significant improvement from the baseline model, which can be confirmed by the leaderboard graph shown in Figure 3.

3.3 CJPE:

The classification task is based on a hierarchical model approach. We have trained our CJPE model using the extended Multi dataset for sub-task 1. In the first step, the Legal pre-trained RoBERTa model is fine-tuned on the multi-dataset. Given a document from a dataset, the last 512 tokens from the sample are extracted and then passed through the legal RoBERTa model and passed through a fully connected layer, and then to the classification layer. Once the model is fine-tuned the weights of the RoBERTa model are updated. In the second stage, the fine-tuned language model is used as a feature extractor and then the features are passed through the BiGRU model and then to a classification layer.

The CJPE sub-task 2 is to extract the most relevant sentences from the document which explains the judgment in the classification task using the model trained in task 1. But we tried solving the problem using the summarizing approach. Here we framed the task as a SEQ2SEQ approach and used a T5-based transformer model to generate the relevant sentences from the given document. We achieved around 45 percent on the given dataset.

4 Data:

We use the LegaEval dataset of the task in SemEval-23. The RR dataset consists of 247 documents having 28986 sentences for training and 30 documents having 2879 sentences for validation. On top of this training dataset, we used a synthesis dataset of around 1000 sentences for low occurrences classes i.e. STA, ISSUE, and PRE_NOT_RELIED. The class distribution for the training and validation dataset is provided in Table 1.

Table 1: Class distribution of the RR dataset

Classes	Training sentences	Dev sentences
ANALYSIS	10695	986
FAC	5744	581
PREAMBLE	4167	509
PRE_RELIED	1431	142
NONE	1423	192
ARG_PETITIONER	1315	70
RPC	1081	92
RLC	752	116
ARG_RESPONDENT	698	38
RATIO	674	72
STA	481	29
ISSUE	367	51
PRE_NOT_RELIED	158	12

The L-NER dataset consists of 1560 preambles and 9435 judgment sentences for training, 125 preambles, and 949 judgment sentences for validation. On top of the provided training dataset, we used 200 more legal documents for training as mentioned in the previous section. The entity distribution for the training and validation is given in Table 2

5 Experimental setup

To train our legal RoBERTa based model, we used the huggingface (Wolf et al., 2019) mask language model training script. We used 5% of the whole dataset as a validation dataset with 15% masking probability. We initialized the model weight from the RoBERTA base model and used the same

Table 2: Entity distribution of the L-NER dataset

Classes	Train	Val
COURT	2367	296
PETITIONER	3068	211
RESPONDENT	3862	315
JUDGE	2325	174
LAWYER	3505	589
DATE	1885	222
ORG	1441	159
GPE	1398	183
STATUTE	1804	222
PROVISION	2384	258
PRECEDENT	1351	177
CASE_NUMBER	1040	121
WITNESS	881	58
OTHER_PERSON	2653	276

RoBERTa base tokenizer. The model was trained for 3 epochs i.e. 600K steps with batch size 16 and learning rate of $1e-5$. The model was trained on 32 GB V100 GPU for 5 days.
We trained multiple models for the different Legal-Eval sub-tasks. All the models use our legal LLM model to extract contextual work representations.

5.1 RR:

We trained the RR-SentenceTransformers model for 26 epochs with a learning rate of $5e{-}5$ on 16GB T4 GPU for 8 hours. We applied a dropout of 0.3 on the contextual word embedding extracted from the SentenceTransformers and BiLSTM layer in a sequential classification block. With the same hyperparameters,the RR-ATTN-POOL was trained for 36 epochs on the same GPU, which took around 10 hours, and the RR-ATTN-POOL-SYNTH model for 52 epochs, which takes around 13 hours. We use $\omega = 0.6$ for all of our experiments in the RR task.

5.2 L-NER:

For the LegalLM-NER model, we finetuned our legal LLM for 4 epochs with RobertaForTokenClassification (Wolf et al., 2019) on 16GB t4 GPU for 3 hours with a learning rate of $5e-5$, batch size 4 and other default hyperparameters provided in the script. The Ensembled-Legal-NER used the same hyperparameters and trained the model for 8 epochs on the same GPU machine for 7 hours. The Generic NER model was trained similarly to the LegalLM-NER model. Our Spacy Legal NER model is trained with the same configuration file provided in https://github.com/Legal-NLP-EkStep/legal_NER

5.3 CJPE:

Classification Task: For the classification sub-task 1, the dataset is split into train tests and valid with 32305,1517 and 994 documents respectively. All the text in the document is lowered for fine-tuning tasks. The stage one fine-tuning model was trained for 5 epochs with a batch size of 8 and learning rate of 2e-6 and max sequence length of 512 We used the PyTorch library to train the deep learning model and the transformers package to extract the embeddings from the large language model. The stage two fine-tuning task is trained for 5 epochs with a batch size of 32 and learning rate of 2e-5 and max sequence length of 512 The F1-score metric is used to evaluate the classification model. Explainability task: The explainability model dataset size is around 50 documents. Here F1-score and ROUGE2 are the metrics used to evaluate the model performance.

6 Results

Our RR models are evaluated on the development dataset using the micro average F1-score. The results are provided in table 3.

Table 3: Evaluation results on RR development dataset

Model	Precision	Recall	F1-score
RR-SentenceTransformers	0.759	0.779	0.76
RR-ATTN-POOL	0.762	0.78	0.764
RR-ATTN-POOL-SYNTH	0.781	0.771	**0.770**

The Legal NER models are evaluated using the standard F1-score used in (Segura-Bedmar et al., 2013). The result is provided in Table 4.

Table 4: Evaluation results on L-NER development dataset

Model	Precision	Recall	F1-score
LegalLM-NER	88.64%	91.96%	90.27%
Ensembled-Legal-NER	90.14%	90.52%	90.33%
Spacy+Generic-NER+Rules	90.48%	90.31%	**90.39**%

Finally, the evaluation result for the CJPE classification task is provided in table 5

Table 5: Evaluation results of our CJPE classification model

Model	Precision	Recall	F1-score
CJPE classification	66.76%	66.65%	66.71%

7 Conclusion

In this work, we proposed solutions for all the tasks in LegalEval a sub-task of SemEval-23. Apart from these solutions, we also proposed an LLM based on RoBERTa base for Indian legal documents. Among our proposed solutions, especially the L-NER proposed models provide the best result, which also stands at the top rank in the L-NER sub-task of SemEval-23.

References

Daniyal Alghazzawi, Omaimah Bamasag, Aiiad Albeshri, Iqra Sana, Hayat Ullah, and Muhammad Zubair Asghar. 2022. Efficient prediction of court judgments using an lstm+ cnn neural network model with an optimal feature set. *Mathematics*, 10(5):683.

Ting Wai Terence Au, Ingemar J Cox, and Vasileios Lampos. 2022. E-ner–an annotated named entity recognition corpus of legal text. *arXiv preprint arXiv:2212.09306*.

Arthur Brack, Anett Hoppe, Pascal Buschermöhle, and Ralph Ewerth. 2021. Sequential sentence classification in research papers using cross-domain multi-task learning. *arXiv e-prints*, pages arXiv–2102.

Yanguang Chen, Yuanyuan Sun, Zhihao Yang, and Hongfei Lin. 2020. Joint entity and relation extraction for legal documents with legal feature enhancement. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1561–1571.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Saptarshi Ghosh and Adam Wyner. 2019. Identification of rhetorical roles of sentences in indian legal judgments. In *Legal Knowledge and Information Systems: JURIX 2019: The Thirty-second Annual Conference*, volume 322, page 3. IOS Press.

Prathamesh Kalamkar, Astha Agarwal, Aman Tiwari, Smita Gupta, Saurabh Karn, and Vivek Raghavan. 2022a. Named entity recognition in indian court judgments. *arXiv preprint arXiv:2211.03442*.

Prathamesh Kalamkar, Aman Tiwari, Astha Agarwal, Saurabh Karn, Smita Gupta, Vivek Raghavan, and Ashutosh Modi. 2022b. Corpus for automatic structuring of legal documents. *arXiv preprint arXiv:2201.13125*.

Kankawin Kowsrihawat, Peerapon Vateekul, and Prachya Boonkwan. 2018. Predicting judicial decisions of criminal cases from thai supreme court using bi-directional gru with attention mechanism. In *2018 5th Asian Conference on Defense Technology (ACDT)*, pages 50–55. IEEE.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Elena Leitner, Georg Rehm, and Julian Moreno-Schneider. 2019. Fine-grained named entity recognition in legal documents. In *Semantic Systems. The Power of AI and Knowledge Graphs: 15th International Conference, SEMANTiCS 2019, Karlsruhe, Germany, September 9–12, 2019, Proceedings*, pages 272–287. Springer.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Vijit Malik, Rishabh Sanjay, Shouvik Kumar Guha, Shubham Kumar Nigam, Angshuman Hazarika, Arnab Bhattacharya, and Ashutosh Modi. 2021a. Semantic segmentation of legal documents via rhetorical roles. *arXiv preprint arXiv:2112.01836*.

Vijit Malik, Rishabh Sanjay, Shubham Kumar Nigam, Kripa Ghosh, Shouvik Kumar Guha, Arnab Bhattacharya, and Ashutosh Modi. 2021b. Ildc for cjpe: Indian legal documents corpus for court judgment prediction and explanation. *arXiv preprint arXiv:2105.13562*.

Oleg Metsker, Egor Trofimov, Max Petrov, and Nikolay Butakov. 2019. Russian court decisions data analysis using distributed computing and machine learning to improve lawmaking and law enforcement. *Procedia Computer Science*, 156:264–273.

Ashutosh Modi, Prathamesh Kalamkar, Saurabh Karn, Aman Tiwari, Abhinav Joshi, Sai Kiran Tanikella, Shouvik Guha, Sachin Malhan, and Vivek Raghavan. 2023. SemEval-2023 Task 6: LegalEval: Understanding Legal Texts. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, Toronto, Canada. Association for Computational Linguistics (ACL).

Shounak Paul, Arpan Mandal, Pawan Goyal, and Saptarshi Ghosh. 2022. Pre-training transformers on indian legal text.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Erik F Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*.

Isabel Segura-Bedmar, Paloma Martínez Fernández, and María Herrero Zazo. 2013. Semeval-2013 task 9: Extraction of drug-drug interactions from biomedical

texts (ddiextraction 2013). Association for Computational Linguistics.

Benjamin Strickson and Beatriz De La Iglesia. 2020. Legal judgement prediction for uk courts. In *Proceedings of the 3rd International Conference on Information Science and Systems*, pages 204–209.

Fernando Trias, Hongming Wang, Sylvain Jaume, and Stratos Idreos. 2021. Named entity recognition in historic legal text: A transformer and state machine ensemble method. In *Proceedings of the Natural Legal Language Processing Workshop 2021*, pages 172–179.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Zhuopeng Xu, Xia Li, Yinlin Li, Zihan Wang, Yujie Fanxu, and Xiaoyan Lai. 2020. Multi-task legal judgement prediction combining a subtask of the seriousness of charges. In *Chinese Computational Linguistics: 19th China National Conference, CCL 2020, Hainan, China, October 30–November 1, 2020, Proceedings 19*, pages 415–429. Springer.

Qihui Zhao, Tianhan Gao, Song Zhou, Dapeng Li, and Yingyou Wen. 2022. Legal judgment prediction via heterogeneous graphs and knowledge of law articles. *Applied Sciences*, 12(5):2531.