TeamUnibo at SemEval-2023 Task 6: A transformer based approach to Rhetorical Roles prediction and NER in Legal Texts

Yuri Noviello and **Enrico Pallotta** and **Flavio Pinzarrone** and **Giuseppe Tanzi**
University of Bologna
{yuri.noviello, enrico.pallotta,
flavio.pinzarrone, giuseppe.tanzi} @studio.unibo.it

Abstract

This study aims to tackle some challenges posed by legal texts in the field of NLP. The LegalEval challenge (Modi et al., 2023) proposes three tasks, based on Indial Legal documents: Rhetorical Roles Prediction, Legal Named Entity Recognition, and Court Judgement Prediction with Explanation. Our work focuses on the first two tasks. For the first task we present a context-aware approach to enhance sentence information. With the help of this approach, the classification model utilizing InLegalBert as a transformer achieved 81.12% Micro-F1. For the second task we present a NER approach to extract and classify entities like names of petitioner, respondent, court or statute of a given document. The model utilizing XLNet as transformer and a dependency parser on top achieved 87.43% Macro-F1.

1 Introduction

The exponential growth of pending legal cases in highly populated countries, such as India, has highlighted the need for automation in the judicial process. Although a full automation of the legal domain may not be feasible, intermediate tasks can be automated to assist legal practitioners and improve efficiency. Nevertheless, the unique nature of legal texts presents challenges for direct application of NLP models and techniques developed for common texts, thus requiring the development of specialized NLP approaches specifically tailored to the legal domain. The objective of the first task is to segment a given legal document by predicting the rhetorical role label for each sentence such as a preamble, fact, ratio, arguments, etc. These are referred to as **Rhetorical Roles (RR)**. This segmentation is a fundamental **building block** for many legal AI applications like judgment summarizing, judgment outcome prediction, precedent search, etc. Legal documents are typically long (tens of pages), unstructured, noisy (e.g., grammatical and spelling mistakes due to manual typing in courts), and use different lexicons (legal jargon), which makes end-to-end transformer-based pre-trained models for sentence classification ineffective. In this paper we propose an approach based on the usage of transformer-based models pre-trained on legal text and fine-tuned on a Rhetorical Roles dataset in a **context aware** way, to make them able to accurately segment even long documents. We first experimented with simple single sentence classification transformers (both pre-trained on general texts and legal texts) and then, leveraging on the fact that in a document nearby sentences classes tend to influence each other, we experimented enriching the sentence representations with information coming from the context. First attaching 2 BiLSTM layers to the transformer embedding model and then simply extending the input tokens of the trasformer by adding the tokens from a context window around the sentence. The second task, legal NER, allows to effectively generate metadata information that can be exploited for many legal application like knowledge graph creation, co-reference resolution and in general to build any query-able knowledge base that would allow faster information access. To address this task, we've experimented two transformers-based approaches with increasing model complexity. We tried to reproduce both the baseline's methods proposed by P.Kalamkar et al.(Kalamkar et al., 2022a). The first method, a simple linear layer as head and fine-tuning of the whole model was easy to reproduce and we were also able to reach a higher score. For the second one, which uses a transition-based parser, we tried to reproduce it stacking on top of the transformer a BiLSTM and a CRF layer as in (Lample et al., 2016; Dai et al., 2019) and in particular as in the work of R.Yan et al.(Rongen Yan, 2021), by the way in this case we were not able to reach the same performance. We experimented different transformers, namely RoBERTa, InLegalBERT (Paul et al., 2022), which is a variant of

Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023), pages 275–284
July 13-14, 2023 ©2023 Association for Computational Linguistics

BERT trained on indian legal documents, and XLNet given it's ability to process longer sequences of tokens. After having tested all of them for the baseline model, the most performing ones (RoBERTa and XLNet) were then used as base for our custom models. All the experiments have been conducted on the datasets provided by the challenge (Modi et al., 2023; Kalamkar et al., 2022a). Only the best models were trained on the whole dataset (including the test set) in order to produce the best results possible for the challenge's submission. With this retraining procedure, the context aware model using InLegalBERT reached a Micro-F1 score of **81.12%** on the submission data of task A, while our custom model reached **87,4%** F1 Score on the submission data of task B.

2 Related Works

Rhetorical Role prediction is a task in NLP aimed at identifying the role played by individual sentences or clauses in a larger discourse structure, such as a news article or scientific paper. This task is important for tasks such as summarization, text generation, and information retrieval. Traditional approaches to Rhetorical Role prediction have relied on hand-crafted features and statistical models such as Conditional Random Fields (CRFs) (Lafferty et al., 2001), and Support Vector Machines (SVMs) (Cortes and Vapnik, 1995).

Recently, deep learning models such as Transformers have shown significant promise in Rhetorical Role prediction. Transformers, introduced by (Vaswani et al., 2017), have shown impressive results in various NLP tasks, including language modeling, machine translation, and sentiment analysis. In Rhetorical Role prediction, the Transformer model is typically used as an encoder to produce contextualized representations of the input text. These representations are then fed into a downstream classification layer to predict the rhetorical role of each sentence or clause. Several studies have shown that using pre-trained Transformer models, such as BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019), can significantly improve Rhetorical Role prediction performance over traditional approaches.

Named Entity Recognition (NER) is a fundamental task in NLP, aimed at identifying and extracting named entities, such as person names, locations, organizations, and other named entities from textual data. Several deep learning models have achieved

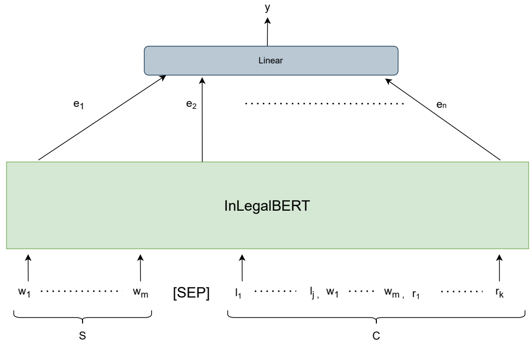


Figure 1: **M2**: Context aware InLegalBERT

state-of-the-art performance in NER, with the introduction of neural network architectures such as Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), and more recently, Transformers. In NER, the Transformer model is typically used as an encoder to produce contextualized representations of the input tokens. These representations are then fed into a downstream classification layer to predict the named entity tags. Furthermore, recent works have explored combining the Transformer model with BiLSTM and CRF to improve the sequence labeling performance. CRF is a probabilistic model that takes into account the sequence of labels and their dependencies, which can help address the label bias problem that arises in token-level classification, while Bidirectional Long Short-Term Memory (BiLSTM) is a neural network architecture that uses two layers of Long Short-Term Memory (LSTM) cells (Hochreiter and Schmidhuber, 1997) to capture both past and future context of each input token. Several studies have demonstrated that incorporating CRF into the Transformer-based NER model can lead to further performance gains.

3 System Description

Task A - Rhetorical Roles prediction

In this task the usual pipeline of an NLP task was followed. During the preprocessing step we focused on the creation of the context's window for each sentence. Starting from the position of the sentence in the document we considered a context window of varying size, such that each time the total number of tokens of the sentence plus the context would not exceed the maximum sequence length of the transformer (usually 512). The context created in this way was then used only in the aforementioned context-aware models.

At a first glance, the Rhetorical Role segmentation task can be seen as a simple text classification problem in which we have to classify each sentence of a document as belonging to one of the Rhetorical Role classes, independently of each other. For this reason we first selected the standard versions of **BERT** and **RoBERTa** as baselines to be tested in their sentence classification setting. In these models a transformer architecture produces the embedding of the sentence, which is then processed through a dropout layer and a linear classifier.
Drawing inspiration from (Malik et al., 2021), we leveraged on the fact that sentences in legal judgements documents follow a tight structure and do not abruptly change RR, indeed the text tends to maintain a topical coherence. Hence, we propose a solution that enables reasoning on the context of the sentences when classifying.
We considered a model that utilizes a pre-trained BERT embedding module. The module is then followed by the integration of two Bi-LSTM layers and a linear classifier, with an average pooling operation performed at token level between the two LSTM layers. We then evaluated two models with the same architecture as the baselines, utilizing pretrained transformers, Legal-RoBERTa [1] (trained on legal documents) and InLegalBert. These models were fed with the target sentence and its surrounding context to enable the BERT modules to generate contextually enriched sentence embeddings, which were then fed into a linear classifier to output classes predictions.
The following naming convention will be used from now on to refer to the described models:

- **B1** : Single sentence BERT
- **B2** : Single sentence RoBERTa
- **M0** : Single sentence InLegalBERT + 2 BiLSTM
- **M1** : Context aware Legal-RoBERTa
- **M2** : Context aware InLegalBERT

Task B - Legal Named Entity Recognition

Unlike many NER applications, where the input text is already in a processed form, our system was designed to handle raw text and perform end-to-end named entity recognition, preserving the original text format. Therefore, no text cleaning was performed (except for RoBERTa, see section 4), and labels were mapped from words to tokens using the **B-I-O** labeling format (Ramshaw and Marcus, 1995). The code for transforming our raw data to a dataset format is an heavy adaptation of the one proposed by S.Subedi(Subedi). Furthermore, given the limited amount of tokens that a transformer can process and the fact that some texts in our dataset were too long (1.44% of samples exceed 512 tokens, with a peak of 5000), we developed a **sliding window** approach, for both the training and inference steps, that allowed us to avoid input truncation.
The sliding window for the training procedure is triggered only if the text provided is longer than the maximum length processable by the transformer; in this case the window will have size equal to the maximum length and it will move with a **stride** that is half of the window size. We use this stride to prevent entities from being cut by the sliding window, ensuring their full appearance at least once. At inference time, the code for the sliding window approach is an adaptation of NLPSandbox[2]. It processes the texts as described above; hence the label assigned to an entity is the one with maximum of the average of the scores given by the inference on each sliding window.
For the baseline architecture, following (Kalamkar et al., 2022a), we used the standard token classification model, so a simple fully connected layer on top of a transformer model (Figure 2a). The model experimented were:

- **BM1** : RoBERTa
- **BM2** : InLegalBERT (Paul et al., 2022)
- **BM3** : XLNet (Yang et al., 2019)

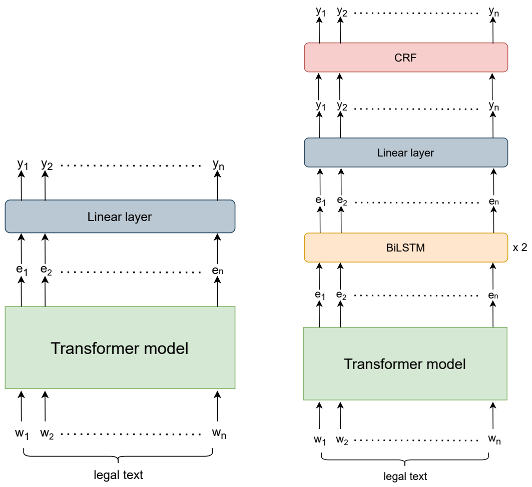
XLNet, contrarily to the first two, can process inputs of any dimension, nonetheless we used 1024 tokens as we were restricted by memory limitations. Regarding the custom models, we mainly experimented two variants using the best performing transformer in the baseline models:

- **CM1** : RoBERTa - BiLSTM - CRF (Dai et al., 2019)
- **CM2** : XLNet - BiLSTM - CRF (Rongen Yan, 2021)(Dai et al., 2019)

They share the same structure, differing only for the transformer backbone: two BiLSTM layers, with a dropout (rate 0.5) between them, and a fully connected layer with a CRF on top (Figure 2b). Other variants using either BiLSTM or CRF were tested but with no notable results.

[1]https://huggingface.co/saibo/legal-roberta-base.

[2]https://github.com/nlpsandbox/phi-annotator-huggingface



(a) **BM**: Transformer + Linear classifier

(b) **CM**: Transformer + BiLSTM + CRF

Figure 2: Architectures of NER models

4 Data

Task A - Rhetorical Roles prediction

The dataset used in this task is the **Corpus for Automatic Structuring of Legal Documents**(Kalamkar et al., 2022b). This dataset contain a corpus of 40,305 sentences annotated with 12 different RRs, as one can see in Table 1. For the purpose of the challenge, smaller annotated datasets were released, divided into a training and a development set of size 28,986 and 2,878, in order to make training and evaluation processes easier. A non-annotated test dataset was also released during the submission phase of the task, and was used to compute the metrics and build a leaderboard. As we can see from Table 1 the class distribution

Class	Count
ANALYSIS	14300
FAC	8045
PREAMBLE	6116
NONE	2037
PRE_RELIED	1934
ARG_PETITIONER	1771
RPC	1562
RLC	1081
ARG_RESPONDENT	1068
RATIO	1014
STA	625
ISSUE	535
PRE_NOT_RELIED	217
TOTAL	**40305**

Table 1: Corpus for Automatic Structuring of Legal Documents

in the corpus is highly imbalanced, reflecting the inherent imbalance in legal documents. These documents consist of long discussions and analysis sections, along with short and standardized formulas for preambles and final judgements. Hence, balancing the class distribution is not a straightforward task.

Task B - Legal Named Entity Recognition

The dataset used in this task is the **Legal Named Entity Recognition Corpus** (more details in (Kalamkar et al., 2022a)). This dataset contains a corpus of 16,570 sentences extracted from Indian court judgements [3] and annotated with 14 entities and roles (described in (Prathamesh Kalamkar)). Sentences can represent either the preamble or a judgement sentence of a court judgement, so different sentences may belong to the same court document. For the purpose of the challenge, a smaller annotated dataset was released. It is divided into a training and a development set of size 10,995 and 1,074, in order to make training and evaluation processes easier. Typically preambles are larger than judgements and they contain more entities. Some entities are present only in the preamble, others only in the judgment text and some in both. Nevertheless, preambles and judgments are treated in the same way, since when we perform inference there is no distinction between them. The dataset conversion to B-I-O format is implemented using a sliding-window approach, since multiple sequences exceed the maximum limit of the tokenizers. The pre-processing step involved in this study was limited to the elimination of multiple white spaces in sentences and it was only performed for experiments utilizing RoBERTa models. The motivation

Named Entity	Found in	Count	Avg.Words
PRECEDENT	Judgment	1351	10
COURT	Both	2367	7
CASE_NUMBER	Judgment	1040	4
RESPONDENT	Both	3862	4
PETITIONER	Both	3068	3
LAWYER	Preamble	3505	2
STATUTE	Judgment	1804	2
PROVISION	Judgment	2384	2
ORG	Judgment	1441	2
DATE	Judgment	1885	1
GPE	Judgment	1398	1
WITNESS	Judgment	881	1
OTHER_PERSON	Judgment	2653	1
JUDGE	Both	2325	1
TOTAL		**29964**	

Table 2: Legal Named Entities Recognition Corpus

[3] https://indiankanoon.org/

is that only RoBERTa tokenizer considers as token multiple spaces, hence if a sentence contains a lot of white spaces the length of the meaningful context is drastically reduced. XLNet and BERT-like tokenizers, instead, do not tokenize white spaces, so removing them will be useless for these models. We have decided not to remove punctuation and to use only case-sensitive models, since many entities contain periods and commas and the capitalization could be a hint for entities like organizations, judges, petitioners and others.

5 Experimental setup and results

Task A - Rhetorical Roles prediction

The experiments of this work were performed on the five aforementioned models. As previously stated, only two annotated datasets were released for this challenge: a training set and a development set. In order to properly test and compare our models we further split the training set into a training and a validation set, while keeping the development set as a test set to compare the generalization capabilities of our models. We used the validation set to perform hyperparameter tuning for all models and then compared the results on the test set of the different models. The models were implemented using the PyTorch framework, through the use of `AutoModelForSequenceClassification` class of the HuggingFace library, which allows loading different transformer models with a linear layer on top. With regards to **M0**, we re-defined the model classes from HuggingFace by adding the BiLSTM between the transformer and the linear classifier. All the models were trained and evaluated using the HuggingFace `Trainer`[4] API to guarantee correctness and reproducibility of the results. The chosen values for the hyperparameters are reported in Table 3. The baselines and model **M0** have been trained with a higher batch size (128) in order to speed up a bit the training process. Unfortunately this wasn't possible with the context-aware models, since their larger input size combined with large batch sizes exceeded the GPU memory provided by Google Colab. All the experiments have been conducted on Google Colab with free plan.

Parameter	Value
epochs	3
batch size	16
learning rate	2×10^{-5}
optimizer	AdamW
scheduler	linear
weight decay	0.01
label smoothing	0.02
precision	16-bit

Table 3: Key training procedure parameters

The results obtained from the validation and test set are shown in Table 4. We used the F1 score micro average (which coincides with accuracy in this multiclass setting) as the main metric, since it is the one that determined the challenge leaderboard. It is clear that InLegalBERT (**M2**) outperforms the other models across all metrics. The baseline models **B1** and **B2** reached Micro-F1 scores of 65% and 72% respectively, with **M0** performing slighlty worse. On the other hand, the use of context-aware models (**M1** and **M2**) results in a 10% improvement in Micro F1, reaching Micro-F1 scores of 80% and 82% respectively.
To submit our results and participate in the challenge, **M2** was then retrained using all the available annotated data (the union of the training and the development set) in order to be sure to maximize performances. The model obtained a Micro-F1 score of 81.12% on the submission data, that resulted in the 8th position in the leaderboard.

Task B - Legal Named Entity Recognition

The results presented in the subsequent section reflect the same configuration used for task A (i.e. further split of the train set and using the validation set as a test set). For the final submission, all the available data was utilized for training without conducting evaluation. The models were implemented using the PyTorch framework, in particular for the baseline model we used the `AutoModelForTokenClassification`

Model	Validation Set			Test Set		
	WP	WR	F1	WP	WR	F1
B1	63.0	65.0	65.0	64.0	65.0	65.0
B2	67.0	69.0	69.0	72.0	72.0	72.0
M0	56.0	63.0	63.0	60.0	65.0	65.0
M1	**77.0**	76.0	76.0	79.0	80.0	80.0
M2	**77.0**	**77.0**	**78.0**	**81.0**	**82.0**	**82.0**

Table 4: Performance on validation and test set

[4] https://huggingface.co/docs/transformers/main_classes/trainer.

class of the HuggingFace library, that allows to load different transformer models with a linear layer on top. Instead, for our custom architecture we created new classes in order to add a BiLSTM and CRF layer. While BiLSTM is commonly included in the standard PyTorch library, the implementation of the CRF we used comes from the `pytorch-crf`[5] library. The model classes from HuggingFace were modified by introducing a BiLSTM layer between the transformer and the linear classifier. The CRF layer then employs the Viterbi Algorithm to calculate the loss and to determine the most probable label. In order to train these models, we used the `Trainer` class of transformers with the parameters described in Table 5. A special treatment was

Parameter	Value
batch size	8 / 4
BM - epochs	3
CM - epochs	5
optimizer	AdamW
learning_rate	5×10^{-5}
scheduler	linear
weight_decay	0.01
warmup_ratio	0.1
precision	16-bit

Table 5: Key training procedure parameters

reserved for RoBERTa based models, since it is trained on clean data, in the end we developed a label remapping function that uses Regex patterns to remap predictions on clean data in the original raw text. This process allowed us to increase performance of the baseline with RoBERTa from 80.08% to 86.7% (shown in the notebook). In accordance with the challenge guidelines, the primary evaluation metric used in this study was the **strict** version of the **F1-score**. For a prediction to be deemed correctly classified, it must not only predict the correct class but also have a perfect match between the predicted entity words and the actual entity words. Apart from that, we also monitored the Precision, Recall and the partial match F1-score, in which just a partial overlap is required. The results obtained on the validation and test set (originally provided as dev-set) are shown in Table 6 and Table 7.

[5]https://pytorch-crf.readthedocs.io/en/stable/

Model	Validation			Test		
	Precision	Recall	F1	Precision	Recall	F1
BM1	81.2	86.5	84.1	84.4	89.0	86.7
BM2	78.0	82.2	80.2	82.0	88.7	85.2
BM3	81.3	**86.8**	**84.0**	84.2	90.1	87.1
CM1	76.0	82.3	79.0	85.5	88.4	87.0
CM2	**85.3**	**86.8**	**84.0**	**85.9**	**90.4**	**88.1**

Table 6: Baseline and custom models' performance on validation and test set

Named Entity	Precsion	Recall	F1 strict	F1 partial
DATE	**96.0**	**97.3**	**96.7**	**98.0**
WITNESS	93.4	98.3	95.8	97.5
LAWYER	95.3	95.4	95.3	97.5
PETITIONER	91.9	96.7	94.2	95.8
JUDGE	93.1	93.7	93.4	96.0
PROVISION	90.2	96.5	93.3	94.9
STATUTE	90.2	95.0	92.5	95.0
COURT	91.2	91.2	91.2	95.3
OTHER_PERSON	88.9	90.2	89.6	96.8
GPE	73.8	89.1	80.7	85.6
CASE_NUMBER	72.8	88.4	79.9	83.6
RESPONDENT	78.0	80.0	79.0	89.2
PRECEDENT	68.0	76.8	72.1	83.3
ORG	60.2	68.6	64.1	75.0
ALL	**85.9**	**90.4**	**88.1**	**92.7**

Table 7: Best model entity-wise performance on test set

Model	F1
BM1	84.75
BM2	82.32
BM3	86.04
CM1	85.80
CM2	**87.43**

Table 8: Models performance on submission set

6 Discussion

Task A - Rhetorical Roles prediction

Looking at the results in Table 4 we can see that the context-aware models clearly outperform the baseline models, since they leverage on context enriched sentence embeddings to perform predictions. The introduction of the context as input enabled the model to gather more information from the semantic meaning of the surrounding sentences, that resulted in more effective sentence embeddings when predicting the Rhetorical Role. Model **M0**, instead, reported similar scores with respect to the baselines, probably due to the fact that the enrichment that a sentence embedding receives comes only from the sentences in the same batch, which are not necessarily equally distributed around the sentence. On the other hand, adding context passages to the input drastically increases the input size of the model to its maximum, which directly affects performances in terms of longer training and inference times. Nonetheless, in a practical use case, when processing a document this operation will be performed just once at the beginning of a longer pipeline. Among the two context-aware models the best one resulted to be the one using InLegalBERT, which is referable to the fact that it has a better understanding of the specific jargon used in indian legal documents, which it was pre-trained on. Analysing the predictions and the errors of this

Class	Precision	Recall	F1	Support
PREAMBLE	**98.0**	**100.0**	**99.0**	507
NONE	96.0	86.0	91.0	190
RPC	83.0	92.0	88.0	92
FAC	81.0	89.0	85.0	581
ANALYSIS	78.0	90.0	84.0	985
ISSUE	66.0	78.0	71.0	51
ARG_RESPONDENT	64.0	74.0	68.0	38
STA	50.0	68.0	58.0	28
PRE_RELIED	74.0	44.0	55.0	142
RATIO	70.0	37.0	48.0	71
RLC	66.0	27.0	38.0	116
ARG_PETITIONER	57.0	18.0	28.0	65
PRE_NOT_RELIED	0.0	0.0	0.0	12
Weighted	**81.0**	**82.0**	**81.0**	**2878**

Table 9: **M2** - Classification report on the test set

model we encountered the same trends reported in (Kalamkar et al., 2022b). From the classification report in Table 9 we can see that PREAMBLE sentences are almost perfectly classified, probably due to the fact that they almost always include proper nouns, repeated formulas, dates and other highly recognizable patterns. Some other classes like FAC, NONE, ANALYSIS and RPC reach high F1-score values. It is important to note that in the RPC (Ruling by Present Court) class the model has a recall of 92%, meaning that it correctly identifies the final judgement of the court in most of the cases (even if it has a small support in the dataset). On the other hand the model performs no predictions for the class PRE_NOT_RELIED, probably due to the very small support in the dataset. Examining the precision and recall scores for the majority class (ANALYSIS), it can be concluded that the model tends to default to this class when uncertain. This has resulted in significant misclassification for classes with low recall scores, with the exception of RLC, which was more frequently misclassified as FAC, as also happened to human dataset annotators (Kalamkar et al., 2022b). This behaviour is probably related to the fact that sentences belonging to the core section of a document of classes PRECEDENT, ANALYSIS and ARGUMENTS all contain discussions of the court or statements from the parts, which only differ for the subject of the discussion (whether it is a precedent case or not) or for the part that is speaking (whether it is the petitioner or the respondent). A potential solution to the problem at hand is the use of a multilabel setup, which would allow the model to predict multiple classes for each sample. With this approach, if the model predicts the majority class, further investigation can be conducted on the other predicted classes to improve or correct the prediction.

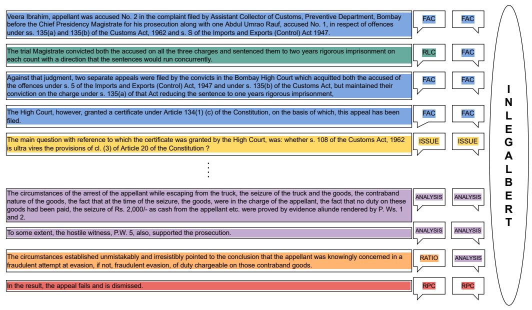
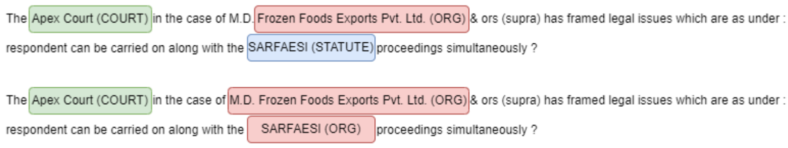


Figure 3: Example of a segmented document

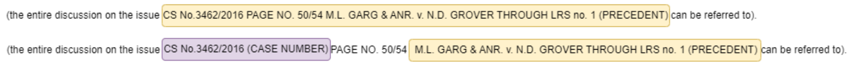
Examining the output example in Figure 3, it is clear that InLegalBert performs well overall. However, it may struggle in predicting the rhetorical role of a sentence if the surrounding context has a different Rhetorical Role. Additionally, it can be observed that sentences labelled with less frequent labels, such as RATIO, are often misclassified as the majority class, ANALYSIS. This tendency of the model to default to the majority class label highlights the need for further improvement.

Task B - Legal Named Entity Recognition

From the results in Table 6 it's clear that the best base for our architecture was XLNet, since it outperformed InLegalBERT and RoBERTa in both baseline and custom models. The results of our best models (Table 8) for the challenge's submission are slightly worse but in line with the performance on our test set. Comparing them with the one obtained by the challenge's organizers (Kalamkar et al., 2022a) we can claim a +5% on their Transformer+Linear model (86% vs 81%), but a -3.6% on their RoBERTa + Transition based dependency parser (87.43% vs 91.1%). The comparison between the first type of architecture holds, as they are largely equivalent. The improvements observed



(a) **ORG**: capitalization confuses the model.



(b) **PRECEDENT**: the case number is not aggregated with the precedent it's referring to.

Figure 4: Examples of typic errors, others are reported in the notebook

in our study may be attributed to differences in training settings. However, such a conclusion cannot be drawn for the second type of architecture, as the details regarding the implementation of the Transition-based dependency parser are limited. The use of BiLSTM and CRF on top led to slight improvements, similarly to the results of the ablation study of (Rongen Yan, 2021). The entity-wise performance were also very predictable, the best F1-scores are for classes marked by discriminating words in their immediate context, like **dates** that present a typical structure, or **lawyers** that are usually introduced by *"Av."*. The worst performance has been for the organization entity (**ORG**). In our notebook we have deeply analyzed some errors for this entity class and we found some typical trends in them. Organizations are usually reported with acronyms in upper case. Since our model is case sensitive, this information is crucial and in some cases could lead to mislabeling (Figure 4a). Long names for organizations are problematic. The presence of punctuation leads the model to split the entity labeling. When ORGs are followed by reference to a law/article/case they can be confused with PRECEDENT that often presents similar structure. A sequence of capitalized words can also be confused with the name of an organization. Since a prediction to be considered as correct must have the exact boundaries and class, longer entities are intrinsically more complex and so they have lower scores: a clear example is PRECEDENT which has on average 11 words in our test set. Beside that, as discussed by the challenge's organizers in (Kalamkar et al., 2022a), a lot of errors are due to the fact that our model doesn't have a global view of the document, but only of a standalone judgement sentence or preamble.

Therefore, the model does not consider information typically included in the preamble of court judgments when predicting entities in a sentence. This leads to frequent errors, such as mistaking a PRECEDENT for a CASE NUMBER (as shown in Figure 4b) or person's roles in court judgments. To reduce verbosity, court judgments only fully quote the precedent the first time, subsequently citing it with just the case number or the first person/institution involved. To address these issues in future work, we could implement a post-processing step (Kalamkar et al., 2022a) or find a way to incorporate document-level context into our model.

7 Conclusion

Given the significance of automating intermediate tasks within the judicial process in the face of growing numbers of pending legal cases in populous countries, the LegalEval challenge calls for the development of NLP techniques tailored to the legal domain.

For **Task A**, we proposed a transformer-based approach for classifying rhetorical roles in legal documents. Our experiment showed that fine-tuning transformer-based models pre-trained on legal text in a context-aware manner led to improved performance in this task. This was found to be more effective compared to the simple single sentence classification transformers pre-trained on either general or legal texts. Thanks to this approach, the classification model utilizing InLegalBert as a transformer achieved 81.12% Micro-F1 on the hidden test set. However, the model shows limited discriminating capacity with respect of some minority classes. The imbalance in the class distribution is likely due to the nature of legal documents, which contain long discussions and statements by the court and parties in the core sections of the document. A potential solution to this challenge is to use a multilabel setup, which would enable the model to predict multiple classes per sample. This would allow for further

investigation of the predicted classes.
For **Task B**, we proposed a transformer-based approach with a dependency parser to detect and classify entities in legal documents. Our experiments showed that fine-tuning a transformer-based model with a simple classification head is already a good solution for this NER task, but adding a dependency parser on top can be useful to increase the performances. Indeed our custom model reached an higher score, but as expected, considering the long training and computational power availability for the work of (Kalamkar et al., 2022a), our approach was not able to get too close to the results of the best proposed baseline. Beside that, a main limitation of our model was that it doesn't process an entire document but only a single sentence. In their paper the organizers proposed some post-processing steps to consider document level context, nonetheless we didn't work on that since the true goal of this competition was to develop an end-to-end neural architecture. As future improvement for our model we could experiment some techniques to inject document level information when processing a single sentence.
To conclude, an ideal evaluation pipeline for legal documents could entail a sequential execution of sentence classification and Named Entity Recognition, utilizing the predicted Rhetorical Role to enhance the performance of the latter.

Links to external resources

- LegalEval challenge website
- LegalEval challenge leaderboard
- Task A: RR Segmentation Training set
- Task A: RR Segmentation Development set
- Task B: Legal NER Training set
- Task B: Legal NER Development set
- GitHub repository: code

References

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.

Zhenjin Dai, Xutao Wang, Pin Ni, Yuming Li, Gangmin Li, and Xuming Bai. 2019. Named entity recognition using bert bilstm crf for chinese electronic health records. pages 1–5.

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

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

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

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

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. pages 260–270.

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. *CoRR*, abs/1907.11692.

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

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.

Aman Tiwari Prathamesh Kalamkar, Astha Agarwal. Repo of the paper "named entity recognition in indian court judgments".

Lance A. Ramshaw and Mitchell P. Marcus. 1995. Text chunking using transformation-based learning.

Depeng Dang Rongen Yan, Xue Jiang. 2021. Named entity recognition by using xlnet-bilstm-crf. *Springer Nature 2021*, 53:3339–3356.

Sanjaya Subedi. Training named entity recognition model with custom data using huggingface transformer.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding.