LegalNER: Understanding Legal Texts

Nidhir Bhavsar
Universität Potsdam,
Matrikel-Nr.: 819171
nidhir.bhavsar@uni-potsdam.de

Abstract

This paper presents an elaborative report on the experiments conducted by team VASPOM for the SemEval task of LegalNER. We propose a multi-model architecture that utilizes two distinct models trained on either i) the 'Judgement', or ii) the 'Preamble' dataset. Our proposed modeling strategies include an ensemble-based procedure consisting of two base XLM-RoBERTa models trained on the LegalNER dataset. While one of the models uses a normal training procedure, the other utilizes a linear-chain conditional random field model. This paper presents crucial discoveries to enhance the robustness and feasibility of the task. The findings are derived to aid in achieving this goal. We make our findings accessible here[1].

1 Introduction

The issue of high case pendency in populous countries such as India has been a longstanding challenge that significantly affects the judicial system's efficiency and effectiveness. According to the National Judicial Data Grid (NJDG) data released on February 1, 2023, there are approximately 59,87,477 pending cases in high courts across the country. To reduce this backlog and improve efficiency, innovative solutions are necessary, such as utilizing Artificial Intelligence (AI). The main goal of the SemEval 2023 task 6; LegalEval: Understanding Legal Texts[2], aims to facilitate the processing and analysis of legal texts, ultimately contributing to the reduction of strain on the judicial system. Out of the 3 available tracks offered under the task we focus on Subtask B; Legal Named Entities Extraction (`LegalNER`). We propose a model that can efficiently label legal text across 14 different tags/classes, distributed across 2 distinct datasets; i) Preamble, ii) Judgement.

Our approach involves utilizing both transformer-based (Vaswani et al., 2017) models and statistical methods to tag words in the legal text with their associated classes. We experiment with various transformer architectures and statistical methods to achieve the best results across all categories. Eventually, we choose InLegalBERT (Paul et al., 2022) and XLM-Roberta (Conneau et al., 2019) as our final models for encoding legal text. To model the distribution for different classes, we use a Conditional Random Fields model (Lafferty et al., 2001), which unlike the standard classifier considers context from both the encoded information and neighboring samples. Additionally, we propose an Ensemble architecture that combines a CRF model with a simple Classifier to produce optimal results for our approach.

Our experiments reveal that certain classes have an indefinite structure and unpredictable nature, leading to a decrease in the overall performance of the model. Furthermore, we observed a significant difference in the structure and presence of similar classes across the datasets. Therefore, combining the datasets would only result in a further decrease in the performance of individual classes and would not facilitate the transfer of information across different data types.

2 Related Work

Named Entity Recognition (NER) has been the subject of extensive research in a variety of domains, ranging from traditional statistical models (Mikheev et al., 1999) to advanced deep neural networks (Devlin et al., 2018). There have been several approaches to training deep-neural networks on domain-specific data. This is demonstrated by the SciBERT (Beltagy et al., 2019) model, which is explicitly trained on scientific data and thus aids in knowledge-based solutions. In the legal domain, NER has been used successfully to aid in the ex-

[1] https://gitup.uni-potsdam.de/vaspom/anlp-legalner

[2] https://sites.google.com/view/legaleval/home

traction of meaningful information from legal texts. One such approach is towards the pre-training language model on a corpus of 12 million legal documents (Chalkidis et al., 2020). This has been shown to outperform the standard BERT model on the contracts-ner (Chalkidis et al., 2017) dataset.

Additionally, there have been numerous contributions towards creating legal corpora. For example, the German legal corpus (Leitner et al., 2020) is a human-annotated dataset with 19 fine-grained entity classes. Another example is the E-NER dataset (Au et al., 2022), which consists of 52 filings from the US SEC EDGAR database. These datasets can be valuable resources for researchers in the legal NLP field.

Recently, transformer-based (Vaswani et al., 2017) architectures have had a significant impact on the field of NLP. Some derivative transformer-based encoders, such as BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019), have demonstrated off-the-shelf performance on a variety of NLP tasks, including NER. Concurrent input processing, together with the self-attention mechanism, helps to overcome the limitations of sequential processing and aids in contextualization.

Transformer models trained on multilingual data such as XLM (Lample and Conneau, 2019), can be used for cross-lingual tasks. XLM-RoBERTa (XLM-R) is a language model that combines the XLM and RoBERTa architectures and delivers state-of-the-art performance across diverse languages. In the specific domain of LegalNER, which entails handling Hindi-rich lexicons and their fusion with the legalistic style of English, XLM-R's adeptness in Hindi, a high-resource language, renders it an ideal solution for comprehending Indian legal textual content. Next, Paul et al. (2022) gathers a vast collection of legal case documents from the Indian Supreme Court and various High Courts of India. and pre-train a BERT-based architecture. Doing this, they achieve state-of-the-art performance on ISLI (Paul et al., 2021) (Indian legal statute identification) and ECtHR-B (Chalkidis et al., 2021) datasets.

Recent works explored the use of combining embeddings retrieved from deep-NN with a CRF model. Huang et al. (2015) uses a BiLSTM architecture in combination with CRF for tasks such as POS tagging, chunking, and NER. Castro et al. (2019) employed ELMO (Peters et al., 2018) embeddings, which are a fusion of character-level features obtained through convolutional neural networks and the hidden states of each layer in a bidirectional Language Model (LM) that consists of BiLSTM (Graves et al., 2005) models. For Portuguese entity recognition, Souza et al. (2019) blends contextual embedding from a BERT model with a CRF model.

ML techniques like ensemble learning combine the predictions of various models to generate an adept learner. This approach works best when the weak learners are diversified and excel at particular tasks where other models could falter. There have been numerous attempts to use the Ensembling procedure for NER. Nayel and Shashirekha (2017) train four different support vector machine(Hearst et al., 1998) classifiers using distinct segmentation techniques, and combine them through either voting or stacking methods. Also, combining transformer-based model predictions has been shown to increase NER accuracy.

3 Task Formalisation

The LegalNER task in SemEval 2023 poses the problem to model a system that can analyze individual sentences extracted from Indian court judgments, available either via the "Judgement", or "Preamble" dataset. The model should be able to infer from a set of predefined entities and assign an entity to a span of tokens from the representative input sentence by taking into account that the token set the entity is depicting is appropriately aligned with the true nature of the entity class/tag under consideration. The task also constraints that the model just have the input phrase to infer upon and that no further context is given from any other external source.

This entails to the task of Named Entity Recognition which includes recognizing tags unique to a sequence of tokens that point to a predefined categorical distribution. Under ideal conditions, the distribution for each label category should be non-overlapping. E.g. given a sequence of input tags $X = \{x_1, x_2, \ldots, x_n\}$, the task is to tag each token to a specific entity type, say $\{\hat{y_1}, \hat{y_2}, \ldots, \hat{y_n}\}$, where each $\hat{y_i} \in \{y_1, y_2, \ldots, y_k\}$ respectively.

As shown in equation 1, for a given label y_i we model a distribution that defines a probability based on the score derived from local features. Here $\vec{f} \in R^m$ is a m-dimensional vector, composing of features functions $f_1, \ldots, f_m$. And θ are the weights used for mapping the features to a scalar

value. As depicted in Equation 2, we compute the joint probability of each token belonging to a particular class and multiply the scores across all tokens to generate a final score for the given input sequence. Finally, we predict the optimal label sequence by maximizing the probability over all possible label sequences, as shown in Equation 3

4 Data

We utilize the LegalNER dataset (Kalamkar et al., 2022), which was made available by the authors of SemEval 2023 task 6 subtask B. The data presented under this task is a sample representation of Indian high court and supreme court judgments. The dataset is divided into varied types, this includes i) Judgement, and ii) Preamble. This is because, under Indian law, a court decision typically falls into one of the aforementioned categories. The "Preamble" of the judgment provides structured metadata such as the names of the participating parties, judges, lawyers/attorneys, date, court, etc. The paragraph after the prologue to the completion of the court judgment is referred to as the "Judgement." In the given LegalNER dataset, there are 14444 Indian court judgment sentences and 2126 judgement preambles annotated with 14 different legal entities.

For each court judgment, there is no explicit portrayal of the boundary between the "Preamble" and "Judgement." The authors consequently relied on the term JUDGEMENT or ORDER to distinguish the court judgment between the two types. If these keywords were not identified, they planned to locate the first consecutive occurrence of a phrase with a verb to determine the boundary for the "Judgement" because the "Preamble" merely constitutes metadata and not grammatically complete sentences.

The authors carried out an extensive procedure to formulate the LegalNER dataset. They intend to extract highly cited court judgement, thus portraying an ideal sample distribution for the dataset. However, they also define 8 case types, which include i) Tax, ii) Criminal, iii) Civil, iv) Motor Vehicles, v) Land & Property, vi) Industrial, & vii) Labour. They define keywords for each of the above-given categories. These keywords are then used to extract court judgement from the IndianKannon[3] website, which contains a database of Indian legal cases. They consider judgments that are in the English language. Based on the keywords-based searching, the derived dataset is then distributed among the "train", "test" and "validation" sets. They use chronological-spans as the key to separate the data (See table 1). Due to the absence of ground-truth labels for the test dataset, we demonstrate our model's performance on the validation set. Additionally, since they span the same time period, there shouldn't be many variations in the quality of the samples in both sets.

The authors used a spacy pretrained model with custom rules to process the raw text judgments and produce entity-rich phrases. Following that, sentence-level annotations were performed on the judgment text without document-level contexts. Four legal experts and four data scientists worked together to annotate the data. In table 2, a thorough explanation of each entity type in the LegalNER dataset is presented. As indicated, the "Preamble" dataset has 5 entity tags, which are likewise included in the "Judgement" dataset with the exception of the tag LAWYER. Table 7 displays the total number of entities-tags present in the "Judgement" and "Preamble" train data.

	Time Range	**Preambles**	**Judgment sentences**	**Entities**
train	1950 to 2017	1560	9435	29964
validation	2018 to 2022	125	949	3216
test	2018 to 2022	441	4060	13365

Table 1: Shows describe the frequency of the "Judgment", & "Preamble" type under the LegalNER dataset. It also presents the chronological distribution of the dataset into "train", "test", & "validation" sets, along with the total entities among each set.

As outlined in section 3, named entity recognition is a sequential prediction task where the objective is to assign appropriate labels to each token in a given input sequence. To represent these sequence labels, various standard tagging methods can be employed. In this regard, we adopt the BIO and BIOES tagging schemes to formalize our labels according to the input sequence, while also taking into account the position of the target entities and encoding them correctly based on a well-defined tagging procedure.

[3] https://indiankanoon.org/

$$p\left(y_i \mid y_{i-1}, x_1 \ldots x_n\right) = \frac{\exp\left(\vec{\theta} \cdot \vec{f}\left(y_{i-1}, y_i, x_1 \ldots x_n\right)\right)}{\sum_{y'} \exp\left(\vec{\theta} \cdot \vec{f}\left(y_{i-1}, y', x_1 \ldots x_n\right)\right)} \tag{1}$$

$$s\left(y_1 \ldots y_n, x_1 \ldots x_n\right) = \prod_{i=1}^{n} \frac{\exp\left(\vec{\theta} \cdot \vec{f}\left(y_{i-1}, y_i, x_1 \ldots x_n\right)\right)}{\sum_{y'} \exp\left(\vec{\theta} \cdot \vec{f}\left(y_{i-1}, y', x_1 \ldots x_n\right)\right)} \tag{2}$$

$$y_1^* \ldots y_n^* = \arg\max_{y_1 \ldots y_n} s\left(y_1 \ldots y_n, x_1 \ldots x_n\right) \tag{3}$$

4.1 BIO

The BIO tagging technique labels each token in a sequence with one of three tags: Beginning (B), Inside (I), and Outside (O). The label "B" is allocated to an entity's initial token, "I" to succeeding tokens inside the entity, and "O" to all tokens that do not belong to any entity.

4.2 BIOES

The BIOES tagging technique is an expansion of the BIO tagging approach and comprises five tags: Beginning (B), Inside (I), Outside (O), End (E), and Single (S). The labels "B" and "I" are used in the same way as in the BIO tagging technique, while the label "E" is allocated to the last token of an entity, and the label "S" is applied to a single-token entity.

5 Experiments

This section depicts the many approaches and the various experimental decisions undertook by us in order to obtain conclusive results. We also analyze the outcomes and compare the training methodologies, which leads to greater applicability of the task under discussion. As previously stated, there are two major sub-categories of datasets on which we demonstrate the performance of our suggested models and architectures, this include i) Judgement, ii) Preamble. Next subsection discusses about the various adaptation we choose to get optimal results on the aforementioned dataset.

5.1 Methodology

We employ multiple approaches toward modeling an optimal procedure to tackle the task of Legal-NER for tagging legal text sentences. We investigate transformer-based encoder models such as InLegalBERT and XLM-RoBERTa for extracting contextual embedding. Using these context-aware representations, we deploy five distinct processing strategies and likelihood maximization processes to arrive at the ideal modeling configuration that can tag data with the highest accuracy. These modeling types include training a transformer-based encoder in conjunction with a set of linear projection layers. We also train a model that incorporates a transformer-encoder followed by a Conditional Random Fields (CRF) model, thus attaining to both the input features and previously predicted labels through its transition matrix. We obtain different results depending on the tagging process, i.e. BIO or BIOES, due to our ability to train two independent models with different ways of encoding target labels. We also make an effort to address low performance in minority classes. Finally, we conclude our experimentation with an Ensemble model that employs two distinct types of integrating strategies to predict relevant labels for input tokens.

5.1.1 XLM-R: Token Classification

We experiment with the Cross-Lingual version of pre-trained RoBERTa which is another transformer-based architecture. XLM-R[4] is trained extensively on a total of 88 languages with each language accounting to be in GBs of textual data spread across datasets like CommonCrawl. The model shows the off-the-chart performance of the GLUE (Wang et al., 2018) benchmark and shows distinguishing results on datasets like XNLI (Conneau et al., 2018), or NER dataset like CoNLL03 (Tjong Kim Sang and De Meulder, 2003). For the purpose of this task, we finetune XLM-R on the LegalNER dataset for Named Entity Recognition.

We enable the XLM-R model's output to pass through a series of linear transformation layers, mapping these embeddings to a vector reflecting the activation scores for each set of unique labels in the given dataset form. The model is then tuned in favor of the target labels using the standard CrossEntropy Loss function and likelihood optimization techniques in order to attain the best state for the given system.

[4] https://huggingface.co/xlm-roberta-base

Named Entity	Extract Form	Description
COURT	Preamble, Judgment	Name of the court which has delivered the current judgment if extracted from the preamble. Name of any court mentioned if extracted from judgment sentences.
PETITIONER	Preamble, Judgment	Name of the petitioners/appellants/revisionist from current case
RESPONDENT	Preamble, Judgment	Name of the respondents/defendants/opposition from current case
JUDGE	Preamble, Judgment	Name of the judges from the current case if extracted from the preamble. Name. of the judges of the. current as well as previous cases if extracted from judgment sentences.
LAWYER	Preamble	Name of the lawyers from both the parties
DATE	Judgment	Any date mentioned in the judgment
ORG	Judgment	Name of organizations mentioned in text apart from the court.
GPE	Judgment	Geopolitical locations which include names of states, cities, villages
STATUTE	Judgment	Name of the act or law mentioned in the judgement
PROVISION	Judgment	Sections, sub-sections, articles, orders, rules under a statute
PRECEDENT	Judgment	All the past court cases referred to in the judgement as precedent. Precedent consists of party names + citation(optional) or case number (optional)
CASE_NUMBER	Judgment	All the other case numbers mentioned in the judgment (apart from precedent) where party names and citation is not provided
WITNESS	Judgment	Name of witnesses in current judgment
OTHER_PERSON	Judgment	Name of all the persons that are not included in the petitioner, respondent, judge, and witness

Table 2: Contains a detailed description of entity tags defined under the SemEval 2023, task 6, Subtask B LegalNER dataset. There are 14 distinct entities spread across the 2 different dataset types; "Judgement", and "Preamble".

5.1.2 InLegalBERT: Token Classification

InlegalBERT[5] is a modified version of BERT with the same pre-training methods. However, it differs from the conventional model as it is solely trained on legal data. The training process uses two existing BERT-based models, Legal BERT and CaseLawBERT (Zheng et al., 2021), which are pre-trained legal domain encoder models. These models are trained on an Indian judicial judgment dataset, derived from the IndianKannon website

Similar to XLM-R, We adopt a similar procedure of transforming the contextual embedding produced by the InLegalBERT via a series of linear layers, to output a score reflecting the probability of the label being assigned to the specific token in the input sentence.

5.1.3 XLM-R with linear chain CRF

In this setup, we enhance the XLM-RoBERTa model by including a linear chain Conditional Random Field (CRF) layer. The primary goal of this integration is to allow the CRF layer to capture the sequential relationships between the labeled entities, as well as their proportionality with incoming text data. The CRF layer receives logits from a linear transformation and models the likelihood of a series of labels given the input text. The CRF layer, in particular, calculates a score for each potential label sequence based on a transition matrix and a collection of feature functions that capture the local context of the current token and its surrounding tokens. The latter is often accomplished by utilizing a contextually aware text encoder (e.g. XLM-RoBERTa). The score for each label sequence is used to compute the probability of that sequence and is used in the training and decoding of the model. The CRF layer enhances the model's overall performance by ensuring label consistency and capturing the global context of the input sequence. (See section A.4 in Appendix)

[5] https://huggingface.co/law-ai/InLegalBERT

5.1.4 Ensemble XLM-R

Here we employ an ensembling procedure that combines the predictions of multiple base models to produce the final prediction. The main purpose of using the ensembling procedure is that we intend to identify the most promising aspect of each of the constituent base models and fuse them together thus deriving an ideal outcome for the given task in general. To perform ensembling, we choose to adopt 2 integration strategies specifically applied over logits produced by each base model under consideration. This includes i) soft-voting (weighted average), and ii) max-voting (mode). Both of these techniques have their own advantages and are discussed later. We choose to use a configuration of 2 previously finetuned/trained versions of the XLM-R model, this includes i) XLM-R finetuned using normal CrossEntropyLoss, and ii) XLM-R extended to a linear-chain Conditional Random Field layer (CRF).

In addition, we use various postprocessing approaches to align the logits for the aforementioned base models. We choose the XLM-R model for this ensemble configuration because the transformer model utilizes a word-piece tokenization process to translate the input phrase to its initial token embeddings. Out-of-vocabulary terms are often tokenized into smaller chunks of character-set that are recognized by the model's vocabulary. Since the tokenizer produces input tokens based on the tagging process used, aligning the target entity labels to the input tokens becomes necessary. This is why integrating two transformer models with different vocabularies is difficult.

5.1.5 Additional Experimentation

Apart from the modeling techniques mentioned above, we also experiment with a few other additional configurations, some as a part of the preliminary process and others as variants to the above-mentioned arrangements.

Preliminary approach: Initially, we attempted to train a token classification model using the DistilBERT transformer-based encoder. DistilBERT is a compressed version of the original BERT model, that achieves comparable performance on certain NLP tasks, thus being a potential alternative to the larger models. Next, as part of the initial approaches we also train a similar system proposed by Giorgi et al. (2019). Since the task involves two distinct datasets, we attempted to tackle both simultaneously, such that the model tags sequences while also detecting the format of the input text, posing as a multi-task strategy.

Weighted CrossEntropy loss: we try to address the issue of class imbalance in the "Judgement" dataset, where minority classes such as JUDGE, PETITIONER, and RESPONDENT are severely underrepresented compared to majority classes, leading to a decrease in the overall performance of our system. To tackle this, we train a transformer-based model using a weighted cross-entropy loss function, with weights calculated by applying a scaling operation to the counts of each entity class in the dataset.

Tagging techniques: We apply BIO and BIOES tagging techniques to target labels and dynamically align them with model-specific tokenizers for training.

Finally, we also create an additional dataset that combines both the "Judgement", and "Preamble" datasets. We train multiple model configurations using the generated dataset. See Table 9 in Appendix.

5.2 Training

We perform training of the above-mentioned configuration on the LegalNER dataset, which includes i) Judgement, ii) Preamble, and iii) Combined. For training, we standardize each modeling approach with a closely similar hyperparameter setting as mentioned in table 8 in Appendix. We train all our models on standard Google Colab[6] using a cuda enabled Tesla T4 GPU with a memory availability of 15360MiB. For training, computationally expensive models such as XLM-R extended with CRF, we use the Google colab "pro", which avails us of 90 compute units of high-resource Google cloud VMs.

The algorithm 1 described below, shows the general training procedure adopted by us. The given steps can be standardized towards any of the described approaches under section 5.1. However, for model configurations that extend to the CRF layer, we utilize a different loss function and prediction procedure.

6 Results

We use the standard F1 metrics, as well as precision and recall, to compare the performance of our proposed approaches. We utilize the seqeval[7]

[6] https://colab.research.google.com/
[7] https://pypi.org/project/seqeval/

Algorithm 1: Training and Prediction Algorithm for Named Entity Recognition

Input: Training data $\mathcal{D}$, Test data $\mathcal{D}'$, Model M, Loss function $\mathcal{L}_{CE}$, Optimizer O, Number of epochs E
Output: Trained model M
for e in $1, \ldots, E$ **do**
  for $\boldsymbol{x}, \boldsymbol{y}$ in $\mathcal{D}$ **do**
    $\hat{\mathbf{y}} \leftarrow M(\mathbf{x})$; // Forward pass
    $\mathcal{L} \leftarrow \mathcal{L}_{CE}(\hat{\mathbf{y}}, \mathbf{y})$; // Calculate loss
    O.zero_grad() ; // Zero out gradients
    $\mathcal{L}$.backward() ; // Backward pass
    O.step() ; // Update parameters
  end for
end for
for $\boldsymbol{x}$ in $\mathcal{D}'$ **do**
  $\hat{\mathbf{y}} \leftarrow M(\mathbf{x})$; // Forward pass
  $\hat{\mathbf{y}} \leftarrow \text{softmax}(\hat{\mathbf{y}}, dim = 1)$; // Apply softmax function along dimension 1
  $\hat{\mathbf{y}} \leftarrow \text{argmax}(\hat{\mathbf{y}}, dim = 1)$; // Apply argmax function along dimension 1 to get predicted labels
  // Do something with predicted labels :)
end for

python package to derive our results. Furthermore, because ground-truth labels are not available for the test dataset in the LegalNER task, we deliver our results on the validation/development set. Additionally, as shown in table 1, since the test and dev sets are derived from the same chronological period, the results achieved on the dev set should be comparable to the model's performance on the test data.

The results of the original models and their associated strategies trained on the Judgement dataset are described in Table 3. These strategies include training models with BIO or BIOES tagging and using XLM-RoBERTa or InLegalBERT for encoding input textual information. Upon examining the table, it becomes apparent that the majority of models can correctly tag the DATE entity class. This is due to the ease with which DATE can be distinguished from other entity classes.

In contrast to the adversarial XLM-R-based models, all of the modeling configurations that utilize InLegalBERT exhibit lower performance. Upon close examination of some of the confusion plots (see figure 5a, 5c), it becomes evident that InLegalBERT-based models frequently fail to tag full entities, making them vulnerable to type matching errors.

The XLM-R-based model efficiently extracts short entities such as WITNESS, PROVISION, STATUTE, LAWYER, and COURT. In particular, InLegalBERT somehow exhibits degradation in performance when using BIOES-based tagging. Even though we expected of getting better results since BIOES tagging utilizes separate tags to mark single and multi-word entities.

We tried to enhance the model's overall performance by utilizing weighted cross-entropy loss. Our attempt resulted in conclusive outcomes, specifically for the PETITIONER category, which belongs to the three underrepresented classes in the 'Judgement' data. We obtained a relatively high f1 score of 0.744. However, this approach caused some majority classes to underperform. Despite implementing weighted cross-entropy loss, there was no noticeable enhancement for the RESPONDENT entity class.

We observed that the utilization of the linear-chain CRF resulted in improvements in specific classes while maintaining good performance in other classes. Notably, it demonstrated significant improvement in tagging entities such as JUDGE and GPE, which often involve names of people and locations specific to the region. The CRF's ability to utilize label transition references allowed for the proper identification and tagging of these types of entities.

Taking a broader view, it can be observed that entities such as PETITIONER, RESPONDENT, and ORG exhibit poorer performance compared to other classes. Furthermore, the entity class PRECEDENT consistently achieves lower scores across all of the models. This is attributed to the fact that precedent names are typically lengthy (with an average of 14 tokens), so missing even a few tokens can result in the entire entity being labeled as incorrect.

We present the performance of an independently trained InLegalBERT model on the Pream-

ENTITY CLASS	XLM-R (v3)			XLM-R (v6)			XLM-R (v7)			InLegalBERT (v1)			InLegalBERT (v2)			InLegalBERT (v3)			COUNT
	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	
CASE_NUMBER	**0.856**	0.807	**0.910**	0.819	0.816	0.822	**0.856**	**0.835**	0.878	0.686	0.678	0.694	0.655	0.626	0.685	0.593	0.616	0.571	121
COURT	**0.875**	**0.872**	**0.879**	0.834	0.839	0.829	0.852	0.853	0.850	0.859	0.863	0.854	0.846	0.827	0.865	0.825	0.828	0.823	178
DATE	0.969	0.960	0.979	**0.987**	**0.986**	**0.987**	0.979	0.969	0.989	0.984	0.979	0.988	0.969	0.957	0.981	0.968	0.966	0.970	222
GPE	0.801	0.770	0.835	0.809	**0.799**	0.819	**0.839**	0.794	**0.889**	0.766	0.769	0.763	0.775	0.719	0.840	0.739	0.745	0.734	183
JUDGE	0.837	0.720	1.000	**0.921**	**0.853**	**1.000**	0.814	0.700	0.972	0.783	0.750	0.818	0.772	0.639	0.975	0.707	0.690	0.725	8
ORG	0.688	0.626	0.764	**0.700**	**0.680**	**0.722**	0.680	0.645	0.719	0.676	0.610	0.759	0.655	0.576	0.759	0.648	0.599	0.706	159
OTHER_PERSON	**0.911**	**0.926**	**0.895**	0.876	0.878	0.874	0.879	0.909	0.851	0.860	0.832	0.889	0.902	0.922	0.883	0.882	0.872	0.892	276
PETITIONER	0.356	0.265	0.542	0.554	0.409	0.857	0.646	0.512	0.875	0.644	0.613	0.679	**0.744**	**0.690**	**0.806**	0.658	0.625	0.694	9
PRECEDENT	**0.802**	0.781	**0.825**	0.782	0.728	0.844	0.781	**0.804**	0.760	0.748	0.712	0.787	0.682	0.658	0.708	0.614	0.610	0.618	177
PROVISION	0.919	0.896	0.943	0.909	0.884	0.936	**0.918**	**0.894**	**0.943**	0.873	0.896	0.851	0.846	0.861	0.832	0.808	0.830	0.788	258
RESPONDENT	0.444	0.320	0.727	**0.583**	0.467	**0.778**	0.500	**0.800**	0.364	0.500	0.556	0.455	0.211	0.286	0.167	0.222	0.333	0.167	5
STATUTE	0.919	0.896	0.942	**0.929**	**0.940**	**0.918**	0.902	0.874	0.932	0.887	0.870	0.904	0.902	0.888	0.916	0.891	0.882	0.900	222
WITNESS	**0.934**	**0.898**	**0.973**	0.879	0.836	0.926	0.906	0.844	0.978	0.912	0.914	0.909	0.894	0.873	0.915	0.883	0.871	0.895	58

Table 3: Table shows the F1, Precision and Recall scores for the above stated model. This includes the standard token-classification models either on XLM-RoBERTa or InLegalBERT, alongwith the CRF based XLM-R version v6, and v7. A detailed review of model configuration is provided under table.

ENTITY CLASS	F1	P	R	COUNT
COURT	0.908	0.881	0.938	118
JUDGE	0.870	0.823	0.923	166
LAWYER	0.905	0.904	0.906	589
PETITIONER	0.886	0.848	0.927	202
RESPONDENT	0.851	0.839	0.864	310

Table 4: Shows the performance of the InLegalBERT model on 'Preamble' dataset.

ENTITY CLASS	Ensemble XLM-R (max)			Ensemble XLM-R (soft)			COUNT
	F1	P	R	F1	P	R	
CASE_NUMBER	**0.866**	**0.844**	**0.890**	0.853	0.835	0.873	121
COURT	**0.890**	**0.895**	**0.886**	0.875	0.878	0.871	178
DATE	0.975	0.972	0.979	**0.976**	**0.970**	**0.982**	222
GPE	**0.833**	**0.798**	0.871	0.831	0.782	**0.887**	183
JUDGE	**0.900**	**0.818**	**1.000**	0.847	0.735	1.000	8
ORG	0.707	0.644	**0.782**	**0.708**	**0.670**	0.749	159
OTHER_PERSON	**0.923**	**0.933**	**0.912**	0.915	0.946	0.886	276
PETITIONER	0.560	0.412	0.875	**0.667**	**0.511**	**0.958**	9
PRECEDENT	**0.817**	**0.809**	**0.825**	0.805	0.800	0.810	177
PROVISION	0.938	0.921	0.956	**0.941**	**0.919**	**0.964**	258
RESPONDENT	0.538	0.467	0.636	**0.625**	**1.000**	**0.455**	5
STATUTE	**0.945**	**0.921**	**0.970**	0.938	0.914	0.965	222
WITNESS	**0.941**	**0.912**	**0.973**	0.930	0.878	0.989	58

Table 5: Shows the performance of Ensemble-based XLM-R model using either 'soft' or 'max' voting procedures.

ENTITY CLASS	XLM-R (v8)			InLegalBERT (v2) + Ensemble XLM-R ("max")			COUNT
	F1	P	R	F1	P	R	
CASE_NUMBER	0.854	0.805	**0.910**	**0.866**	**0.844**	0.890	121
COURT	0.874	0.868	0.879	**0.898**	**0.889**	**0.907**	296
DATE	**0.977**	0.970	**0.984**	0.975	**0.972**	0.979	222
GPE	0.753	0.683	0.838	**0.833**	**0.798**	**0.871**	183
JUDGE	**0.920**	**0.893**	**0.948**	0.871	0.822	0.926	174
ORG	0.647	**0.701**	0.601	**0.707**	0.644	**0.782**	159
OTHER_PERSON	0.909	0.899	0.918	**0.923**	**0.933**	**0.912**	276
PETITIONER	**0.871**	**0.891**	0.851	0.866	0.813	**0.925**	211
PRECEDENT	0.775	0.756	0.795	**0.817**	**0.809**	**0.825**	177
PROVISION	0.897	0.875	0.921	**0.938**	**0.921**	**0.956**	258
RESPONDENT	**0.870**	**0.880**	**0.860**	0.846	0.831	0.861	315
STATUTE	0.902	0.908	0.896	**0.945**	**0.921**	**0.970**	222
WITNESS	0.907	0.931	0.885	**0.941**	**0.912**	**0.973**	58
LAWYER	**0.949**	**0.947**	**0.951**	0.905	0.904	0.906	589

Table 6: Shows the results obtained on the main proposed models by us on the combined 'Judgement', and 'Preamble' datasets. (Left) shows an integration approach that combines the InLegalBERT (v2) and Ensemble XLM-R with max-voting. (Right) shows an XLM-R linear-chain CRF model trained explicitly on the Combined dataset.

ble dataset in Table 4. The model achieves an f1 score of 0.85 or higher and is capable of accurately tagging entity classes unique to the dataset. This is because the InLegalBERT model was trained on Indian court judgments and is able to differentiate between different sections of the judgment, including the prologue that introduces the case and contains metadata that is later used in the verdict. Notably, the model can differentiate between the Names of the RESPONDENT, PETITIONER, JUDGE, and LAWYER. Figure 5d illustrates that the model obtains an almost singular left-diagonal line of true positives, indicating its high degree of accuracy in predicting the majority of labels.

In Table 5, we compare two ensemble models based on XLM-RoBERTa. The first ensemble model employs the max-voting strategy, while the second one uses the soft voting approach. From the scores, we can draw an initial conclusion that both approaches perform well overall. The f1 score is above 0.8 for most of the classes, which is a reliable score. However, some classes have lower scores than others. For instance, the model achieves low f1 scores for classes like RESPONDENT, PETITIONER, and ORG. The first two classes belong to the minority group in the 'Judgement' dataset, with only 5 and 9 instances, respectively, in the dev set, which may suggest that they are more difficult to predict accurately. However, this explanation does not apply to the ORG class. By analyzing the confusion matrix for both models (refer to Figure 4d and 4c), we found that almost 30% of the ORG tags are misclassified as either GPE or COURT. This is because, in Indian judgments, the ORG tags

are always accompanied by their official location in the state, which is common to both GPE and COURT tags.

Table 6 presents a comparison of the results obtained from two of the best-performing modeling strategies for the current LegalNER task. One strategy involves merging the results obtained from a combination of InLegalBERT trained exclusively on the Preamble dataset and an Ensemble model that utilizes a max-voting strategy to combine the predictions of finetuned XLM-R with an XLM-R extended with a linear-chain CRF layer. Both models are trained independently on the Judgement data. The other strategy involves using XLM-R with a linear-chain CRF trained on the combined Judgement and Preamble dataset.

Upon analysis of Table 6, it is evident that both modeling strategies have their own unique advantages. The former strategy performs exceptionally well on classes such as PETITIONER, RESPONDENT, and JUDGE, which are common to both datasets but underrepresented in the Judgement dataset. The latter model, on the other hand, performs better on the remaining classes and achieves the highest F1 score for the PRECEDENT class among all proposed strategies. This is significant as the PRECEDENT class constitutes one of the longest entity sequences in the dataset (refer to Figure 3a).

It should be noted that a model trained specifically for entity tagging across a sequence of input tokens always tends to exhibit higher recall than precision because it attempts to tag as many tokens as possible for each inclusive entity while accurately reporting the results at each training epoch.

6.1 Error Analysis

While conducting a thorough analysis, we discovered that the task dataset contains numerous anomalies. These anomalies, combined with the lack of contextual information, resulting in poor model performance. Annotators have a tendency to assign tokens in the Judgement dataset to PETITIONER/RESPONDENT/CASE_NUMBER instead of PRECEDENT. For example, instead of classifying `the state versus Om Prakash` as PRECEDENT, the annotators label `state` as RESPONDENT and `Om Prakash` as PETITIONER.

The annotators typically use `C.J.` or `J.` as identifiers for the JUDGE tag/entity in both the Judgement and Preamble datasets. The entity usually includes the judge's name and assigned salutations (e.g., Dr.). However, the Preamble dataset is unusual in including these salutations as part of the entity, while Judgement does not. This impedes the ability to merge datasets and conduct an unbiased tagging process.

The models lack contextual support when they are trained on judgments, resulting in the disregard of some entities and mislabeling them as something else. This is particularly noticeable in cases where the entity refers to a person's name. For example, in the sentence "Arun Surendra Patil age 35 years," since there is no context indicating that "Arun" is a PETITIONER, the model misclassifies it as OTHER_PERSON.

In some cases, models can misunderstand PRECEDENT as CASE_NUMBER, such as in the example of `2006 (8) SCC 581`, which is actually a PRECEDENT but is mistakenly recognized as a CASE_NUMBER. This misidentification is evident in figure 4d, which has the highest f1 score for precedent identification but still classifies about 10% of PRECEDENT entities as either COURT or CASE_NUMBER.

We previously discussed how the labels ORG, GPE, and COURT can be misidentified as one another due to their shared locational attributes in the text. Additionally, the label ORG can also be mistakenly identified as OTHER_PERSON, such as in the case of ORG `PS Sangrur` being classified as OTHER_PERSON. It is clear that such misidentifications require external knowledge to be corrected.

7 Conclusion

We developed a Multi-model system for the SemEval 2023, task 6, Subtask B; LegalNER, which involves recognizing named entities from Indian court judgments. Our system consists of two separate models for tagging Judgement and Preamble data, as well as a transformer-based encoder model, XLM-RoBERTa, with a linear chain Conditional Random Field layer trained on combined Judgement and Preamble data. While our proposed model performs well on most of the stated entity classes spanning the two datasets, anomalies in the dataset, under-representation of certain class entities, or lack of contextual external information can cause deviations from the originally expected level of predictability. To address these issues, we

plan to adopt post-processing strategies such as coreference resolution, which can refer and align distinct entities properly, and to use an Ensemble training approach as described by Pietiläinen and Ji (2022). Ensemble models tend to highlight the finer aspects of each base model, and training them simultaneously allows the models to share encoded information via backpropagation.

In conclusion, our proposed Multi-model system for LegalNER in SemEval 2023 shows promising results, but further improvements can be made by adopting the strategies stated above.

8 Limitations and Ethical considerations

In this study, we adopted a multi-model approach that combined two distinct models trained on Judgement and Preamble, as explained previously. However, our LegalNER model only utilizes sentence-level information and may not perform well when presented with a document as input. Additionally, the constituent models are independent and cannot be substituted for one another. For instance, the model trained on Preamble cannot be used to identify the common classes for the Judgement dataset, as the representation and identification of these classes differ between both datasets. Therefore, it is not possible to overlay them together.

In the LegalNER task, an appropriate tagging system must extract entities from longer classes like PRECEDENT to shorter classes like WITNESS and associate party names with their respective entity classes. These entities provide an overview of the original court verdict. However, the task faces significant challenges in dealing with colloquialism, multilingual aspects, dialects, and writing styles used in judgments.

The task organizers derive the LegalNER dataset from an open database of Indian court judgments, which may improve entity tagging accuracy but raises ethical concerns. Sourcing judgments from an open-web database implies non-compliance with personal information (privacy), potentially exposing sensitive information such as sexual misconduct, abusive violence, defamation, and marital problems. This information may indirectly bias the model towards certain verdicts. However, the organizers' main goal is to investigate named entity identification and facilitate the decision-making process, not predict judgments or involve AI in decision-making. The organizer avoids prompting the using external information or a knowledge base to prevent discrimination against involved parties but may overlook context when instructing models to identify entities.

(Kalamkar et al., 2022) provide both an overview of the LegalNER dataset and propose an effective base-model approach to address the issue. They adopt a two-system stacking method, with the initial tagging process performed using the RoBERTa encoder model, followed by a transition-based parser to reinforce the tags and reduce type-matching errors. The authors also propose post-processing procedures such as coreferencing original PRECEDENT with their shorter form in the same judgment or pairing PROVISION with its corresponding STATUTE, which they suggest could improve results. Although they outperform our findings in categories such as RESPONDENT, PETITIONER, and ORG, we did not compare our results with theirs because they evaluated their approach on the test set, which we were unable to access.

Overall, to ensure accurate integration into the actual judicial system, it is crucial to have a well-aligned tagging procedure that eliminates any anomalies. This will improve producibility and accuracy.

References

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

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Pedro Castro, Nádia Félix, and Anderson Soares. 2019. Contextual representations and semi-supervised named entity recognition for portuguese language.

Ilias Chalkidis, Ion Androutsopoulos, and Achilleas Michos. 2017. Extracting contract elements. In *Proceedings of the 16th Edition of the International Conference on Articial Intelligence and Law*, ICAIL '17, page 19–28, New York, NY, USA. Association for Computing Machinery.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The muppets straight out of

law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online. Association for Computational Linguistics.

Ilias Chalkidis, Manos Fergadiotis, Dimitrios Tsarapatsanis, Nikolaos Aletras, Ion Androutsopoulos, and Prodromos Malakasiotis. 2021. Paragraph-level rationale extraction through regularization: A case study on European court of human rights cases. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 226–241, Online. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Alexis Conneau, Guillaume Lample, Ruty Rinott, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: evaluating cross-lingual sentence representations. *CoRR*, abs/1809.05053.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

John M. Giorgi, Xindi Wang, Nicola Sahar, Won Young Shin, Gary D. Bader, and Bo Wang. 2019. End-to-end named entity recognition and relation extraction using pre-trained language models. *CoRR*, abs/1912.13415.

Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. 2005. Bidirectional lstm networks for improved phoneme classification and recognition. In *Artificial Neural Networks: Formal Models and Their Applications – ICANN 2005*, pages 799–804, Berlin, Heidelberg. Springer Berlin Heidelberg.

M.A. Hearst, S.T. Dumais, E. Osuna, J. Platt, and B. Scholkopf. 1998. Support vector machines. *IEEE Intelligent Systems and their Applications*, 13(4):18–28.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, abs/1508.01991.

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

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, page 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *CoRR*, abs/1901.07291.

Elena Leitner, Georg Rehm, and Julián Moreno-Schneider. 2020. A dataset of german legal documents for named entity recognition.

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.

Andrei Mikheev, Marc Moens, and Claire Grover. 1999. Named entity recognition without gazetteers. In *Ninth Conference of the European Chapter of the Association for Computational Linguistics*, pages 1–8.

Hamada Nayel and H. L. Shashirekha. 2017. Improving NER for clinical texts by ensemble approach using segment representations. In *Proceedings of the 14th International Conference on Natural Language Processing (ICON-2017)*, pages 197–204, Kolkata, India. NLP Association of India.

Shounak Paul, Pawan Goyal, and Saptarshi Ghosh. 2021. Lesicin: A heterogeneous graph-based approach for automatic legal statute identification from indian legal documents. *CoRR*, abs/2112.14731.

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

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *CoRR*, abs/1802.05365.

Aapo Pietiläinen and Shaoxiong Ji. 2022. AaltoNLP at SemEval-2022 task 11: Ensembling task-adaptive pretrained transformers for multilingual complex NER. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1477–1482, Seattle, United States. Association for Computational Linguistics.

Fábio Souza, Rodrigo Frassetto Nogueira, and Roberto de Alencar Lotufo. 2019. Portuguese named entity recognition using BERT-CRF. *CoRR*, abs/1909.10649.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. *CoRR*, abs/1804.07461.

Lucia Zheng, Neel Guha, Brandon R. Anderson, Peter Henderson, and Daniel E. Ho. 2021. When does pretraining help? assessing self-supervised learning for law and the casehold dataset of 53,000+ legal holdings. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law*, ICAIL '21, page 159–168, New York, NY, USA. Association for Computing Machinery.

A Appendix

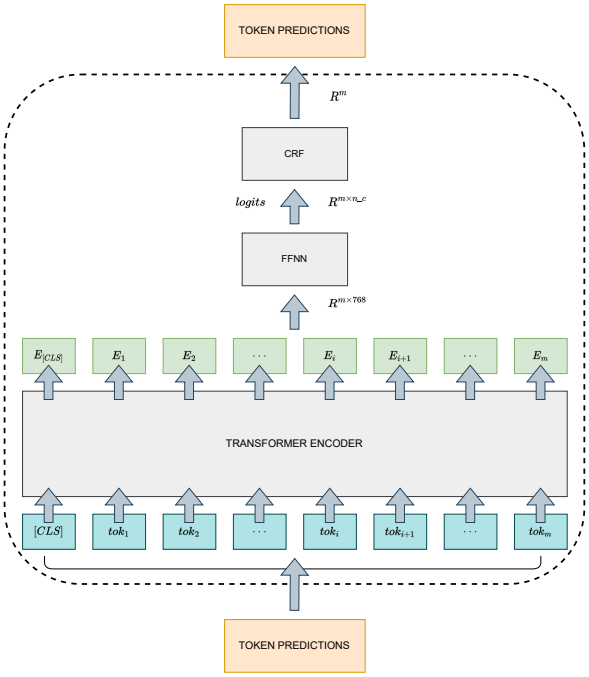


Figure 1: Shows the architecture used by us with an extension of a linear-chain CRF layer.

A.1 Personal takeaways

As I began this project, my learning objectives included drawing upon my prior experience in NLP projects to identify potential limitations in the SemEval 2022 workshop task. I aimed to familiarize my group members with the potential of SemEval and motivate them to align with the objective. Despite our original plan to compete and submit a definitive model, we ultimately had to adjust our strategy due to workload and a delayed start compared to the task start date. Thus, I helped guide my team members in identifying relevant approaches by leveraging my strong NLP background and previous language data project experience. Additionally, my proficiency in the PyTorch framework and prior coding of certain code frameworks enabled us to conduct numerous experiments.

Regarding my personal contribution to the project, I actively pursued mathematical information to guide my research and identified crucial anomalies through rigorous manual dataset study. I found it fascinating that conditional random fields could aid in entity identification, and I was pleased to see the superior performance of the XLM-RoBERTa model compared to InLegalBERT. I also studied the impact of data imbalance and

how simple weighted cross-entropy loss improves the performance of minority classes. Throughout the project, I gained several insights into different aspects of NER.

As for my main takeaways from the project, I learned that starting with multiple approaches simultaneously is crucial to ensure continued analysis if one approach fails. However, due to the absence of combining approaches initially, we experienced a deficiency in deriving a conclusive model. As a result, we had to switch to a CRF-based approach to achieve our objectives. Understanding which language model is suitable and whether we possess adequate computational resources to train it is also crucial, as we had to switch to utilizing Google colab pro to ensure the appropriate functionality of the CRF model. Finally, working in a group with each member contributing their unique strengths allowed us to merge them together and achieve better outcomes.

A.2 Transformer architecture

The transformer architecture (Vaswani et al., 2017) has revolutionized the field of natural language processing. Recent development in the field of NLP is all compliant with the derivative of transformer-based models which can avail us to perceive complex tasks efficiently. At its core, the transformer is a simple attention-based system that identifies a contextual representation of a given token, making it aware of its neighboring tokens and deriving a better semantic representation. The transformer-based encoder comprises three main components: i) self-attention, ii) multi-head attention, and iii) positional encodings.

A.2.1 Self-attention

We create a query vector, a key vector, and a value vector for a set of tokens in an input sentence. To do this, we multiply each word vector by three separate learnable matrices (W_q, W_k, and W_v). Then, we perform self-attention by multiplying the query vector q_i (representing the token at position i) with the transposed key vectors from the neighboring tokens. This produces a scalar representation for each neighboring token in relation to the current token. Next, we scale these scores down by a constant value (usually $\sqrt{dk}$, where dk represents the dimensionality of the key vector) and apply the softmax operation to obtain weights for each neighboring token and the current token. We then take the dot product of these weights with the value vector and sum them to generate the contextual representation of the current token. Figure 2 describes the entire procedure visually.

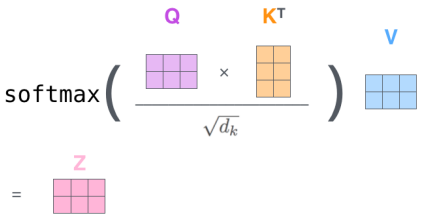


Figure 2: Vectorized visual representation of self-attention used by transformer architecture.

A.2.2 Multi-head attention

In a normal transformer architecture, the model uses multi-head attention to obtain contextual embeddings for each token in an input sentence. The model distributes the input embeddings into multiple heads, typically eight, and performs a self-attention procedure with distinct Query W_{qi}, Key W_{ki}, and Value W_{v_i} weights that it learns independently. The model then combines the contextual representation obtained from each head and applies a final linear transformation to produce the output representation.

A.3 Positional encodings

The transformer does not process input text sequentially like RNNs, making it unaware of sentence structure. Therefore, the model needs to generate an encoding that informs the transformer of the position of each word in the sentence. To achieve this, positional encodings are derived. Positional encoding is a technique utilized in transformers to include the position of tokens in the input sequence. The technique involves adding a fixed sinusoidal function with a unique frequency for each position, allowing the model to differentiate between tokens based on their sequence position.

A.4 linear-chain CRF

The linear-chain CRF is a more computationally efficient version of the normal conditional random field model. It focuses only on adjacent labels instead of all combinations of sequence labels, making it more efficient. Essentially, it is a sequence classification problem that aims to find a probabilistic model for a sequence of labels y given an input sequence X transformed into a feature representation. Unlike standard classification problems that

use the softmax operation to calculate label probabilities, linear-chain CRF uses additional learnable weights to model the probability of a label y_k followed by y_{k+1}. This captures the dependency between successive labels. Equation 4 shows that the model comprises an emission score U, which represents the probability of y_k given input x_k, a transition score T, which represents the probability of y_k followed by y_{k+1}, and a partition function Z that normalizes the probability over all possible sequences.

$$P(\mathbf{y} \mid \mathbf{X}) = \frac{\exp\left(\sum_{k=1}^{\ell} U\left(\mathbf{x}_k, y_k\right) + \sum_{k=1}^{\ell-1} T\left(y_k, y_{k+1}\right)\right)}{Z(\mathbf{X})} \quad (4)$$

$$Z(\mathbf{X}) = \sum_{y_1'} \cdots \sum_{y_\ell'} \exp\left(\sum_{k=1}^{\ell} U\left(\mathbf{x}_k, y_k'\right) + \sum_{k=1}^{\ell-1} T\left(y_k', y_{k+1}'\right)\right) \quad (5)$$

A.4.1 Loss Function

In CRF, just like any other supervised classification problem, the main aim is to minimize expected loss during training. To achieve this, we use a loss function L, which calculates a score by matching ground-truth labels with the hypothesis derived by transforming input features. We calculate the loss function L by applying negative log-likelihood to the original score function, as described in Equation 7. To calculate the emission scores $U(x_k, y_k)$, which are obtained by passing input features through an encoder model (logits), we use a forward pass. We then compute the transition score from the current label y_k to the following label y_k+1. Furthermore, since the partition function depends exponentially on the total label $|Y|$, we use dynamic programming and recursion to optimize the functional dependencies accordingly. We simplify this process in the actual implementation, as described in Equation 8.

$$L = -\log(P(\mathbf{y} \mid \mathbf{X})) \quad (6)$$

$$L = Z_{\log}(\mathbf{X}) - \left(\sum_{k=1}^{\ell} U\left(\mathbf{x}_k, y_k\right) + \sum_{k=1}^{\ell-1} T\left(y_k, y_{k+1}\right)\right) \quad (7)$$

$$\log \sum_k \exp\left(z_k\right) = \max(\mathbf{z}) + \log \sum_k \exp\left(z_k - \max(\mathbf{z})\right) \quad (8)$$

A.4.2 Vertibi Algorithm

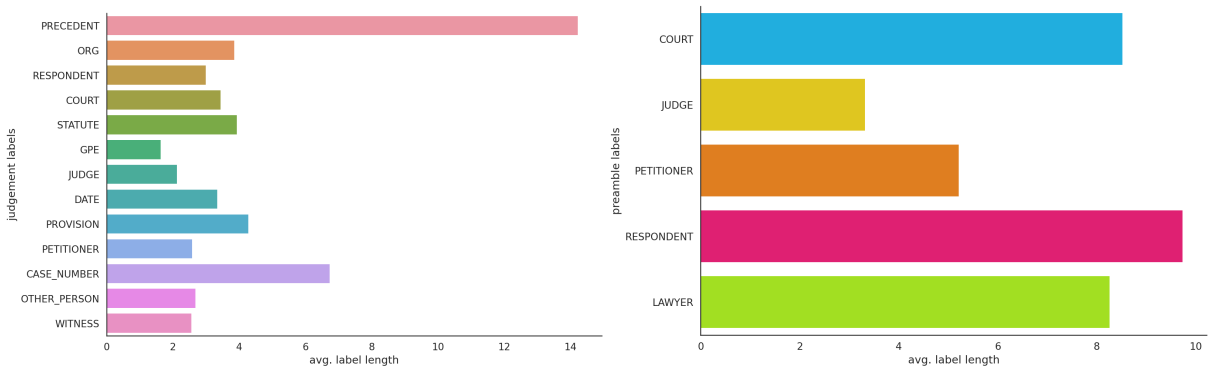
During the testing phase, CRF presents a challenge where we have to utilize both the forward and backward algorithm, also known as the forward and backward pass, simultaneously to determine the label that maximizes the probability of $p(y_k|X)$ at every timestamp k. However, it is not necessary to calculate the backward algorithm to find the most likely sequence of labels. Instead, we can simply keep track of the maximum scores at each timestep during the forward algorithm. Once the forward algorithm is completed, we can follow the backward trace of the maximum operations (argmax) to decode the sequence that maximizes the scores.

Entity	**Judgment Count**	**Preamble Count**
COURT	1293	1074
PETITIONER	464	2604
RESPONDENT	324	3538
JUDGE	567	1758
LAWYER	NA	3505
DATE	1885	NA
ORG	1441	NA
GPE	1398	NA
STATUTE	1804	NA
PROVISION	2384	NA
PRECEDENT	1351	NA
CASE NUMBER	1040	NA
WITNESS	881	NA
OTHER PERSON	2653	NA
Total	17485	12479

Table 7: Shows the counts of legal entities for training data in "Judgment", and "Preamble"

Hyperparameter	Value
Epochs	20
Batch Size	2
Max Sequence Length	256
Optimizer	AdamW
Output Hidden Size	768
Dropout Rate	0.2
Learning Rate	1e-6
Weight Decay	1e-4
Early Stopping	5

Table 8: Shows the general hyperparameter setting used to train the model with varied configurations. Note, we tend to change certain values based on computing set-up or experimentation.



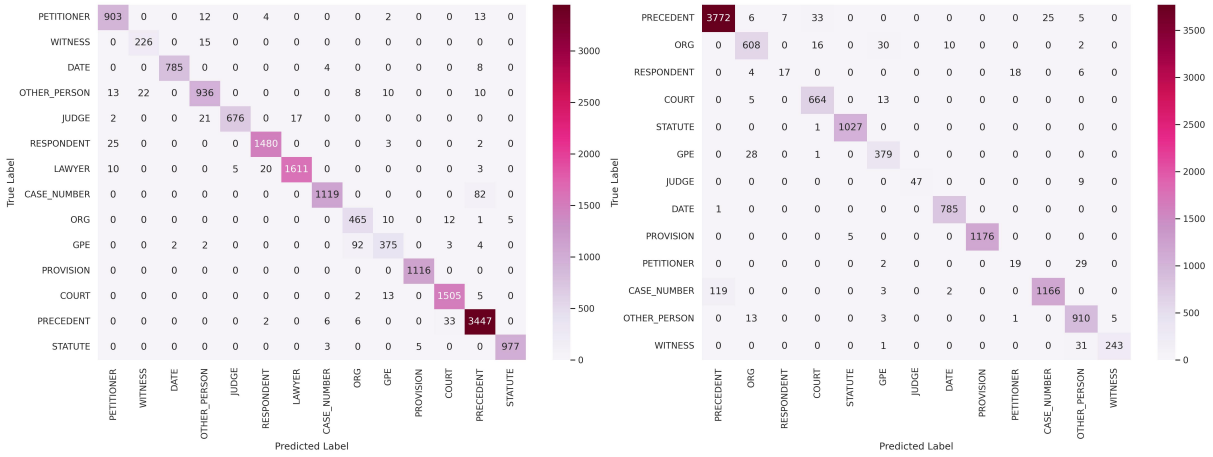
(a) shows the average length of each tag in the judgment dataset.(b) shows the average length of each tag in the preamble dataset.

Version Name	**Encoder Type**	**Extract Form**	**Tagging Type**	**Training Method**
v1	InLegalBERT	Judgement	BIO	normal CrossEntropy
v2	InLegalBERT	Judgement	BIOES	weighed CrossEntropy
v3	InLegalBERT	Judgement	BIOES	normal CrossEntropy
v4	InLegalBERT	Preamble	BIOES	normal CrossEntropy
v1	XLM-RoBERTa	Judgement	BIO	normal CrossEntropy
v2	XLM-RoBERTa	Judgement	BIOES	weighted CrossEntropy
v3	XLM-RoBERTa	Judgement	BIOES	normal CrossEntropy
v6	XLM-RoBERTa	Judgement	BIO	linear-chain Conditional Random Field (CRF)
v7	XLM-RoBERTa	Judgement	BIOES	linear-chain Conditional Random Field (CRF)
v8	XLM-RoBERTa	Judgement + Preamble	BIOES	linear-chain Conditional Random Field (CRF)

Table 9: Shows the different training configurations proposed under the submission of team VASPOM for task LegalNER at SemEval 2023.

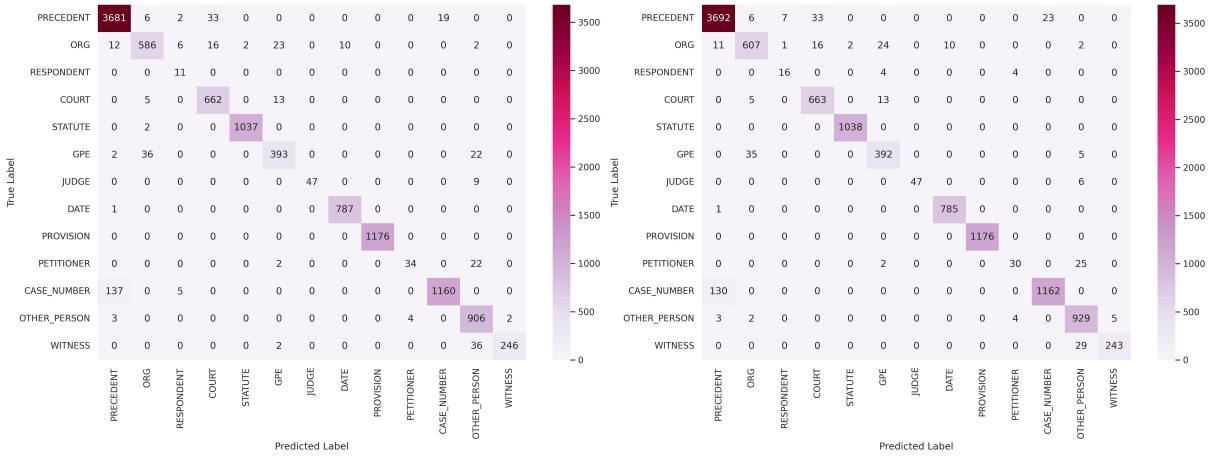
Voting Strategy	**Model 1**				**Model 2**			
	Encoder Type	**Extract Form**	**Tagging Type**	**Training Method**	**Encoder Type**	**Extract Form**	**Tagging Type**	**Training Method**
soft	XLM-R	Judgement	BIOES	normal CrossEntropy loss	XLM-R	Judgement	BIOES	linear chain Conditional Random Field (CRF)
max	XLM-R	Judgement	BIOES	normal CrossEntropy loss	XLM-R	Judgement	BIOES	linear chain Conditional Random Field (CRF)

Table 10: Shows the different ensembling approaches proposed under the submission of team VASPOM for the task of LegalNER at SemEval 2023



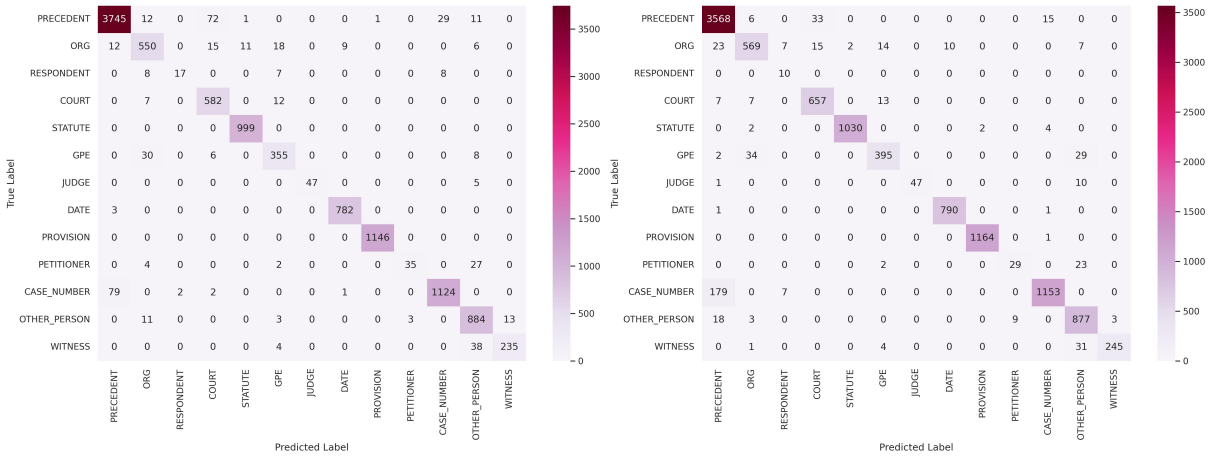
(a) XLM-R (v8)

(b) XLM-R (v3)



(c) Ensemble XLM-R (soft)

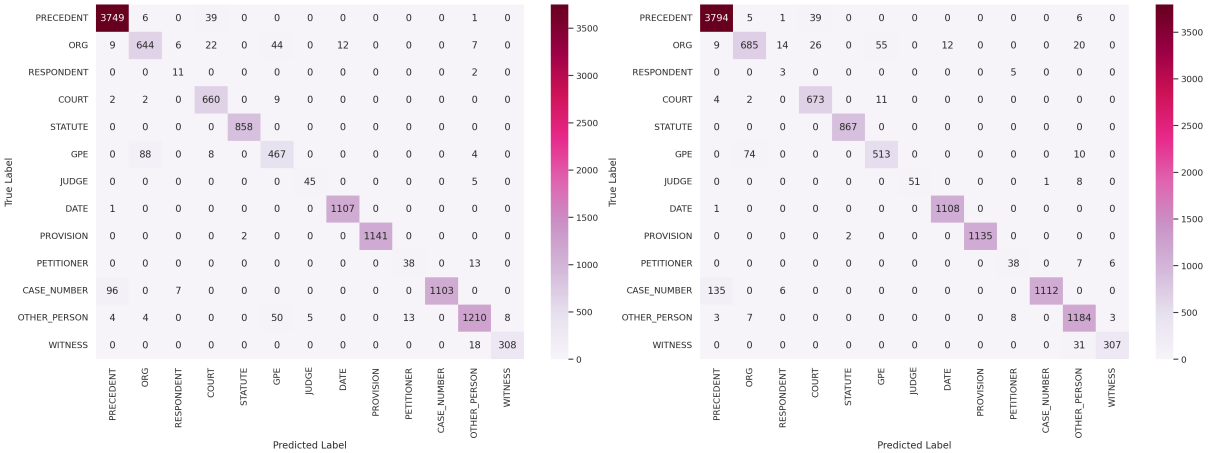
(d) Ensemble XLM-R (max)



(e) XLM-R (v6)

(f) XLM-R (v7)

Figure 4: Shows the confusion matrix for each of the token classification models trained using the XLM-RoBERTa encoder. This includes all of the normal training configurations with and without linear-chain CRF. Additionally, it also shows the confusion matrix for the Ensemble XLM-R model. For more information related to model configuration see table 9, 10.



(a) InLegalBERT (v1)

(b) InLegalBERT (v2)

(c) InLegalBERT (v3)

(d) InLegalBERT (v4)

Figure 5: Shows the confusion matrix derived for each of the token classification models trained using an InLegal-BERT encoder. For more information related to model configuration see table 9.