Indian Legal Case Judgment Document Mining: Semantic Segmentation and Information Extraction

by
Antara Das

Under the supervision of

Dr. Vikram Goyal

Submitted in partial fulfilment of the requirements for the degree of Master of Technology, CSE

Department of Computer Science Engineering
Indraprastha Institute of Information Technology - Delhi
May, 2024

Certificate

This is to certify that the thesis titled *"Indian Legal Case Judgment Document Mining: Semantic Segmentation and Information Extraction"* being submitted by **Antara Das** to the Indraprastha Institute of Information Technology Delhi, for the award of the Master of Technology, is an original research work carried out by him under my supervision. In my opinion, the thesis has reached the standards fulfilling the requirements of the regulations relating to the degree.

The results contained in this thesis have not been submitted in part or full to any other university or institute for the award of any degree/diploma.

May, 2024

Dr. Vikram Goyal

Department of Computer Science Engineering
Indraprastha Institute of Information Technology Delhi New
Delhi 110 020

Acknowledgements

I express my deep gratitude to my advisor, Dr. Vikram Goyal, for providing me with the remarkable opportunity to work on an industrial research project in collaboration with Kronicle Research, Dr. Pawan Goyal and Dr. Saptarshi Ghosh. Across my thesis and academic pursuit, my advisor's expert guidance, unwavering support, and encouragement have been invaluable.

I extend my gratitude to Dr. Pawan Goyal and Dr. Saptarshi Ghosh for providing me with their knowledgeable insights and guidance for this work. I am grateful to Mr. Joseph Pookkatt, director of Kronicle Research, and his team for providing novel datasets for this work and helping me with their expert legal domain knowledge. I shall also thank my PhD mentor, Shounak Paul, for his critical support and guidance throughout the journey and for enhancing my knowledge.

Furthermore, I am thankful to my family and friends for their constant moral and emotional support.

Abstract

Developing NLP-based techniques to automate tasks in the Indian legal domain is highly demanding due to the enormously increasing volume of legal text documents, intricate legal terminologies, and the need for efficient information retrieval and document analysis for legal professionals. These techniques streamline extensive processing, extraction, and understanding of legal information, aiding more productivity within the judicial framework. In this work, we have experimented with two tasks: Task 1 deals with extracting eight legal domain-specific named entities from the Indian court judgment texts, and Task 2 is on semantic segmentation of Indian case judgment documents into different functional or rhetorical components such as Facts, Arguments, Judgment statement etc. We have introduced two new large corpora for each task, which enabled us to experiment with different transformer-based models. For Task 1, we propose a hybrid approach combining a BERT-CRF model for token classification and uniquely designed rule-based information extraction. The semantic segmentation task can be modelled in two ways: a high-level approach automatically segregates a given text document into multiple functional chunks using a subtask called Label Shift Prediction, and another detailed approach classifies the rhetorical roles of those text chunks. We have extensively experimented on Task 2 to improve prior research by introducing different ways to incorporate the Label Shift Prediction task to enhance the hierarchical BERT-based approach of the rhetorical role identification task. Also, in Task 2, we worked on a dataset with more fine-grained RR labels and huge label imbalances and significantly improved the performance of rare labels using a dynamically weighted loss. Further we have experimented with cross domain performance of RR and LSP prediction models and shown that finetuning a model with a small corpus of a target domain is can efficiently provide solution for cases from unseen domain.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Indian courts are bogged down with a massive number of cases. Most of the case documents are highly unstructured, noisy, spelling error-prone, and lengthy texts that use legal jargon terms. Manually scrutinizing them is arduous, but every task in the judicial framework cannot be fully automated as all the cases are different and need to be treated uniquely with domain expertise and knowledge. So, for a solution, we can look towards Natural Language Processing, which can convert the texts and efficiently organize the data, which will be helpful in extracting the most important aspects of a lengthy judgment document. These NLP and AI-powered approaches can streamline extensive processing, extraction, and understanding of legal information, aiding more productivity within the judicial framework. The documents can be structured by organizing the paragraphs or different sections for particular aspects of the text. Also, we might unearth some valuable terms or entities from the text that can be kept aside for quick reference in the future. Earlier works for legal automation tasks include legal NER, court judgment prediction, rhetorical roles identification, similar legal document retrieval, legal statutes identification and so on. This project tries to contribute to two downstream tasks in the legal domain: Legal Named Entity Extraction and Semantic segmentation of legal case judgment documents. In the following subsections, we try to define our tasks and motivation for them.

1.2 Problem Definition: Legal NER

The legal NER task performed in this work is associated with extracting eight legal domain-specific named entities from the Indian court judgment texts (as shown in Fig. 1.1). When we need to unearth a particular entity from a text document, we can use Named Entity Recognition (NER). Although there have been some off-the-shelf models for the NER task, legal entities are much more diverse and different than the ones used in existing models. Thus, it is necessary to have a legal domain-specific NER model.

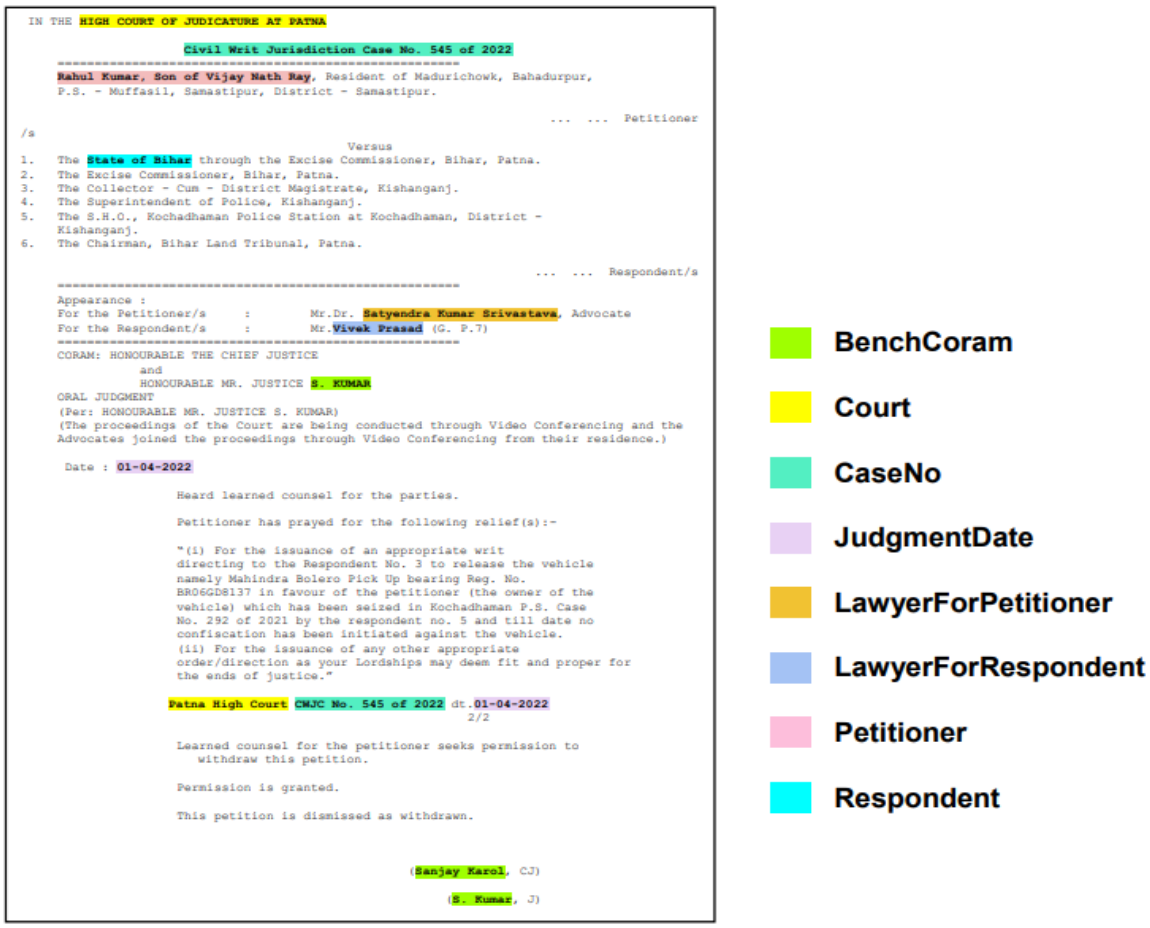

Fig. 1.1: Legal NER Task

1.3 Problem Definition: Semantic Segmentation

The semantic segmentation task can be done effectively using Rhetorical Roles (RR) prediction (task example shown in Fig. 1.3). Rhetorical roles illustrate the function of a particular sentence in the Legal Document. Some of these rhetorical roles include "facts of the case", "arguments by the petitioner", "final judgment", and so on. However, a subtask called Label Shift prediction can also segment a given text document into multiple semantically meaningful contiguous chunks of text (use case example shown in Fig. 1.2). This label shift task can be defined as a binary classification task to predict whether the rhetorical role of a sentence will shift from its previous sentence S_{i-1}, given any input sentence S_i (1 represents 'Shift' and 0 represents 'No Shift').

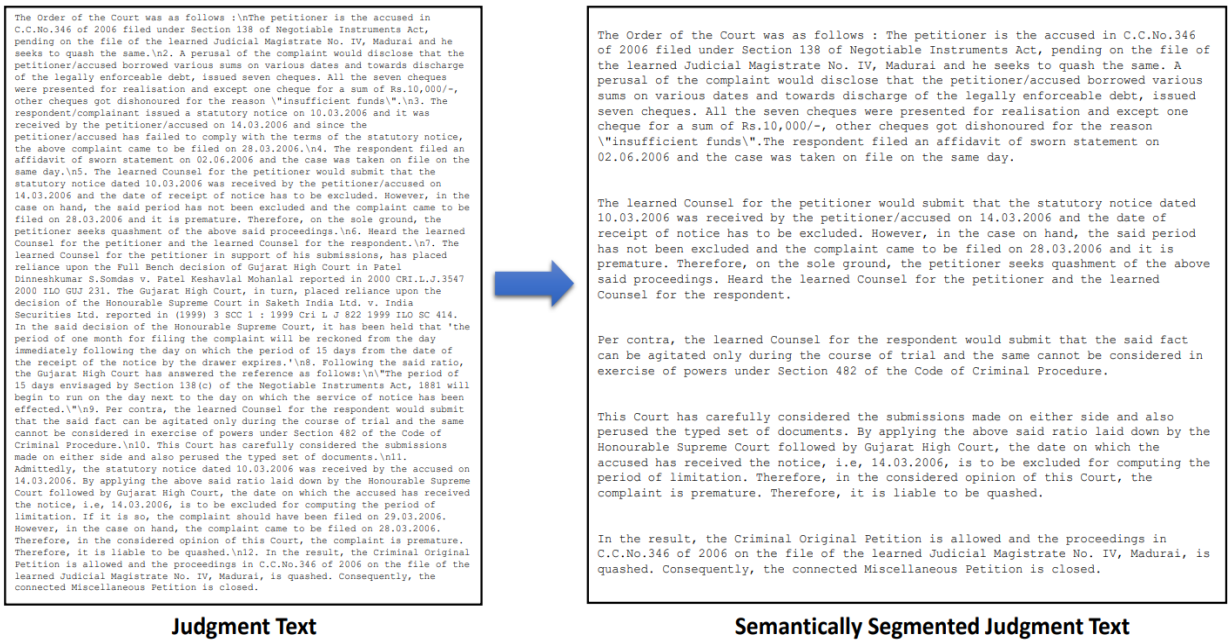

Fig. 1.2: Label Shift Prediction Task

Automated identification of rhetorical roles can be achieved by a multi-class classification task to predict the rhetorical role of each sentence given in an input case judgment document. If a legal professional can understand the sections according to their purpose in a lengthy case judgment text or learn some essential metadata about the case, it will be easier to do selective reading according to their requirements. Traditional ML methods with handcrafted features have worked at times, but now the focus has been shifted towards sophisticated Deep Learning architectures, which help us to gather features more efficiently.

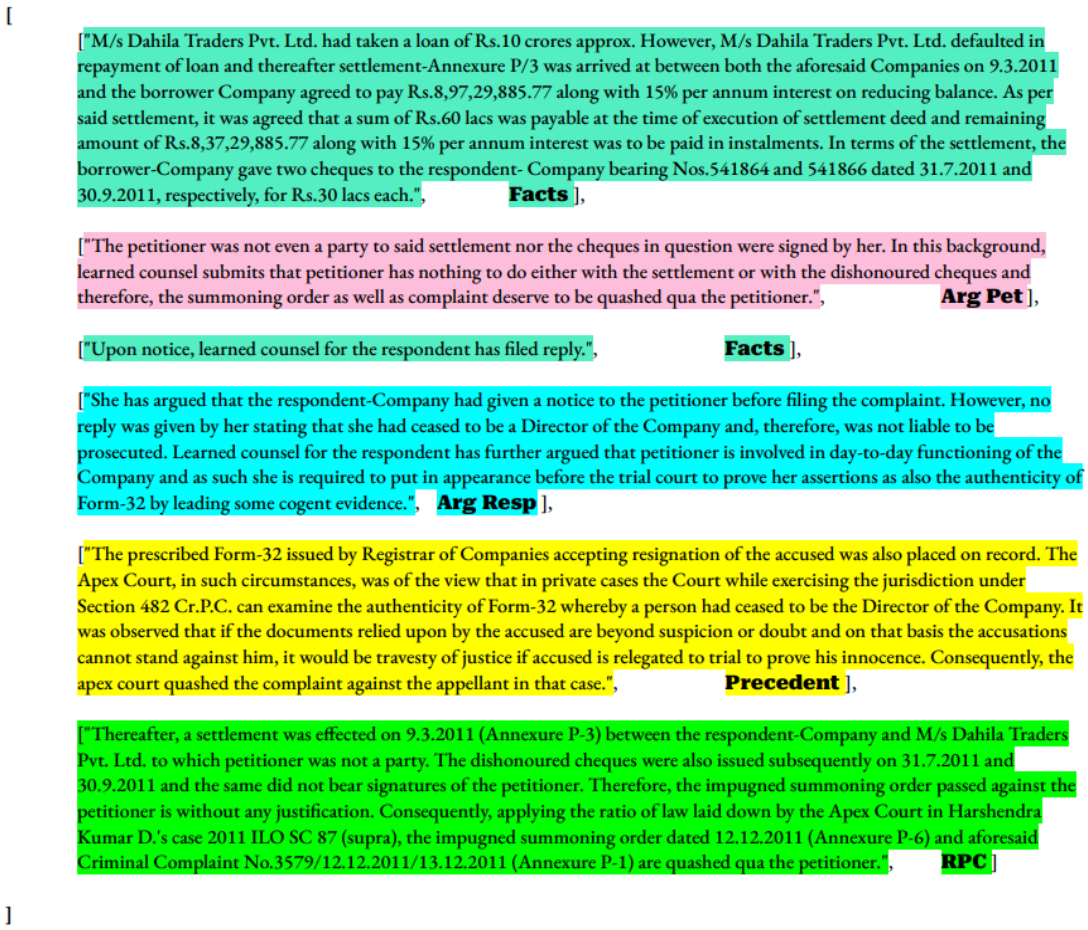

Fig. 1.3: Semantic Segmentation of legal document using rhetorical roles

Chapter 2

Related Work

2.1 NER:

In the earlier era of classical NLP, Zhou et al., 2002 [11] introduced the NER task using a chunk tagger based on the Hidden Markov Model. Conditional Random Fields (Laferty et al., 2001 [26]) also gained popularity for the NER task. Huang et al., 2016 [17] proposed a BiLSTM-CRF model for chunk tagging and experimented on CoNLL2000 and CoNLL2003 datasets for NER and POS tagging tasks. Research in the legal NER domain is rarely performed, though many works have been done in European languages like German, Romanian, Hungarian, etc. Wang et al., 2020 [14] proposed a global attention-enabled BiLSTM-CRF model for chunk tagging to categorize Chinese judicial texts. Kalamkar et al., 2022 [16] proposed a large corpus of 46545 annotated legal named entities mapped to 14 legal entity types. The named entities present in this corpus consist of legal domain-specific tags like "PETITIONER", "COURT", "LAWYER", etc., as well as flat entities like "ORG", "GPE", "OTHER_PERSON", which are also applicable for general texts. Our work on legal NER contributes primarily to extracting eight predefined legal named entities using a hybrid approach of deep learning and rule-based code.

2.2 Pretraining Models:

Legal data mining has been an active research area for over a decade. This research area has gained more popularity in the past few years after different variations of pretrained BERT models entered the desk and adequate gold datasets became available for different downstream tasks. Numerous legal datasets covering various regions, such as the US, EU, China, and India, have been released in recent years. This has prompted the application of pre-trained transformer models like BERT (Devlin et al., 2019 [23]) on these datasets. Given the distinctive nature of legal texts and the use of legal jargon terms in the texts, there is considerable potential for enhancing models like BERT to make them more useful for different legal-AI tasks, which are pre-trained on general-domain data. Several initiatives have focused on pre-training transformers specifically for the legal domain:

1. Chalkidis et al. [20] developed LegalBERT by pre-training BERT-base on legislative and court documents from the EU, UK, US, European Court of Justice (ECJ), and European Court of Human Rights (ECtHR).
2. Zheng et al. [12] introduced CaseLawBERT, pre-trained on a corpus comprising US case law documents and contracts.
3. Henderson et al. [18] created an extensive corpus called Pile of Law, consisting of documents from the US, Canada, and the EU (not limited to case law) and trained BERT-large on this dataset, resulting in the PoLBERT model.
4. Xiao et al. [13] released Lawformer, a model based on Longformer [2], pre-trained on Chinese legal texts.
5. Paul et al. [4] introduced InLegalBERT and InCaseLawBERT by retraining the earlier released models LegalBERT and CaseLawBERT, respectively, on approx 5.4 million case documents from the Indian Supreme Court and High Courts.

2.3 Automatic Rhetorical Roles Identification:

The task of automated identification of rhetorical roles has evolved from classical ML approaches using handcrafted features to deep learning methods with hierarchical attention-based networks or multi-task learning frameworks. Many experiments for understanding the rhetorical roles in court case documents were performed as a subtask for summarizing those documents (Farzindar et al., 2004 [24]; Hachey et al., 2006 [25]; Saravanan et al., 2010 [3]). Saravanan et al., 2010 [3] used Conditional Random Fields (Laferty et al., 2001) [26] for the same task, considering seven rhetorical roles. They experimented on cases from three domains – rent control, income tax and sales tax. Labelling sentences as facts or principles using handcrafted features and the Multinomial Bayes Classifier was explored by Shulayeva et al., 2017 [27]. Segmenting a document into a few high-level functional parts was addressed by Savelka et al., 2018 [28] on U.S. court documents using CRF with handcrafted features. Next, Bhattacharya et al., 2019 [10] proposed a BiLSTM-CRF model with sent2vec embeddings, and later in their following work, Bhattacharya et al., 2021 [9] experimented with a HierBiLSTM-CRF architecture with different types of embeddings like Law2vec, LegalBERT, sent2vec etc. They used two datasets from the Indian Supreme Court and the UK court, with 50 cases from five different domains of law in both datasets, each with seven rhetorical role classes. The UKSS dataset was used here to show the generalizability of the trained model in the cross-jurisdiction dataset. Kalamkar et al., 2022 [7] created a large corpus of RRs and proposed transformer-based [1] baseline models for RR prediction and showed the SciBERT-HSLN model as their proposed model. Paul et al. [4] show the utility of their pretrained BERT model for the downstream task of rhetorical role identification using a HierBERT architecture, and they performed experimentation for this task on the ISS and UKSS dataset proposed by Bhattacharya et al. 2019 [10]. The latest contribution to semantic segmentation was made by Mailk et al., 2022 [5] with the introduction of another new dataset from the IT and CL domain of law, having 40 cases each from both domains. They proposed a novel multi-task learning approach to RR prediction using an auxiliary task called Label Shift prediction. The proposed dataset in this work is annotated with 13 different fine-grained rhetorical roles. However, in their experiments, they worked on only 7 roles by merging related labels to avoid lower performance issues of rare labels. We extend the last two approaches in our proposed method and experiment with a more fine-grained set of rhetorical roles than previous works.

Chapter 3

Dataset

A standard Indian court judgment document consists of two separable sections: "preamble" and "judgment text". The preamble encompasses structured metadata such as the names of parties, judges, lawyers, judgment date, and court details. This section generally does not use complete sentences to represent the pieces of information. Following the "preamble" section, a line typically contains a keyword like "ORDER" or "JUDGMENT", which indicates the beginning of the next section, and the rest of the text following this is considered the "judgment text" section. Give a case judgment document file, these sections can automatically divided using a rule-based approach. The named entities can be present in both mentioned sections, whereas the rhetorical roles can be labelled only to sentences of the "judgment text" section.

We propose new datasets for NER and semantic segmentation tasks in this work. All of the data annotations are done by legal professionals of the Kronicle Research organisation using the tool “Label Studio”. We have considered 1000 and 500 legal case judgment documents from different State High Courts and the Supreme Court of India as input text data for Task 1 and Task 2, respectively. One of our objectives was to ensure that the model could understand the document structure and text patterns that vary across different courts and perform well despite all the variations. All of the experiments reported in this work are performed on these datasets. The individual dataset insights and statistics are given below.

3.1 Legal NER

For the NER task, we were provided with 1000 annotated case documents, which consisted of 50 cases from each of the 19 different State High Courts and the Supreme Court of India, with the judgment date ranging from 2021 to 2022. The annotators considered only eight fine-grained legal domain-specific named entity tags here, not generic flat entities like ORG, GPE, DATE, etc., used in other NER tasks. The list of the named entities and their descriptions are given in Table 3.1.

Named Entity	Description
Court	Name of the court delivering the current judgment
CaseNo	Case number(s) of the current case for which judgment is performed
Petitioner	Name(s) of the petitioners or appellants of the current case
Respondent	Name(s) of the respondents or oppositions of the current case
JudgmentDate	The date on which the judgment of the current case is announced
BenchCoram	Name(s) of the judges present in the judgment bench of the current case
LawyerForPetitioner	Name(s) of the lawyer representing the petitioners
LawyerForRespondent	Name(s) of the lawyer representing the respondents

Table 3.1: Legal Named Entity descriptions

The tags described above**,** except '**JudgmentDate' and 'Court'**can have more than one distinct entity in the document. It's expected to have at least one entity per document for each tag. Most of these named entities are primarily present in the preamble section of the document, whereas few others can be present in both the preamble and judgment-text. A total of 10503 entities are present in the corpus. The distribution of the tags is presented in Fig. 3.1 below.

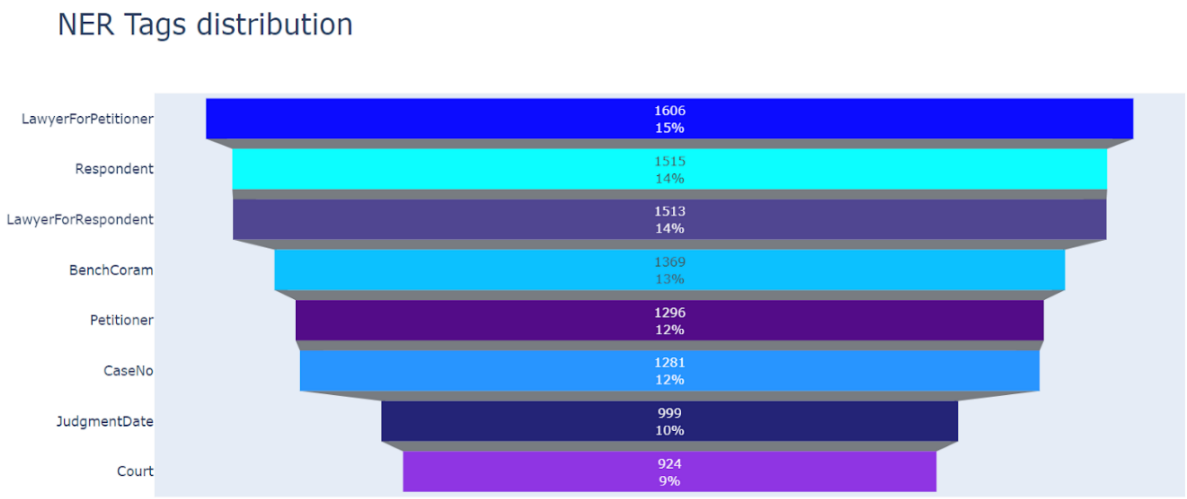

Fig. 3.1: NER Tags distribution in the entire dataset of 1000 legal cases

The entities from 800 cases are used to prepare the training dataset, and the entities from the rest of the 200 cases are used as dedicated test datasets. This train-test splitting is done strategically to ensure that both splits contain enough variations of the cases. Further, validation data is randomly generated, taking 10% of the training data during the training.

3.2 Semantic Segmentation

We were given 500 judgment cases for the semantic segmentation task with sentence-level annotation of rhetorical roles. The text-based statistics of the data are given in Table 3.2 to get an insight into the overall length and density of the documents. This dataset has varying cases from 17 different State High Courts and the Supreme Court of India, with a wide range of judgment dates from 2001 to 2022, as shown in dataset statistics of Table 3.3.

Property	Minimum	Average	Maximum
Tokens per sentence	2	31	507
Sentences per document	3	17	1018
Tokens per document	40	521	47607

Table 3.2: Text Statistics of RR annotated corpus

Court	Case Count	Judgement Years
ALLAHABAD HIGH COURT	23	2008 to 2020
ANDHRA PRADESH HIGH COURT	10	2009 to 2019
CALCUTTA HIGH COURT	16	2020
CHHATTISGARH HIGH COURT	1	2019
DELHI HIGH COURT	28	2008 to 2020
GUJARAT HIGH COURT	12	2020 to 2021
GAUHATI HIGH COURT	10	2017 to 2020
HIMACHAL PRADESH HIGH COURT	2	2019
KARNATAKA HIGH COURT	33	2018 to 2021
KERALA HIGH COURT	177	2007 to 2020
MADRAS HIGH COURT	31	2009 to 2020
MADHYA PRADESH HIGH COURT	11	2001 to 2019
BOMBAY HIGH COURT	49	2003 to 2021
PATNA HIGH COURT	1	2011
PUNJAB AND HARYANA HIGH COURT	32	2008 to 2020
RAJASTHAN HIGH COURT	4	2008 to 2012
SUPREME COURT OF INDIA	52	2001 to 2021
UTTARAKHAND HIGH COURT	8	2009 to 2011

Table 3.3: Court and Year wise distribution of cases in the RR annotated corpus

According to the legal experts we consulted, 14 fine-grained rhetorical roles are possible over the legal case judgment corpus. However, due to the extremely rare availability of a few roles in our corpus (typically less than 1%), we have merged a few related labels and

considered 10 labels for all of the reported experiments in this work. The gold binary labels for the LSP task are derived using this rhetorical role annotation based on whether two adjacent sentences have the same rhetorical role or not. The list of the rhetorical roles we considered and their descriptions are given as follows:

Facts: It describes the history of the case including the events that led to the case filing.
Issues: It is the point for determination by the court. Courts frame the issues based on the points disputed by the parties in the case.
RLC (Ruling by Lower Court): It is the final decision or ruling of the lower courts dealing with the same case. It also includes an analysis of the case or observations made by the lower court.
Arg Pet (Argument for Petitioner): Arguments which have been put forward by the petitioner/appellant in the case. All the cases argued by the petitioner's lawyer are included under this.
Arg Resp (Argument for Respondent): Arguments that have been put forward by the respondent in the case. All the cases argued by the respondent's lawyer are included under this.
Authority: Reference made by the court to authoritative books, lexicons, legal principles and maxims, international conventions, or studies.
Statute Applied: The Court often cites statutory provisions or law while discussing how the prevalent law applies to the facts and circumstances of a given case.
Precedent: A judgment or decision of a court that is cited in a subsequent dispute as an analogy to justify deciding a similar case or point of law in the same manner. It is relied upon by the Court to develop its reasoning. This label can be divided into 4 finer labels: 'Precedent Relied', 'Precedent not Relied', 'Precedent Distinguished', and 'Precedent Overruled'. The last three variations are found to occur in extremely rare cases.
Analysis: Court discussion and observation regarding the applicability of the law, precedents, and statute in the present case. We have merged the 'Ratio of the decision' label with this.
RPC (Ruling by Present Court): It is the final decision, conclusion and order of the court.

The dataset consists of a total of 24500 sentences across all documents. The overall distribution of the rhetorical roles is presented in Fig 3.2. The dataset is strategically divided into three splits, with 80% training, 10% validation, and 10% test data. It is made sure that all of these splits contain at least one example for every label.

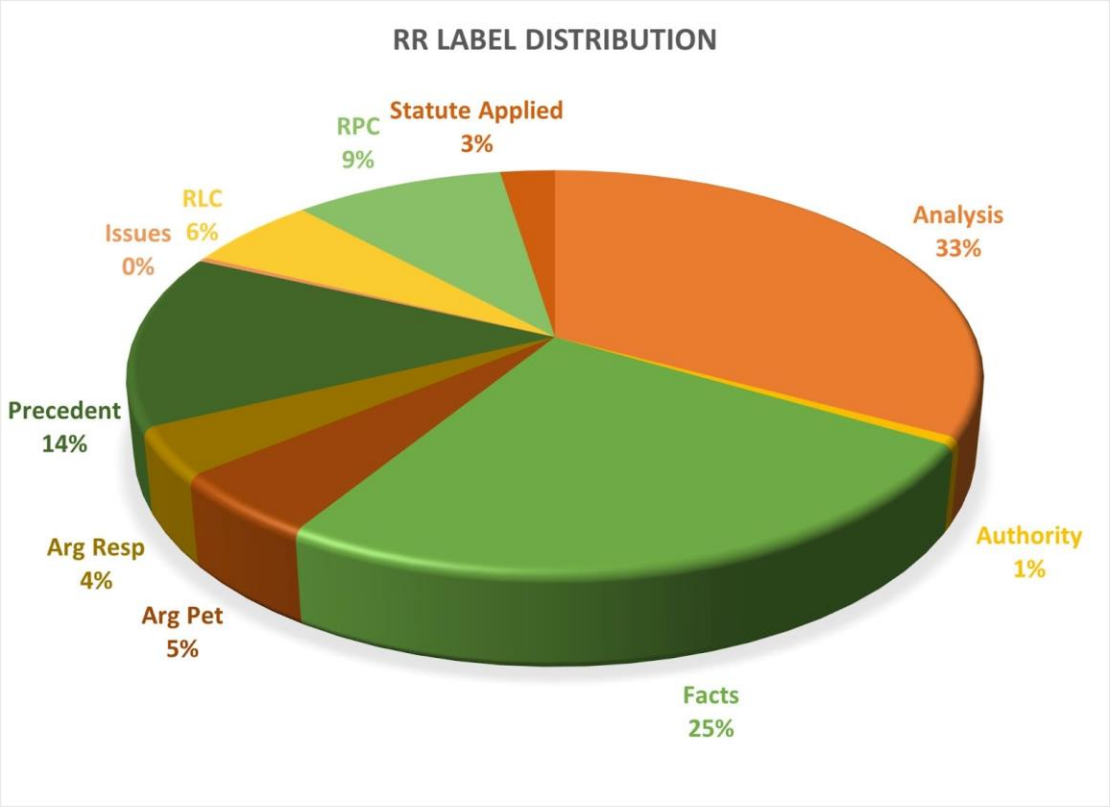

Fig. 3.2: Distribution of Rhetorical roles in the entire dataset

Chapter 4

Proposed Methodology

4.1 Baselines: Legal NER

In the first baseline, we modelled the task as an information extraction task using an open-source pretrained SpaCy model called **'en_legal_ner_trf'**. This model was trained and released by the OpenNyai organisation for legal-NER task [16]. We performed only inference using this model to extract the desired eight tag values. To avoid issues of exceeding the maximum length allowed for input by this model, we chunked the entire document into multiple sections and then passed them as input to extract the NER tag values. The model can predict values for 14 different tags, as discussed in section 2.1, and a few of those are exactly the same as the tags defined by us. For those tags like '**Petitioner', 'Respondent',** and **'BenchCoram',** all of the values predicted by the model can be used directly. The model predicts a list of generic values for the tags '**Court**', '**CaseNo**' and '**JudgmentDate**', whereas we need the current case-specific values. To serve our purpose, we only consider the first value predicted for these three tags by the model, as it's highly probable that those will come from the preamble, which will consist of our desired values. For the tags '**LawyerForPetitioner**' and '**LawyerForRespondent**', the model can't predict these values explicitly, but it can predict the values for 'Lawyer', which is basically a combination of these two tags. This is how we utilise this model to get our first baseline.

For the second baseline, we implement an existing approach of BiLSTM-CRF (described in section 2.1). For getting the token embeddings, Word2vec is used. This approach models the task as a token classification problem. Here, a sequence of 'n' tokens (present in the entire case judgment document) represents a training example, and the model output is the list of 'n' BIO_tags for each input token. Then, we perform BIO-to-Span mapping to extract the final named entities.

4.2 Proposed Model: Legal NER

The proposed hybrid model (shown in Fig. 4.1) uses a PLM-based architecture for token classification and rule-based information extraction. The model architecture and workflow are described here.

We used a legal domain-specific Pretrained Language Model (PLM) called 'Indian-LegalBERT' for the token classification model, followed by a fully connected and Conditional Random Field (CRF) layer. The details of the 'Indian-LegalBERT' model is discussed in section 2.

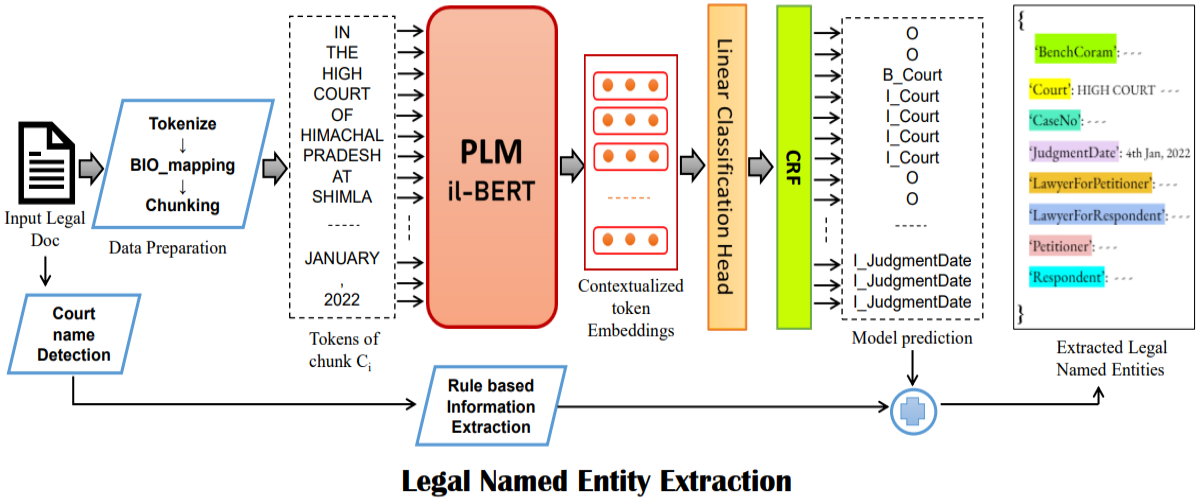

Fig. 4.1: Legal Named Entity Extraction

Here, the training examples or batches consist of a contiguous chunk of text from a case judgment text document. Legal documents can be longer than the sequence limit of 512 tokens that BERT-base models can accommodate. That's why we divide each document into a group of contiguous chunks where each chuck has up to 510 tokens, and the ending token should be punctuation to avoid truncation in the middle of a sentence. Given a case judgment document and the index spans of the named entities present in the document, as the first step, tokenization is done on the entire text. Then, we perform Span-to-BIO mapping to obtain the labels of each token in the document. Since we have eight named entity tags, so after this step, each token will be mapped to one of 17 BIO_tags (2 * 16 + 1). Next, the chunking operation is performed (as described above) to prepare the batches. Using the PLM, we obtain the contextualized token embeddings for each token of any input chunk (C_i), and then a fully connected linear layer with dropout and softmax activation function is applied

to the embeddings to obtain class-wise probability scores. Following that, it's passed to a CRF layer since it works well on sequence labelling tasks with a balanced dataset by taking into account the label-label interaction using transition scores (probability of a label given its previous label). After predicting the tokenwise labels, BIO-to-Span reverse mapping is done to extract the named entities predicted by the model.

Based on our legal expert's advice and our data exploration, we discovered that the entities belonging to the tags 'BenchCoram', 'JudgmentDate', 'Court' and 'CaseNo' generally have some identifiable cure phrases occurring preceding them or as part of them. Also, these entities occur either in the preamble section or at the extreme bottom of the document, so we have a fixed set of locations to look for this information. Some examples of the cue phrases are given as follows:

(1) If the entity 'JudgmentDate' is present in the preamble section, then it's generally preceded by "Date: " or "DATED: " etc. If it's present in the "judgment-text" section, it typically occurs in the last 2-3 lines of the document without any other text around it.

(2) If the entity 'BenchCoram' is present in the preamble section, then it's generally preceded by "Bench: " or "Bench Coram: " etc.

(3) The entity 'CaseNo' must occur in the first few lines of the preamble section, and it consists of some legal terms like 'writ petition' or 'Crr' or 'Crl' etc.

Utilizing this property, we designed a rule-based approach to extract the named entities for these four tags. Since these cue phrases can vary a lot based on the court jurisdiction and the type of case, we have curated a list of rules for each court in the dataset. Given an input case judgment document, first, we need to predict its 'Court' name and, based on that court, specific rules for other tags applied to it. Once strings are extracted based on matches with the rules, those strings should be post-processed to remove any extra space or unnecessary characters hanging around the target entity value. After extracting entities using this rule-based method, they are combined with the model's prediction set.

4.3 Baselines: LSP

In the baseline models, the label shift prediction task is proposed as a sequence classification task where the task is to predict whether the rhetorical role of a sentence will shift from its previous sentence S_{i-1}, given any input sentence S_i (1 represents 'Shift' and 0 represents 'No Shift'), for all sentences in the input sequence (representing a case judgment document).

In the first baseline, we use a BERT-CRF architecture commonly adopted in various classification tasks. Here, a batch of input documents is used as input to the model 'Indian-LegalBERT', and the output [CLS] embeddings of the sentences are passed to a linear layer (with Sigmoid activation) followed by CRF layer to take into account the label sequence dependencies. Here, the negative log-likelihood loss is used as it is well suited to the top layer having CRF.

The second baseline builds upon the first baseline by processing the [CLS] embeddings of the sentences in a Bidirectional LSTM layer to capture the contextual information across sentences.

The third baseline is a variation of the second baseline, where the CRF layer is removed, and a weighted cross-entropy loss is applied to the outputs of the fully connected layer.

4.4 Proposed Models: LSP

In the proposed models, the label shift prediction task is modelled as a sequence classification task, where each sequence element is a sentence pair. Given any sentence pair {Si-1, Si}, we need to predict a binary value representing whether they have the same rhetorical role or not (0 denotes 'Same' and 1 denotes 'Not Same'). A high-level encoder is encompassed in all proposed models to ensure the contextual information across an entire document is considered while predicting the labels for each input text.

The first proposed model (given in Fig. 4.2) builds upon the third baseline model, which has a BERT-BiLstm architecture. After generating the contextualised sentence embeddings for each sentence Si, the embedding of each sentence pair (S_{i-1}, S_i) is concatenated to get a joint embedding required for sentence pair classification. Following that, the linear classification head (with sigmoid activation layer) gets the label-wise probabilities, which are trained using weighted cross-entropy loss.

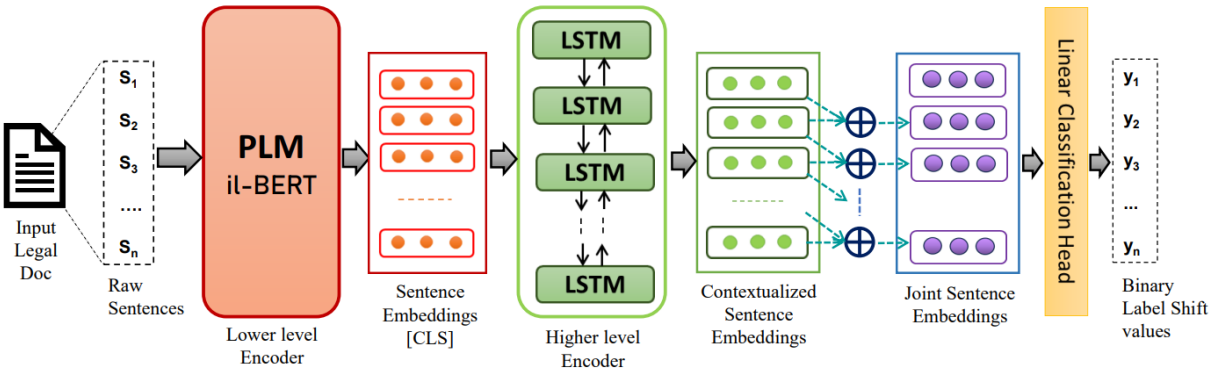

Fig. 4.2: Proposed LSP model-1

For the second proposed model, we process the input data differently (before passing it to the model) than the above described models. Here, each instance of the document is represented as "S_{i-1} [SEP] S_i ", which implies that each pair of adjacent sentences is concatenated with spaces and a separator token in between. The intuition behind this is to allow the model to apprehend the relationship and semantic similarity of consecutive sentences. The model architecture has the pretrained language model 'Indian-LegalBERT' as a low-level encoder followed by a Bidirectional Lstm layer in its core. It has two variants, with or without the topmost CRF layer, as shown in Fig. 4.3 and Fig. 4.4, respectively, similar to the second and third baselines.

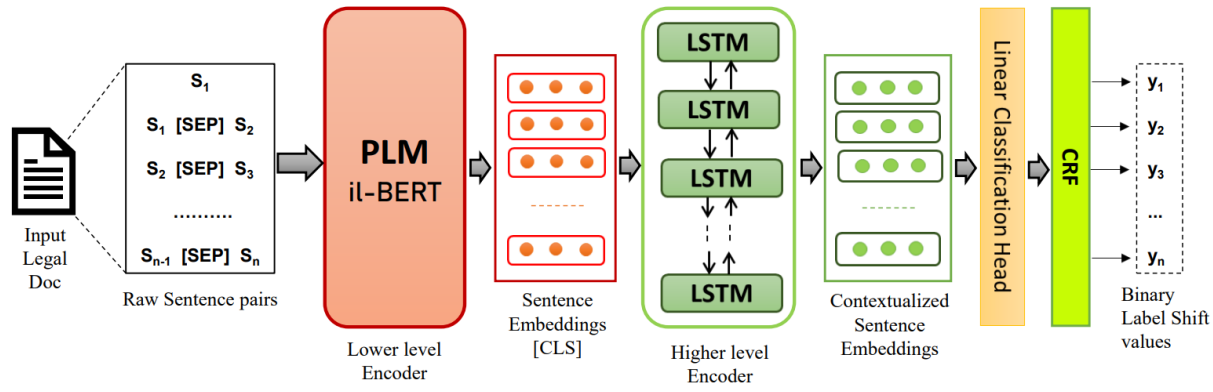

Fig. 4.3: Proposed LSP model-2(a)

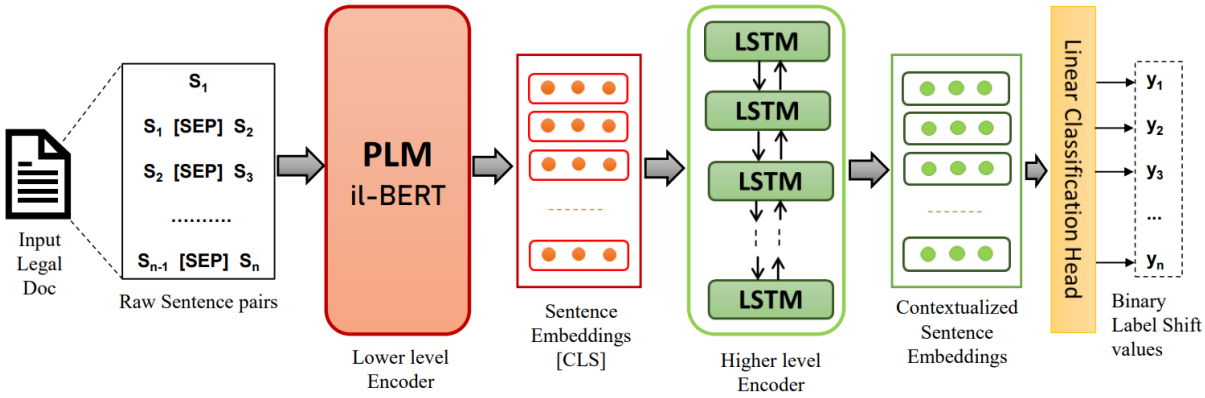

Fig. 4.4: Proposed LSP model-2(b)

Implementation Details:

For all of the models described in this section, we use different initial learning rates for different layers (1e-5 for the Bert model, 3e-4 for the LSTM layer and 1e-3 for the fully connected layers). Also, we use a cosine scheduler with warm steps (20% of the total number of steps). For all the baselines and the proposed model 1, the maximum sequence length of the Bert model is set to 256 tokens, whereas for the 2nd proposed model, it is set to 510 to avoid truncation since it uses longer input texts. These models are trained for 20 epochs (with early stopping).

4.5 Baselines: RR Identification

The baselines of the rhetorical role prediction task start with simple models like finetuning different BERT-based models for sentence classification. Along with the generic "BERT-base" model [23], we have used two other legal domain-specific pretrained language models for this named "Indian-LegalBERT" and "Indian-CaseLawBERT" [4]. The first few models treat each sentence discretely for the prediction task, and it does not take into account which document it belongs to and the contextual information. A variation of it uses the default cross-entropy loss. Another variation uses weighted cross-entropy loss, where weights are inversely proportional to the class frequencies.

Next, we use a set of baselines where this task is formulated as a sequence labelling task, where given a list of sentences belonging to a document as an input, the model has to predict the rhetorical roles for every sentence. The first baseline in this category uses a BERT-CRF architecture, where the [CLS] embeddings obtained from a BERT-based model are

forwarded to a fully connected classification layer with Softmax activation. Following that, the CRF layer works on the label dependencies to get the final set of predicted labels. The subsequent baselines are implemented using the best existing works for the rhetorical role prediction task.

The first model in the existing work is based on a hierarchical LSTM-CRF model (**HierBiLSTM-CRF**). This model uses a Sent2Vec encoder (trained on Indian Supreme Court documents) to obtain the sentence embeddings of an input document. These embeddings are then processed in the Bidirectional LSTM network to get contextually aware sentence-level embeddings. Those are fed to a classifier head, followed by a CRF layer to get label predictions using emission (probability of a label given the sentence) and transition scores (probability of a label given the previous label).

The next model (**HierBERT-CRF**) was proposed by the same author [4], where they introduced "Indian-LegalBERT" and "Indian-CaseLawBERT" as pretrained language models in the legal domain. The "Indian-LegalBERT" or "Indian-CaseLawBERT" model is used to get the sentence embedding. On top of that, the bidirectional LSTM and CRF layer works similarly as described above to get the final set of predictions. We tried another variant of this architecture (**HierEncoder-CRF**) by replacing the bidirectional LSTM layer with a transformer encoder layer as a high-level encoder working on top of the 'BERT' model.

The last state-of-the-art approach for this task [5] frames it in a multitask learning framework, as described in section 2. The architecture uses a frozen 'BERT' as a low-level sentence encoder. Here, the embeddings of sentences S_{i-1} and S_{i+1} generated by the top layer of the LSP model are concatenated before and after, respectively, with the contextualized embedding of sentence S_i from the RR model. Through this expanded embedding format for each sentence and a combined loss (loss of LSP and RR task), the model tries to hint at the classification layer of the relationship among adjacent sentences and their label interactions. Unlike other models discussed here, the layers of the 'BERT' model are not trained. We have also experimented with a minor variation of this model by adding better optimization techniques like using layer-wise different learning rates and using the initial embeddings from specialized PLM "Indian-LegalBERT".

4.6 Proposed Models: RR Identification

We have tried to address the class imbalance issue in our RR dataset in our proposed models by formulating a dynamically weighted loss. Here, we have used weighted cross-entropy loss for the multiclass classification, where the weight vector is dynamic during the training. The weight vector has two components. The first component is static, which is based on the effective frequency of a label 'lb' (representing a rhetorical role) as given by w^1_{lb}. This component will help the model pay attention to the less frequent labels of the dataset, which are rarely seen by the model during the training phase, using higher weightage for them. The second component, w^2_{lb}, is inversely proportional to the F1 score of every label, which will vary during the training epochs. The final weight is a dynamic vector w_{lb} having a weighted sum of these two components, where the weights are hyperparameters.

$$w^1_{lb} = \frac{n(S_{lb})}{n(S)}, \qquad S_{lb} = set\ of\ sentences\ with\ label\ "lb",$$
$$S = all\ the\ sentences$$

$$w^2_{lb} = \frac{1}{f_{lb}}, \qquad f_{lb} = F1\ score\ of\ label\ "lb"\ in\ the\ current\ epoch$$

$$\boldsymbol{w_{lb} = \alpha w^1_{lb} + \beta w^2_{lb},}$$

$$\alpha, \beta\ are\ constant\ hyperparameters,$$
$$w_{lb} is\ the\ dynamic\ weight\ of\ label\ "lb"$$

All of the proposed models consist of the "Indian-LegalBERT" model as a low-level encoder in their architecture. The models are explained as follows:

HierBERT-LGLDL

It is a Hierarchical BERT with Lsp Guided Linear classification head and Dynamically weighted Loss. This model has three variations, 1(a), 1(b) and 1(c), which differ based on the hierarchical layers used to generate sentence embeddings. All three variants utilize the [CLS] embeddings of sentences as well as individual token embeddings generated by the low-level encoder to fetch a richer embedding for each sentence to be classified. The [CLS] embeddings give a global representation of any sentence (e_i), and hence, it's commonly used for classification tasks. Additionally, an attention-based pooling mechanism applied to the token-wise embeddings may aid in capturing some local features of a sentence in the embedding (f_i). Thus, combining these two, using embedding concatenation ($e_i \oplus f_i$), can better represent a sentence overall.

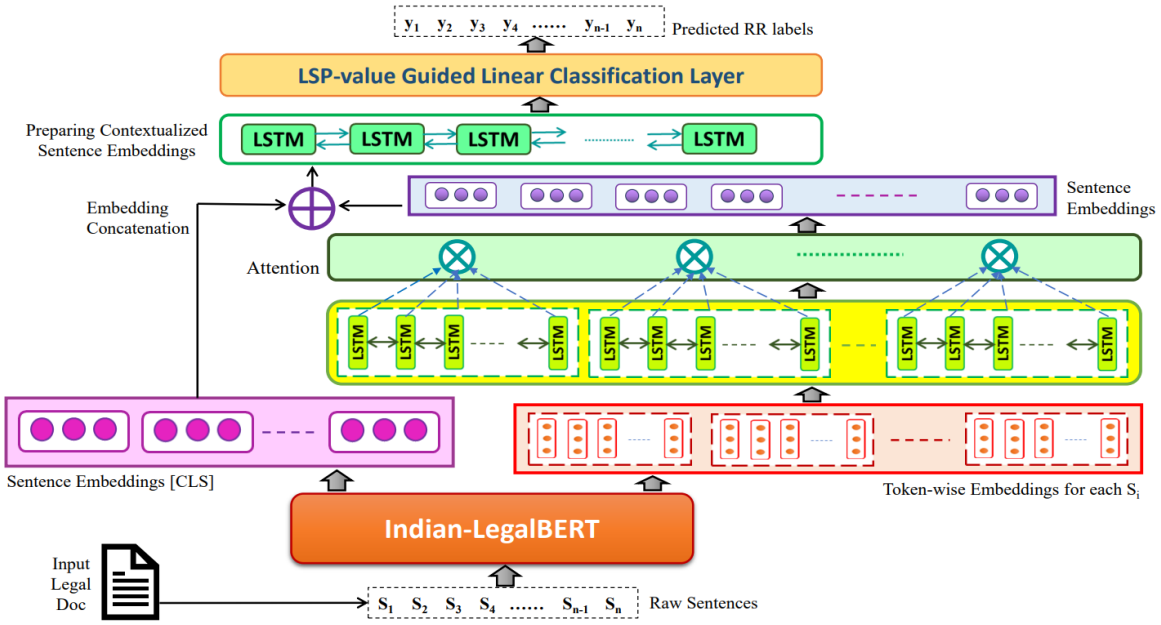

Fig. 4.5: RR Model - HierBERT-LGLDL (1a)

In the first variant, as shown in Fig. 4.5, after obtaining the token embeddings from the "Indian-LegalBERT" model, those are passed to an LSTM module with an Attention Head based on Bahdanau Attention [22]. The Bidirectional LSTM layer further enriches the token embeddings by seeing through the context. Then, the attention module learns to recognize frequently used tokens in sentences with specific roles and produces a single representation with these local features of the sentences. In the second variant (given in Fig. 4.6), the embedding of the last LSTM cell is used as the single representation instead of having the attention head. The following variant of this architecture (given in Fig. 4.7) does not include the LSTM layer processing the token-wise embeddings. It directly applies attention-based pooling on the token embeddings fetched from the lower encoder.

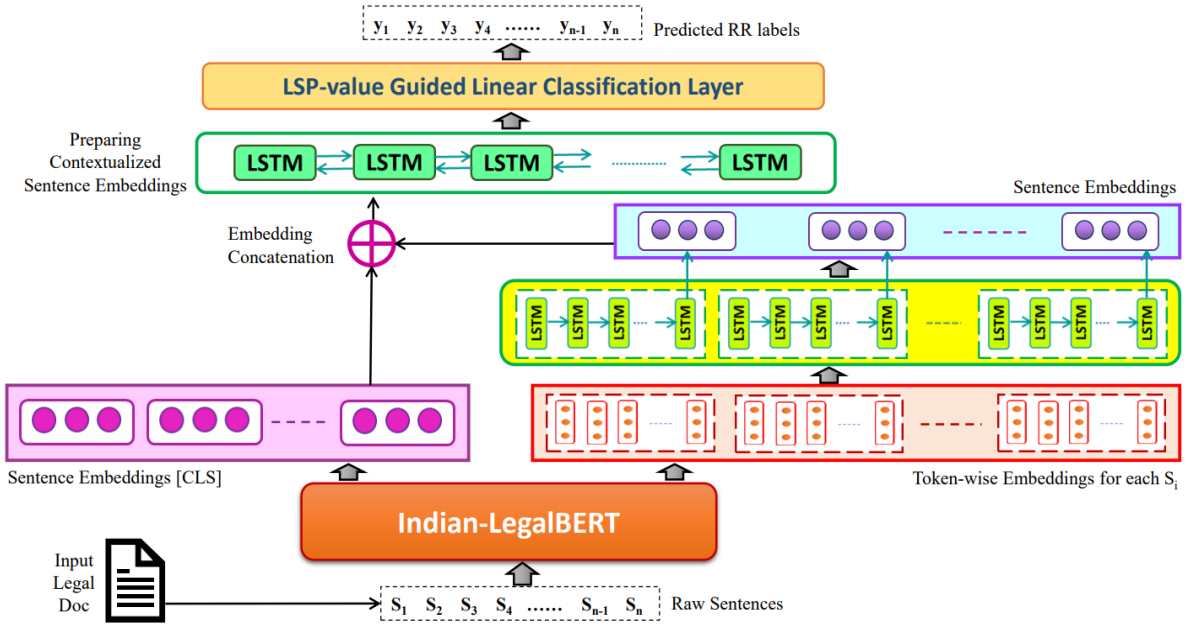

Fig. 4.6: RR Model - HierBERT-LGLDL (1b)

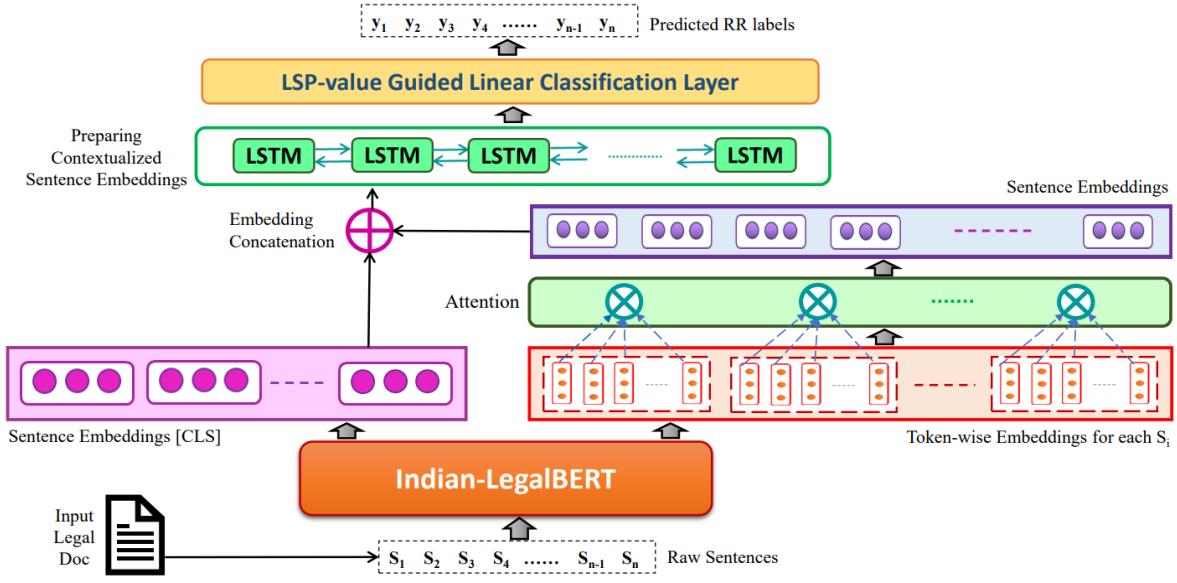

Fig. 4.7: RR Model - HierBERT-LGLDL (1c)

After getting the combined sentence-wise embeddings ($e_i \oplus f_i$), they are fed to the high-level encoder consisting of a bidirectional LSTM network. This high-level encoder generates a sequence of contextually aware embeddings representing the entire document. The fully connected layer at the top uses these embeddings to obtain the class-wise scores. Here, we have used the label shift predictions (obtained from the best-performing proposed LSP model) to produce a hybrid vector of logit scores. If the label of the current sentence is not shifted from its previous sentence label (i.e., $LSP(S_i) = 0$), then the logit scores P_i of the current sentence S_i are updated to a weighted sum (with two learnable parameters as weights) of the present scores and the logit scores P_{i-1} of the previous sentence S_{i-1}. The intuition for this is to make the logits of a sentence dependent on the last sentences when the sentence pair is expected to have the same rhetorical roles. The softmax activation layer is applied to the final logit scores to predict the target output y_i.

$$y_i = \begin{cases} \boldsymbol{argmax(\mu P_i + \lambda P_{i-1})\ if\ LSP(S_i) = 0} \\ \boldsymbol{argmax(P_i)\ otherwise} \end{cases}$$

$$where, y_i \textit{ is the predicted label of sentence } S_i,$$
$$\mu \text{ and } \lambda \textit{ are learnable parameters,}$$
$$P_i \textit{ is the list of label probabilites for the sentences } S_i,$$
$$LSP(S_i) \textit{represents the label shift value of sentence } S_i$$

HierBERT-LGCIDL

The next proposed model is based on a Hierarchical BERT with Lsp Guided Chunked Input. Here, the model architecture (as shown in Fig. 4.8) is very similar to an existing work [4], with a difference in how the input data is prepared. In the existing method, a sequence of sentences belonging to a legal document is directly taken as input by the lower-level encoder. In our proposed approach, a sequence of text chunks is used as input data, where the chunks are prepared by appending a context before every sentence S_i with a separator token in between. The context of a sentence is significant for the rhetorical roles prediction task since, in many examples, the role of a particular sentence is heavily dependent on the texts given before it. In addition to the high-level encoder that tries to capture the context around any sentence, feeding sentences to the low-level encoder with a short context for every sentence will help the model prepare the correct interpretation for the inputs in this task.

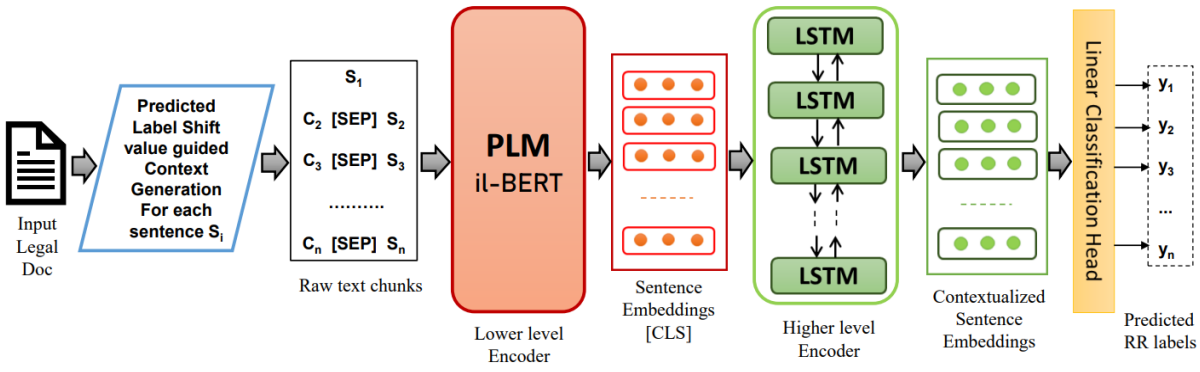

Fig. 4.8: RR model - HierBERT-LGCIDL

Defining this context for each sentence is the actual contribution to this model. A simple approach called "HierBERT-CIDL" considers 'n' sentences preceding any sentence S_i as its context, where 'n' is a hyperparameter. A more thoughtful approach is to use up to 'n' sentences as the context of sentence S_i if the labels of all of those 'n' sentences are predicted to be the same as of S_i. The predicted label shift values can be exploited to sense these label relationships among a fragment of consecutive sentences. Also, it is required to consider that the total number of tokens present in the chunk should be less than 512 to avoid truncation by the BERT-tokenizer due to exceeding the maximum token length limit. The "Prepare_Data()" method below describes the steps to perform this LSP value-guided chunking of sentences given any legal document D as input. As usual, the proposed dynamically weighted cross-entropy loss is applied to the model instead of using CRF with negative log-likelihood loss at the topmost layer.

```
Prepare_Data(D):
        for each Si ∈ D
                j = i
                k = i-1
                while j > 0 and LSP(Sj) !=1 and  (i-j+1) <= 5:
                        j = j-1
                CSi = Si
                while k >= j:
                        if CSi == Si :                          // concatenates Sj and CSi
                                temp = Sk + '[SEP]' + CSi       // with a separator in between
                        else:
                                temp = Sk + CSi                 // concatenates Sj and CSi
                        token_count = length(tokenize(temp))    // gets count of tokens in temp
                        if token_count < 512:
                                CSi = temp
                        else:
                                break
                        k = k-1
                D[i] = CSi                          // updating the ith sentence of document D
        return D
```

HierPEncoder-LGCIDL

This model can be described as Hierarchical Pretrained Encoder with Lsp Lsp-guided chunked Input and Dynamically weighted Loss. In terms of the overall approach of preparing input and processing the input in multiple layers of the model architecture, this model (given in Fig. 4.9) is very similar to the model "HierBERT-LGCIDL" described above. Here, the bidirectional LSTM network is replaced with another pretrained language model called "Longformer" [21] , functioning as the high-level encoder. Since the number of sentences in a legal document can exceed 512 (which is the maximum token length allowed by any BERT-base model), here we cannot use any pretrained encoder model with legal domain knowledge like "LegalBERT" [20], "Indian-LegalBERT" or "Indian-CaseLawBERT" [4] as the high-level encoder. "Longformer" accommodates up to 4096 input tokens or embeddings, enabling our model to simultaneously process all sentences in a document, just like the LSTM network was helping. Additionally, this transformer-based pretrained language model may aid in generating a better contextually aware representation of the sentence using its unique architecture and knowledge.

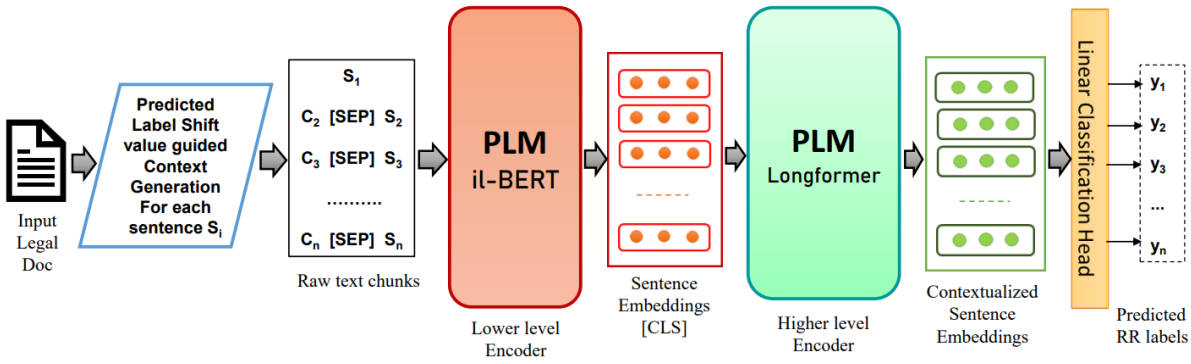

Fig. 4.9: RR model - HierPEncoder-LGCIDL

HierBERT-CADL

The first experimental method is named Hierarchical BERT with Contextual Attention and Dynamically weighted Loss. Like the above two models, this model also tries to use LSP values differently for the context generation of a sentence. Here, the input to the model is a sequence of sentences. Context-aware sentence embeddings are obtained after passing through the low-level and high-level encoder (BiLSTM network here). An attention-pooling layer is applied to it to enhance its context further. For every sentence S_i, this layer considers up to 'n' sentences before it, which are expected to have the same label as S_i based on label shift prediction. The Bahdanau Attention [22] is applied to this chunk of sentence embeddings to produce a single contextual representation for S_i. The top fully connected classification layer then uses these representations to predict the role y_i.

Implementation Details:

For all of the models described in this section, we use different initial learning rates for different layers (1e-5 for the Bert model or longformer, 3e-4 for the LSTM layer and 1e-3 for the fully connected layers). Also, we use a cosine scheduler with warm steps (20% of the total number of steps). For all the baselines and most of the proposed models, the maximum sequence length of the Bert model is set to 256 tokens, except for the last variations of proposed models providing chunked input to the BERT model. For this set of three models, it is set to 510 to avoid truncation since it uses longer input texts. The majority of the models are trained for 20 epochs (with early stopping). Only the models having transformer-based architectures as high-level encoders are trained for 25 epochs.

The learnable hyperparameters of the HierBERT-LGLDL model are set to the equal value pair as (1,1) in one experiment and (2,2) in another. In both experiments, these values are

observed to be linearly decreasing over the training epochs, and their final value after training completion differs by 0.06 and 0.03, respectively.

Chapter 5

Results

5.1 Legal NER

The performance of the baseline and proposed hybrid legal NER solution is given in Table 5.1 below. The SpaCy model was found to have a decent partial accuracy and F1 score. However, it ultimately falls short of strict measures, which indicates that the model is very inefficient in capturing the complete entity, though having a high overlap with the desired entities. While the BERT-CRF approach works well, adding handpicked rules and post-processing incorporated a 7% improvement. In the tag-wise performance graph (shown in Fig. 5.1), tags like "JudgmentDate" and "Court", on which rules could be applied, achieved excellent performance. Also, the strict and partial accuracy is comparable for these tags, whereas, for the rest of the tags, it does not hold.

Performance Metric	Spacy legal-NER	BiLSTM-CRF	BERT-CRF	BERT-CRF + Rules
Strict match Accuracy	0.22	0.51	0.74	**0.81**
Strict match F1 (micro)	0.2	0.59	0.81	**0.82**
Partial match Accuracy	0.92	0.82	0.96	**0.99**
Partial Match F1 (micro)	0.88	0.7	0.88	**0.97**
Average Partial match degree(%)	81.49	74.12	94.54	**94.88**

Table 5.1: Results of Legal Named Entity Extraction

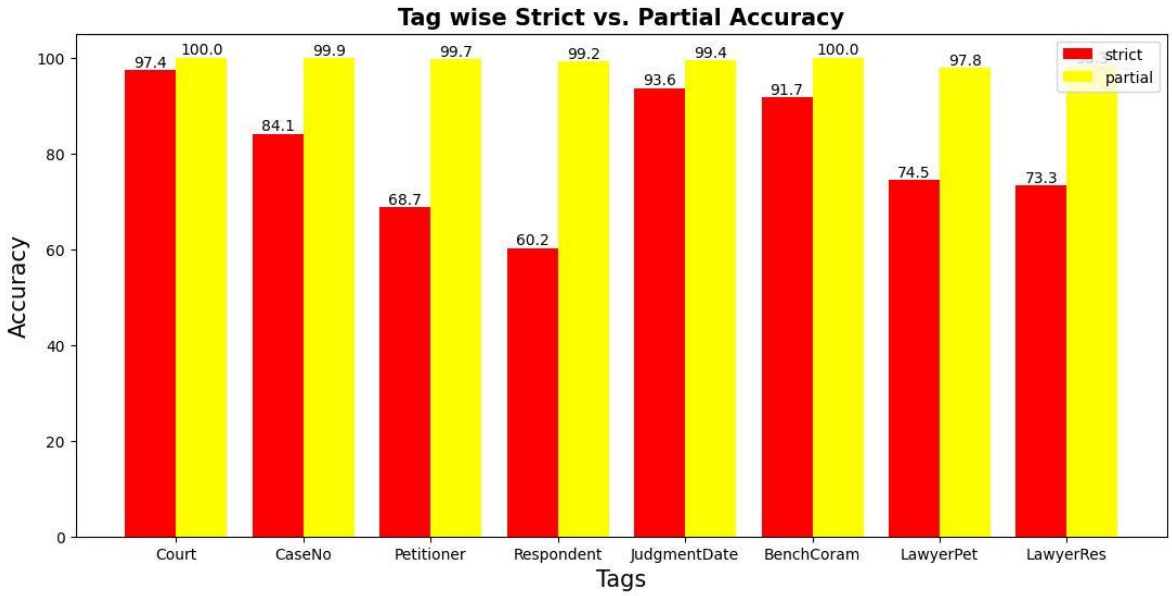

Fig. 5.1: Tag-wise Accuracy for Legal NER

5.2 Semantic Segmentation: LSP

The LLM's performance for the LSP task is moderately good, considering that the LLM is not finetuned. In the baseline models where a sequence of sentences represents the document, the HierBERT architecture with CRF on the top layer performs better than the HierBERT model with weighted cross-entropy loss. However, in the proposed approach, where the document encompasses a sequence of sentence pairs, the HierBERT model with weighted cross-entropy loss (HierBERT-SPCCI) performs far better than the CRF variant (as shown in Table 5.2). Also, the proposed HierBERT-SPC model fails to perform as expected. This result implied that the low-level encoder is capable of generating a much better representation, capturing the semantic meaningfulness of the sentence pairs when the two sentences are concatenated at the input level since it will enable the model to see through all the tokens of both sentences simultaneously resulting in a more appropriate [CLS] embedding for the LS prediction.

Model No.	Model	Macro-F1	Macro-P	Macro-R	Accuracy
B1	LLM infer (Mistral)	0.55	0.55	0.57	0.64
B2	BERT-CRF (SC)	0.71	0.7	0.72	0.8
B3	HierBERT-CRF (SC)	0.74	0.74	0.74	0.83
B4	HierBERT (SC)	0.72	0.71	0.72	0.81
E1	HierBERT (SPC)	0.72	0.71	0.74	0.81
E2	HierBERT-CRF (SPCCI)	0.74	0.73	0.76	0.83
E3	HierBERT (SPCCI)	**0.77**	**0.79**	**0.75**	**0.86**

Table 5.2: Results of Label Shift Prediction Models

5.3 Semantic Segmentation: RR Identification

Model No.	Model	Macro-F1	Macro-P	Macro-R	Accuracy
B1	**LLM infer (Mistral)**	0.29	0.34	0.36	0.37
B2	**BERT**	0.4	0.42	0.41	0.54
B3	**ilBERT**	0.42	0.44	0.41	0.55
B4	**iclBERT**	0.41	0.43	0.42	0.55
B5	**iclERT** (S-wLoss)	0.48	0.51	0.5	0.57
B6	**iclBERT** (S-wLoss)	0.48	0.51	0.47	0.57
B7	**ilBERT-CRF**	0.53	0.56	0.54	0.62
B8	**iclBERT-CRF**	0.52	0.54	0.53	0.62
B9	**HierBiLSTM-CRF**	0.44	0.45	0.44	0.56
B10	**HierBERT-CRF** (ilBERT)	0.6	0.61	0.64	0.71
B11	**HierBERT-CRF** (iclBERT)	0.58	0.59	0.6	0.71
B12	**MTL** (BERT)	0.5	0.54	0.49	0.62
B13	**MTL** (ilBERT)	0.58	0.61	0.58	0.69
B14	**HierEncoder-CRF** (iclBERT)	0.53	0.54	0.54	0.62
E1	**HierBERT-CADL**	0.63	0.63	0.67	0.66
E2	**HierBERT-LGLDL** (1a)	**0.69**	**0.71**	0.71	**0.75**
E3	**HierBERT-LGCDL** (1b)	0.56	0.55	0.71	0.63
E4	**HierBERT-LGCDL** (1c)	0.68	0.71	0.71	0.73
E5	**HierBERT-CIDL**	0.65	0.65	0.69	0.73
E6	**HierBERT-LGCIDL**	**0.69**	0.7	**0.72**	0.72
E7	**HierPEncoder-LGCIDL**	0.66	0.66	0.71	0.65

Table 5.3: Results of Rhetorical Roles prediction models

The performance of all the models of rhetorical role identification is given in Table 5.3. The first part of the table, with the models numbered from B1 to B14, represents the baseline results, and in the following part, the models numbered from E1 to E7 are the performances of the proposed RR prediction models.

First of all, it is noticed that the LLMs perform very poorly in this task. It implies that LLMs find it challenging to make sense of intricate legal language and the nuanced differences between multiple rhetorical roles. So, domain-specific finetuning will be required for an LLM to do well.

In the initial set of baselines where the task is modelled as a generic multi-class sentence classification by finetuning BERT-based models, the models pretrained, especially on the legal data, perform better than the regular BERT-base. Then, adding weighted loss (using the effective frequency of labels) into this setup gives around a 6-7 % boost in the macro-F1, which conveys that the model is heavily affected by the class imbalance issue.

Next, we see a further performance improvement when the task is performed as a sequence labelling operation using a BERT-CRF architecture, considering the sentences as a sequence unit. Here, the model gets [CLS] representations using the BERT-based encoder, which is used directly for emission score calculation at its classifier head. Additionally, the CRF layer considers the transition scores of different labels to make the final prediction. With the introduction of a high-level encoder in "HierBERT-CRF" **[4]** to contextualize the [CLS] sentence embeddings obtained from the BERT-based model, we see a 6-7% increase in the macro-F1 score again. This improvement signifies the importance of context for this task.

Also, it is observed that the "HierBERT-CRF" architecture outperforms the **"**HierBiLSTM-CRF" model with sent2vec embeddings (pretrained on legal corpus) in our experimentation, unlike the result analysis shown in the paper proposing "HierBERT-CRF". The possible reason for this is the dataset used in the previous work was extremely insufficient for a BERT-based architecture to learn well and gauge its performance in comparison to their earlier work, **"**HierBiLSTM-CRF" [9]. In our observation, once we get a significantly larger dataset to finetune the BERT-based models (pretrained on a vast Indian legal corpus), it can produce richer embeddings than the pretrained sent2vec encoder. This hypothesis was also mentioned in the work of [4], which came out to be true in our experiments. Overall, the “HierBERT-CRF” model with “Indian-LegalBERT” used as a low-level encoder gave us the best baseline performance of 0.60 macro-F1 score.

However, our experimented model, "HierEncoder-CRF", having a similar architecture to "HierBERT-CRF" with a transformer encoder layer as the high-level encoder, did not work as well as expected. We suspect that this is probably due to the reason that our dataset was not large enough to properly train a heavy encoder of the transformer, though this dataset manages to work with the bidirectional LSTM network used in "HierBERT-CRF".

Also, we noticed that the multi-task learning framework for RR prediction [5], which was proposed as the latest state-of-the-art model, underperforms significantly in our experiments. In the provided implementation of this approach, we find that the BERT model was not trainable, and the embeddings of sentences are directly inferred, which justifies the cause of lower performance. We performed an experiment modifying the provided implementation of the MTL approach by replacing the underlying BERT-base model with "Indian-LegalBERT" to get more enriched [CLS] sentence embeddings and adding better optimization steps like layer-wise varying learning rate and scheduler, which are also used in our proposed approaches. Doing these modifications aids in an 8% improvement on macro-F1 compared to the given implementation with the default BERT-base, and the model's overall performance becomes comparable to that of the other existing work, "HierBERT-CRF".

Our first experimental model, “HierBERT-CADL”, shows around 3% improvement over the baseline method. It implies that explicitly adding context to a sentence is an approach that helps get better sentence representation. However, the contextual attention pooling method used in this model is probably not the best way to do this.

The following experimental model is one of the proposed models, i.e., "HierBERT-LGLDL." We see a significant improvement of 8-9% in the first and third variants of this model (1a and 1c), which have applied attention pooling to extract local features from the token embeddings of BERT. So overall, this amalgamation of joint sentence embedding, LSP-guided combined logit scores, and dynamically weighted loss enhance the identification of rhetorical roles. If we see the label-wise performance of the predictions (given in Fig. 5.2 and Fig. 5.3), it is observed that the roles present less than 5% in the corpus get enormous push using this method. Using the existing approaches, labels like "Authorities", "Issues", etc., earlier had f1-scores almost close to zero.

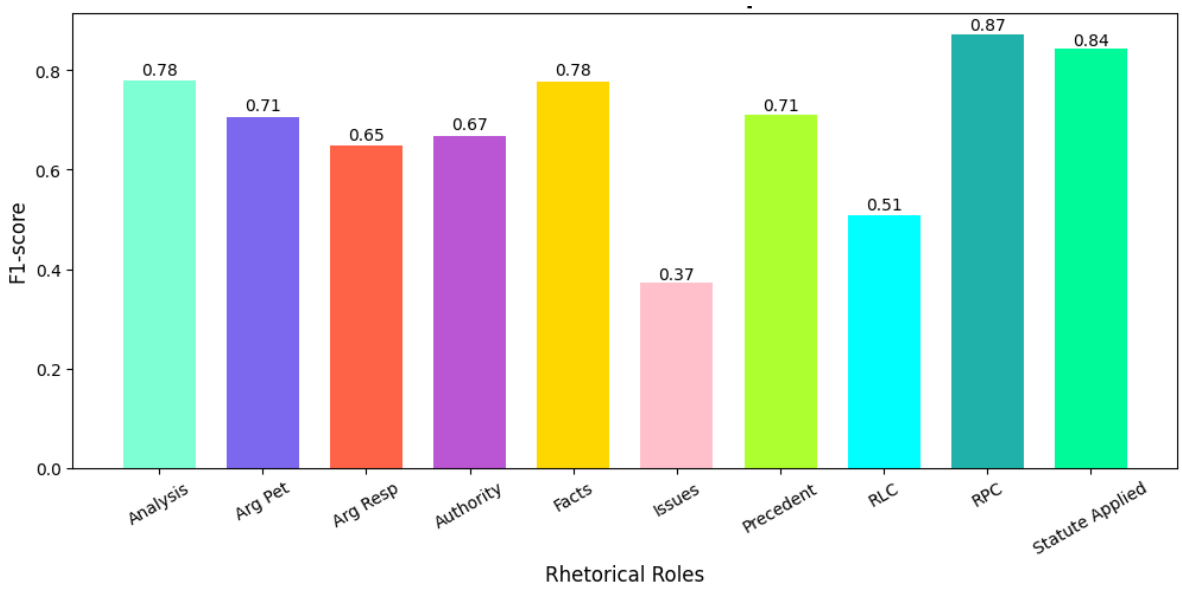

Fig. 5.2: Label-wise F1 scores for RR prediction

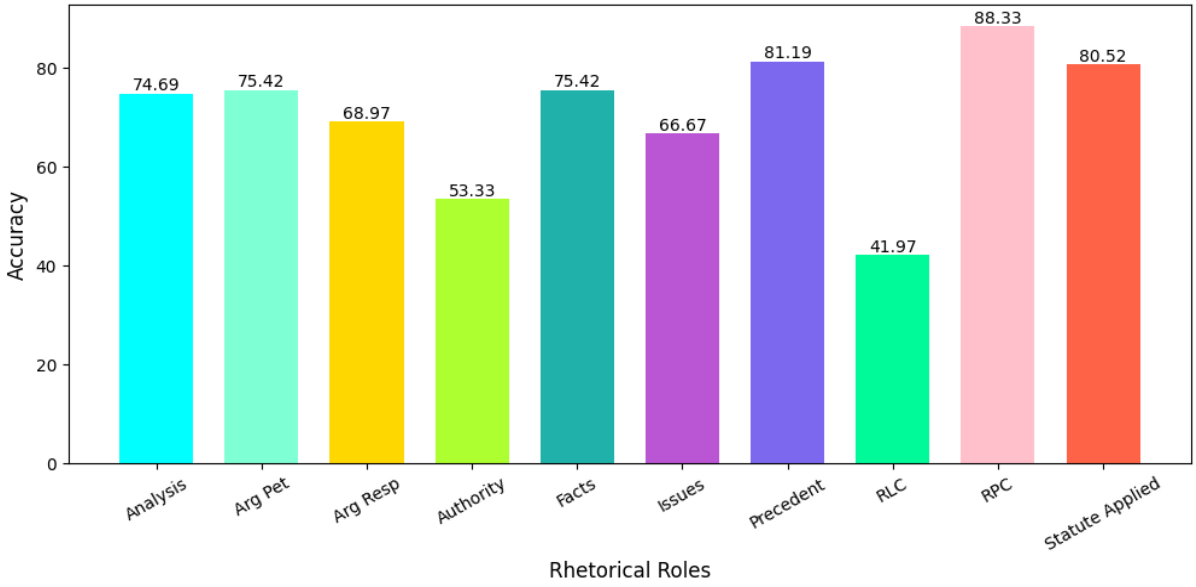

Fig. 5.3: Label-wise Accuracy for RR prediction

In our next category of models, instead of giving sentences of a document as input, a group of chunked inputs are fed to the low-level encoder for explicit contextualization of the individual sentences. We see that if we mandatorily chunk out every sentence as done in the model "HierBERT-CIDL", it aids improvement over the baseline but not as much as for the method using LSP-guided chunking on HierBERT, which again gives 9% improved performance than the best baseline method. The model "HierPEncoder-LGCIDL" uses a "longformer" as the high-level encoder. It is a pretrained transformer-based architecture, and consequently, the overall model performs way better than "HierEncoder-CRF", but it is still not comparable or better than the models with an LSTM network due to overfitting issues.

The confusion matrix of the labels (given in Fig. 5.4) is obtained using the RR prediction values from one of the best-performing models. We find out that many sentences with labels like “Analysis”, “Facts”, and “Precedent” are inherently confusing with many other labels, and sometimes the annotation done by legal professionals also differs for such sentences. Also, the document-wise F1-score distribution of RR labels is given in Fig. 5.5.

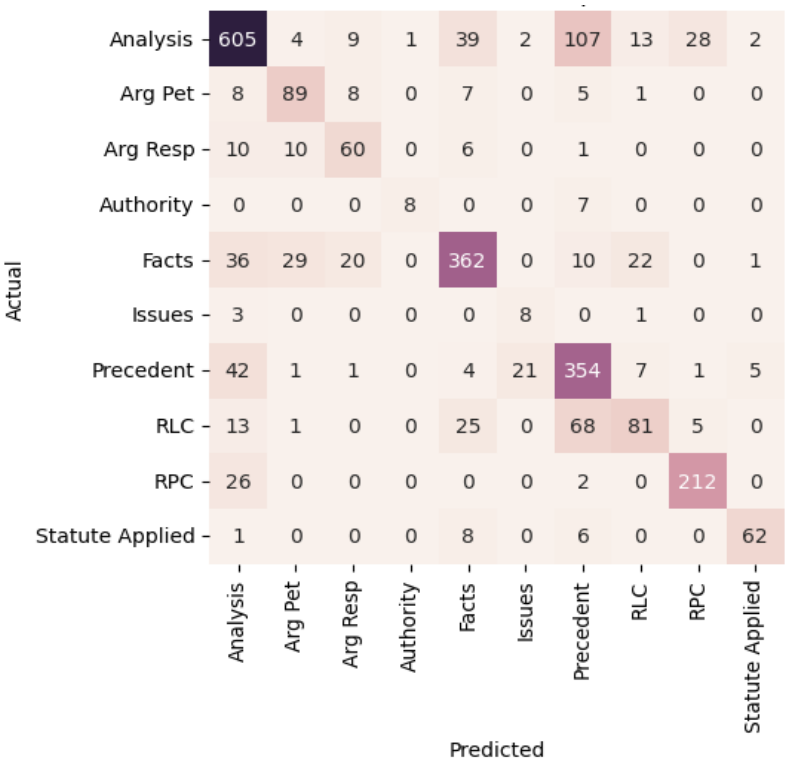

Fig. 5.4: RR prediction confusion matrix

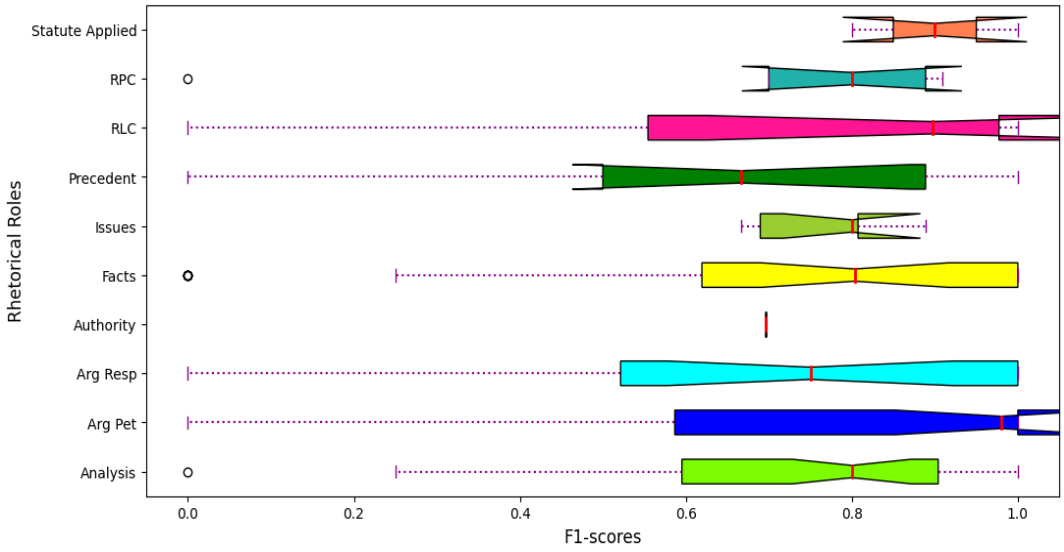

Fig. 5.5: Doc-wise F1-score distribution of RR labels

5.4 Semantic Segmentation: Cross Domain

In this section, we study how the best-proposed models perform on different domains of legislation other than on which it was trained. Also, propose a possible solution that will work best in such scenarios with minimum additional cost or effort. For these experiments, we use another RR annotated dataset having 100 cases coming from diverse domains like "criminal", "civil", "procedure and proposal", and so on. Out of these 100 cases, 50 are used for only test data, 40 for few-shot learning or finetuning, and 10 for validation. The first section of Table 5.4 shows experimental results for the LSP task, and the next section shows it for the RR prediction.

Initially, we perform zero-shot LS and RR prediction on the 50 cases using the trained models, giving the best results. Following this, we perform a few-shot learning using the same model architectures and test these models on the same 50 miscellaneous cases test dataset. The result scores obtained in these two experiments are pretty comparable for the RR prediction, whereas, for LSP, the zero-shot prediction performs way better than few-shot learning. The next experiment is done by finetuning the trained RR or LSP model

using the small 40 + 10 cases dataset and then testing the finetuned models on the 50 cases test set. This last experiment delivers a significant hike in performance metrics (as shown in Table 7). So, based on these experimentations, it can be concluded that models trained on a large corpus of a particular or some combination of legal domains can be beneficial for the prediction of cross-domain cases if they can be finetuned on a small corpus of the target domain, as essentially it has the capability of generalizability for the domain upto a specific limit.

Model	Task Performed	Macro-F1	Macro-P	Macro-R	Accuracy
HierBERT-SPC	Zero-shot LS prediction	0.78	**0.81**	**0.77**	**0.84**
HierBERT-SPC	Few-shot LS prediction	0.71	0.71	0.72	0.77
HierBERT-SPC	Finetuning LSP model	**0.79**	0.8	**0.77**	**0.84**
Proposed RR Model-1a	Zero-shot RR prediction	0.48	0.54	0.48	0.7
Proposed RR Model-2a	Zero-shot RR prediction	0.45	0.52	0.45	0.66
Proposed RR Model-1a	Few-shot RR prediction	0.47	0.51	0.45	0.7
Proposed RR Model-2a	Few-shot RR prediction	0.5	0.53	0.48	0.73
Proposed RR Model-1a	Finetuning RR model	**0.56**	**0.57**	**0.62**	**0.75**
Proposed RR Model-2a	Finetuning RR model	0.52	0.56	0.5	0.73

Table 5.4: Results of Rhetorical Roles prediction models

Chapter 6

Conclusion and Future Work

This work introduces a new large corpus of Indian High courts and Supreme Court case judgment documents for legal NER and rhetorical roles identification tasks. We have proposed a new, unique approach for the NER task, having an extensive collection of handpicked rules. We have extended the latest research work pursued for the RR prediction tasks in multiple ways that outperform the existing works. Also, we are the first to address the problem of class imbalance commonly observed in RR-annotated datasets and have successfully provided a decent solution for the same. Some key take away from our work are given as follows:

(1) The BERT-based models pretrained on the legal domain are extremely useful, and their embeddings are highly rich compared to any other embedding techniques used earlier.

(2) We find that dynamically weighted cross-entropy loss works better on the RR prediction task instead of having a CRF layer at the top of the architecture. This approach mainly applies only to scenarios where the dataset has a huge class imbalance.

(3) The Label shift prediction task is an excellent auxiliary task that can be exploited in the pipeline of rhetorical role prediction. The system's overall efficacy depends on the accuracy of the label shift prediction.

(4) The performance of models is quite dependent on multiple factors, such as the length of the case judgment document, frequency of label shifts, size of the training and testing dataset, and label distribution in each dataset split.

(5) Models are sensitive to the legal subdomains and jurisdictions with a medium level of generalizability in cross domain or jurisdictions. This ability can be utilized by finetuning existing models with a small dataset to get a solution on a new target domain.

Future work on these tasks can be explored by designing better loss to solve the class imbalance issue and better learning texts with confusing roles. Also, it will be worth experimenting with modelling the RR prediction task as a multi-label classification task.

Bibliography

[1] A. Vaswani *et al.*, "Attention Is All You Need." arXiv, 2017. doi: 10.48550/ARXIV.1706.03762.

[2] S. Skylaki, A. Oskooei, O. Bari, N. Herger, and Z. Kriegman, "Legal Entity Extraction using a Pointer Generator Network," in *2021 International Conference on Data Mining Workshops (ICDMW)*, Auckland, New Zealand: IEEE, Dec. 2021, pp. 653–658. doi: 10.1109/ICDMW53433.2021.00086.

[3] M. Saravanan and B. Ravindran, "Identification of Rhetorical Roles for Segmentation and Summarization of a Legal Judgment," *Artif Intell Law*, vol. 18, no. 1, pp. 45–76, Mar. 2010, doi: 10.1007/s10506-010-9087-7.

[4] S. Paul, A. Mandal, P. Goyal, and S. Ghosh, "Pre-trained Language Models for the Legal Domain: A Case Study on Indian Law," in *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law*, Braga Portugal: ACM, Jun. 2023, pp. 187–196. doi: 10.1145/3594536.3595165.

[5] V. Malik *et al.*, "Semantic Segmentation of Legal Documents via Rhetorical Roles," 2021, doi: 10.48550/ARXIV.2112.01836.

[6] A. Kapoor *et al.*, "HLDC: Hindi Legal Documents Corpus." arXiv, 2022. doi: 10.48550/ARXIV.2204.00806.

[7] P. Kalamkar *et al.*, "Corpus for Automatic Structuring of Legal Documents." arXiv, 2022. doi: 10.48550/ARXIV.2201.13125.

[8] Z. Huang, W. Xu, and K. Yu, "Bidirectional LSTM-CRF Models for Sequence Tagging," 2015, doi: 10.48550/ARXIV.1508.01991.

[9] P. Bhattacharya, S. Paul, K. Ghosh, S. Ghosh, and A. Wyner, "DeepRhole: deep learning for rhetorical role labeling of sentences in legal case documents," *Artif Intell Law*, vol. 31, no. 1, pp. 53–90, Mar. 2023, doi: 10.1007/s10506-021-09304-5.

[10] P. Bhattacharya, S. Paul, K. Ghosh, S. Ghosh, and A. Wyner, "Identification of Rhetorical Roles of Sentences in Indian Legal Judgments," 2019, doi: 10.48550/ARXIV.1911.05405.

[11] G. Zhou and J. Su, "Named entity recognition using an HMM-based chunk tagger," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02*, Philadelphia, Pennsylvania: Association for Computational Linguistics, 2001, p. 473. doi: 10.3115/1073083.1073163.

[12] L. Zheng, N. Guha, B. R. Anderson, P. Henderson, and D. E. Ho, "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*, São Paulo Brazil: ACM, Jun. 2021, pp. 159–168. doi: 10.1145/3462757.3466088.

[13] C. Xiao, X. Hu, Z. Liu, C. Tu, and M. Sun, “Lawformer: A Pre-trained Language Model for Chinese Legal Long Documents.” arXiv, May 09, 2021. Accessed: May 21, 2024. [Online]. Available: http://arxiv.org/abs/2105.03887

[14] C. Wang, B. Li, and W. Zhang, “Attention-BLSTM-CRF Based Method for Named Entity Recognition in Judicial Domain,” *J. Phys.: Conf. Ser.*, vol. 1616, no. 1, p. 012108, Aug. 2020, doi: 10.1088/1742-6596/1616/1/012108.

[15] T. Y. S. S. Santosh, P. Bock, and M. Grabmair, “Joint Span Segmentation and Rhetorical Role Labeling with Data Augmentation for Legal Documents,” in *Advances in Information Retrieval*, vol. 13981, J. Kamps, L. Goeuriot, F. Crestani, M. Maistro, H. Joho, B. Davis, C. Gurrin, U. Kruschwitz, and A. Caputo, Eds., in Lecture Notes in Computer Science, vol. 13981. , Cham: Springer Nature Switzerland, 2023, pp. 627–636. doi: 10.1007/978-3-031-28238-6_54.

[16] P. Kalamkar, A. Agarwal, A. Tiwari, S. Gupta, S. Karn, and V. Raghavan, “Named Entity Recognition in Indian court judgments,” in *Proceedings of the Natural Legal Language Processing Workshop 2022*, Abu Dhabi, United Arab Emirates (Hybrid): Association for Computational Linguistics, 2022, pp. 184–193. doi: 10.18653/v1/2022.nllp-1.15.

[17] Q. Huang, Y. Sun, Z. Xing, M. Yu, X. Xu, and Q. Lu, “API Entity and Relation Joint Extraction from Text via Dynamic Prompt-tuned Language Model,” *ACM Trans. Softw. Eng. Methodol.*, vol. 33, no. 1, pp. 1–25, Jan. 2024, doi: 10.1145/3607188.

[18] P. Henderson *et al.*, “Pile of Law: Learning Responsible Data Filtering from the Law and a 256GB Open-Source Legal Dataset.” arXiv, Nov. 29, 2022. Accessed: May 21, 2024. [Online]. Available: http://arxiv.org/abs/2207.00220

[19] A. Graves, S. Fernández, and J. Schmidhuber, “Bidirectional LSTM Networks for Improved Phoneme Classification and Recognition,” in *Artificial Neural Networks: Formal Models and Their Applications – ICANN 2005*, vol. 3697, W. Duch, J. Kacprzyk, E. Oja, and S. Zadrożny, Eds., in Lecture Notes in Computer Science, vol. 3697. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 799–804. doi: 10.1007/11550907_126.

[20] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, “LEGAL-BERT: The Muppets straight out of Law School,” in *Findings of the Association for Computational Linguistics: EMNLP 2020*, Online: Association for Computational Linguistics, 2020, pp. 2898–2904. doi: 10.18653/v1/2020.findings-emnlp.261.

[21] I. Beltagy, M. E. Peters, and A. Cohan, “Longformer: The Long-Document Transformer.” arXiv, Dec. 02, 2020. Accessed: May 21, 2024. [Online]. Available: http://arxiv.org/abs/2004.05150

[22] D. Bahdanau, K. Cho, and Y. Bengio, “Neural Machine Translation by Jointly Learning to Align and Translate.” arXiv, May 19, 2016. Accessed: May 21, 2024. [Online]. Available: http://arxiv.org/abs/1409.0473

[23] Devlin J, Chang MW, Lee K, Toutanova K (2019) Bert: Pre-training of deep bidirectional transformers for language understanding. In: proceedings of NAACL-HLT 2019 pp. 4171–4186, https://huggi ngface.co/bert-base-uncased
[24] Farzindar A, Lapalme G (2004) Letsum, an automatic legal text summarizing system. In: legal knowledge and information systems–JURIX, pp. 11–18
[25] Hachey B, Grover C (2006) Extractive summarisation of legal texts. Artif Intell Law 14(4):305–345
[26] Laferty JD, McCallum A, Pereira FCN (2001) Conditional random felds: probabilistic models for seg menting and labeling sequence data. In: proceedings of the eighteenth international conference on machine learning, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, ICML 01, pp. 282–289
[27] Shulayeva O, Siddharthan A, Wyner AZ (2017) Recognizing cited facts and principles in legal judge ments. Artif Intell Law 25(1):107–126
[28] Savelka J, Ashley KD (2018) Segmenting us court decisions into functional and issue specifc parts. In: legal knowledge and information systems–JURIX, pp. 111–120