

HLDC: Hindi Legal Documents Corpus

Arnav Kapoor[†], Mudit Dhawan[‡], Anmol Goel[†],
T.H. Arjun[†], Akshala Bhatnagar[‡], Vibhu Agrawal[‡], Amul Agrawal[†],
Arnab Bhattacharya[¶], Ponnuram Kumaraguru[†], Ashutosh Modi^{¶*}
[†]IIT Hyderabad, [‡]IIT Delhi, [¶]IIT Kanpur

{arnav.kapoor, anmol.goel, arjun.thekoot}@research.iit.ac.in
{mudit18159, akshala18012, vibhu18116}@iiitd.ac.in, amul.agrawal@students.iit.ac.in,
arnabb@cse.iitk.ac.in, pk.guru@iiit.ac.in, ashutoshm@cse.iitk.ac.in

Abstract

Many populous countries including India are burdened with a considerable backlog of legal cases. Development of automated systems that could process legal documents and augment legal practitioners can mitigate this. However, there is a dearth of high-quality corpora that is needed to develop such data-driven systems. The problem gets even more pronounced in the case of low resource languages such as Hindi. In this resource paper, we introduce the *Hindi Legal Documents Corpus (HLDC)*, a corpus of more than 900K legal documents in Hindi. Documents are cleaned and structured to enable the development of downstream applications. Further, as a use-case for the corpus, we introduce the task of bail prediction. We experiment with a battery of models and propose a Multi-Task Learning (MTL) based model for the same. MTL models use summarization as an auxiliary task along with bail prediction as the main task. Experiments with different models are indicative of the need for further research in this area. We release the corpus and model implementation code with this paper: <https://github.com/Exploration-Lab/HLDC>.

1 Introduction

In recent times, the legal system in many populous countries (e.g., India) has been inundated with a large number of legal documents and pending cases (Katju, 2019). There is an imminent need for automated systems to process legal documents and help augment the legal procedures. For example, if a system could readily extract the required information from a legal document for a legal practitioner, then it would help expedite the legal process. However, the processing of legal documents is challenging and is quite different from conventional text processing tasks. For example, legal documents are typically quite long (tens of pages),

highly unstructured and noisy (spelling and grammar mistakes since these are typed), use domain-specific language and jargon; consequently, pre-trained language models do not perform well on these (Malik et al., 2021b). Thus, to develop legal text processing systems and address the challenges associated with the legal domain, there is a need for creating specialized legal domain corpora.

In recent times, there have been efforts to develop such corpora. For example, Chalkidis et al. (2019) have developed an English corpus of European Court of Justice documents, while Malik et al. (2021b) have developed an English corpus of Indian Supreme Court documents. Xiao et al. (2018) have developed Chinese Legal Document corpus. However, to the best of our knowledge, there does not exist any legal document corpus for the Hindi language (a language belonging to the Indo-European family and pre-dominantly spoken in India). Hindi uses Devanagari script (Wikipedia contributors, 2021) for the writing system. Hindi is spoken by approximately 567 million people in the world (WorldData, 2021). Most of the lower (district) courts in northern India use Hindi as the official language. However, most of the legal NLP systems that currently exist in India have been developed on English, and these do not work on Hindi legal documents (Malik et al., 2021b). To address this problem, in this paper, we release a large corpus of Hindi legal documents (HINDI LEGAL DOCUMENTS CORPUS or HLDC) that can be used for developing NLP systems that could augment the legal practitioners by automating some of the legal processes. Further, we show a use case for the proposed corpus via a new task of *bail prediction*.

India follows a Common Law system and has a three-tiered court system with District Courts (along with Subordinate Courts) at the lowest level (districts), followed by High Courts at the state level, and the Supreme Court of India at the high-

*Corresponding Author

est level. In terms of number of cases, district courts handle the majority. According to India’s National Judicial Data Grid, as of November 2021, there are approximately 40 million cases pending in District Courts (National Judicial Data Grid, 2021) as opposed to 5 million cases pending in High Courts. These statistics show an immediate need for developing models that could address the problems at the grass-root levels of the Indian legal system. Out of the 40 million pending cases, approximately 20 million are from courts where the official language is Hindi (National Judicial Data Grid, 2021). In this resource paper, we create a large corpus of 912,568 Hindi legal documents. In particular, we collect documents from the state of Uttar Pradesh, the most populous state of India with a population of approximately 237 million (PopulationU, 2021). The Hindi Legal Documents Corpus (HLDC) can be used for a number of legal applications, and as a use case, in this paper, we propose the task of *Bail Prediction*.

Given a legal document with facts of the case, the task of *bail prediction* requires an automated system to predict if the accused should be granted bail or not. The motivation behind the task is not to replace a human judge but rather augment them in the judicial process. Given the volume of cases, if a system could present an initial analysis of the case, it would expedite the process. As told to us by legal experts and practitioners, given the economies of scale, even a small improvement in efficiency would result in a large impact. We develop baseline models for addressing the task of bail prediction.

In a nutshell, we make the following main contributions in this resource paper:

- We create a *Hindi Legal Documents Corpus (HLDC)* of 912,568 documents. These documents are cleaned and structured to make them usable for downstream NLP/IR applications. Moreover, this is a growing corpus as we continue to add more legal documents to HLDC. We release the corpus and model implementation code with this paper: <https://github.com/Exploration-Lab/HLDC>.
- As a use-case for applicability of the corpus for developing legal systems, we propose the task of *Bail Prediction*.
- For the task of bail prediction, we experiment with a variety of deep learning models. We propose a multi-task learning model based on trans-

former architecture. The proposed model uses extractive summarization as an auxiliary task and bail prediction as the main task.

2 Related Work

In recent years there has been active interest in the application of NLP techniques to the legal domain (Zhong et al., 2020a). A number of tasks and models have been proposed, inter alia, Legal Judgment Prediction (Chalkidis et al., 2019), Legal Summarization (Bhattacharya et al., 2019; Tran et al., 2019), Prior Case Retrieval (Jackson et al., 2003; Shao et al., 2020), Legal Question Answering (Kim and Goebel, 2017), Catchphrase Extraction (Galgani et al., 2012), Semantic Segmentation (Kalamkar et al., 2022; Malik et al., 2021a).

Legal Judgement Prediction (LJP) involves predicting the final decision from the facts and arguments of the case. Chalkidis et al. (2019) released 11,478 cases from the European Court of Human Rights (ECHR). It contains facts, articles violated (if any), and the importance scores. Malik et al. (2021b) provided 34,816 case documents from the Supreme Court of India for the prediction task. Strickson and De La Iglesia (2020) published 4,959 documents from the U.K.’s Supreme court (the highest court of appeal).

Majority of corpora for Legal-NLP tasks have been in English; recently, there have been efforts to address other languages as well, for example, Xiao et al. (2018), have created a large-scale Chinese criminal judgment prediction dataset with over 2.68 million legal documents. Work on Legal-NLP in languages other than English is still in its incipient stages. Our paper contributes towards these efforts by releasing corpus in Hindi.

Majority of the work in the legal domain has focused on the higher court (Malik et al., 2021b; Strickson and De La Iglesia, 2020; Zhong et al., 2020b); however, the lower courts handle the maximum number of cases. We try to address this gap by releasing a large corpus of district court level legal documents.

Some of the recent work has explored other Legal-NLP tasks in languages other than English. Chalkidis et al. (2021) released a multilingual dataset of 65K European Union (E.U.) laws for topic classification of legal documents. The data was translated into the 23 official E.U. languages and annotated with labels from the multilingual thesaurus, EUROVOC. Luz de Araujo et al. (2018)

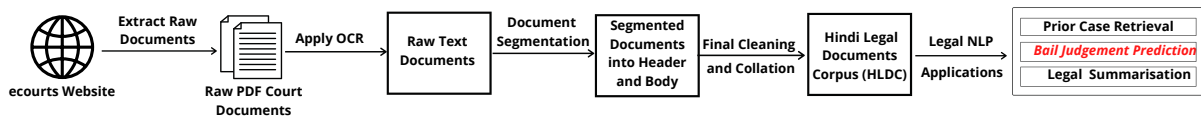


Figure 1: HLDC corpus creation pipeline

have released 70 documents in Portuguese for Legal Named Entity Recognition. The dataset contains specific tags for law and legal cases entities in addition to the normal tags for named entities like person, locations, organisation and time-entities. COLIEE (Competition on Legal Information Extraction/Entailment) tasks (Kano et al., 2019, 2017) have published legal data in Japanese, along with their English translation. The competition has two sub-tasks, a legal information retrieval task and an entailment identification task between law articles and queries. Multiple datasets in Chinese have been released for different tasks, namely Reading Comprehension (Duan et al., 2019), Similar Case Matching (Xiao et al., 2019), Question Answering (Zhong et al., 2020b). Duan et al. (2019) proposed Chinese judicial reading comprehension (CJRC) dataset with about 10K documents and almost 50K questions with answers. Zhong et al. (2020b) presented JEC-QA, a legal question answering dataset collected from the National Judicial Examination of China with about 26K multiple-choice questions. They augment the dataset with a database containing the legal knowledge required to answer the questions and also assign meta information to each of the questions for in-depth analysis. Xiao et al. (2019) proposed CAIL2019-SCM, a dataset containing 8,964 triplets of the case document, with the objective to identify which two cases are more similar in the triplets. Similar case matching has a crucial application as it helps to identify comparable historical cases. A historical case with similar facts often serves as a legal precedent and influences the judgement. Such historical information can be used to make the legal judgement prediction models more robust.

Kleinberg et al. (2017) proposed bail decision prediction as a good proxy to gauge if machine learning can improve human decision making. A large number of bail documents along with the binary decision (granted or denied) makes it an ideal task for automation. In this paper, we also propose the bail prediction task using the HLDC corpus.

3 Hindi Legal Documents Corpus

Hindi Legal Documents Corpus (HLDC) is a corpus of 912,568 Indian legal case documents in the Hindi language. The corpus is created by downloading data from the e-Courts website (a publicly available website: <https://districts.ecourts.gov.in/>). All the legal documents we consider are in the public domain. We download case documents pertaining to the district courts located in the Indian northern state of Uttar Pradesh (U.P.). We focus mainly on the state of U.P. as it is the most populous state of India, resulting in the filing of a large number of cases in district courts. U.P. has 71 districts and about 161 district courts. U.P. is a predominantly Hindi speaking state, and consequently, the official language used in district courts is Hindi. We crawled case documents from all districts of U.P. corresponding to cases filed over two years, from May 01, 2019 to May 01, 2021. Figure 2 shows the map of U.P. and district wise variation in the number of cases. As can be seen in the plot, the western side of the state has more cases; this is possibly due to the high population and more urbanization in the western part. Table 1 shows %-wise division of different case types in HLDC. As evident from the table, majority of documents pertain to bail applications. HLDC corpus has a total of 3,797,817 unique tokens, and on average, each document has 764 tokens.

HLDC Creation Pipeline: We outline the entire pipeline used to create the corpus in Figure 1. The documents on the website are originally typed in Hindi (in Devanagari script) and then scanned to PDF format and uploaded. The first step in HLDC creation is the downloading of documents from the e-Courts website. We downloaded a total of 1,221,950 documents. To extract Hindi text from these, we perform OCR (Optical Character Recognition) via the Tesseract tool¹. Tesseract worked well for our use case as the majority of case documents were well-typed, and it out-

¹<https://github.com/tesseract-ocr>

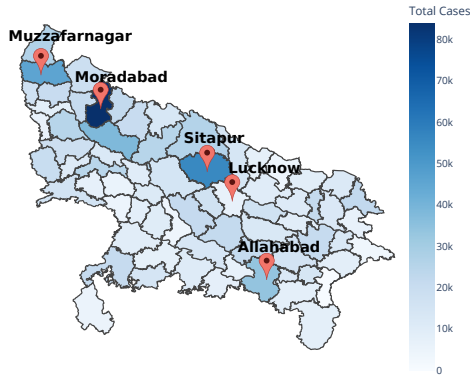


Figure 2: Variation in number of case documents per district in the state of U.P. Prominent districts are marked.

performed other OCR libraries². The obtained text documents were further cleaned to remove noisy documents, e.g. too short (< 32 bytes) or too long (> 8096 bytes) documents, duplicates, and English documents (details in Appendix B). This resulted in a total of 912,568 documents in HLDC. We anonymized the corpus with respect to names and locations. We used a gazetteer³ along with regex-based rules for NER to anonymize the data. List of first names, last names, middle names, locations, titles like पंडित (Pandit: title of Priest), सरजी (Sir: Sir), month names and day names were normalized to <नाम> (Naam: <name>). The gazetteer also had some common ambiguous words (these words can be names or sometimes verbs) like प्रार्थना (Prathna: Can refer to prayer, the action of request or name), गया (Gaya: can refer to location name or verb), किया (Kiya: can refer to infinitive ‘to do’ or name), लिया (Liya: can refer to infinitive ‘to take’ or name). These were removed. Further, we ran RNN-based Hindi NER model⁴ on a subset of documents to find additional entities and these were subsequently used to augment our gazetteer (details Appendix C). Phone numbers were detected using regex patterns and replaced with a <फ़ोन-नंबर> (<phone-number>) tag, numbers written in both English and Hindi were considered.

Legal documents, particularly in lower courts, are highly unstructured and lack standardization with respect to format and sometimes even the terms used. We converted the unstructured doc-

²<https://github.com/JaidedAI/EasyOCR>

³<https://github.com/piyusharma95/NER-for-Hindi>, <https://github.com/balasahebqulave/Dataset-Indian-Names>

⁴<https://github.com/flairNLP/flair>

Case Type	% in HLDC
Bail Applications	31.71
Criminal Cases	20.41
Original Suits	6.54
Warrant or Summons in Criminal Cases	5.24
Complaint Cases	4.37
Civil Misc	3.4
Final Report	3.32
Civil Cases	3.23
Others (Matrimonial Cases, Session Trial, Motor Vehicle Act, etc.)	21.75

Table 1: Case types in HLDC. Out of around 300 different case types, we only show the prominent ones. Majority of the case documents correspond to bail applications.

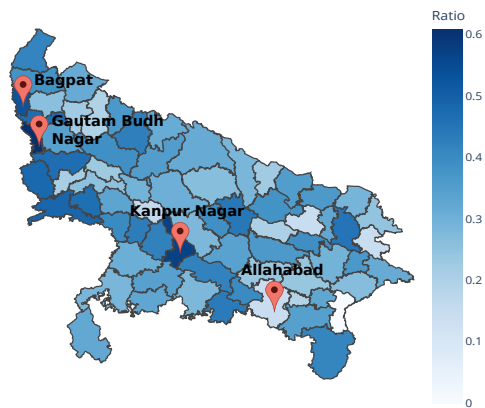


Figure 3: Ratio of number of bail applications to total number of applications in U.P.

uments to semi-structured documents. We segmented each document into a *header* and a *body*. The header contains the meta-information related to the case, for example, case number, court identifier, and applicable sections of the law. The body contains the facts of the case, arguments, judge’s summary, case decision and other information related to the final decision. The documents were segmented using regex and rule based approaches as described in Appendix D.

Case Type Identification: HLDC documents were processed to obtain different case types (e.g., Bail applications, Criminal Cases). The case type was identified via the meta-data that comes with each document. However, different districts use a variation of the same case type name (e.g., Bail Application vs Bail App.). We resolved these standardization issues via manual inspection and regex-based patterns, resulting in a final list of 300 unique case types.

Lexical Analysis: Although Hindi is the official language, U.P. being a large and populous

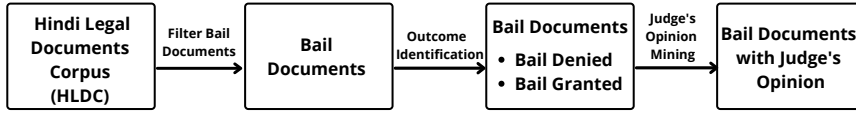


Figure 4: Bail Corpus Creation Pipeline

state, has different dialects of Hindi spoken across the state. We found evidence of this even in official legal documents. For example, the word साकिन (Sakin: motionless) appears 11,614 times in the dataset, 63.8% occurrences of the word come from 6 districts of East U.P. (Ballia, Azamgarh, Maharajganj, Deoria, Siddharthnagar and Kushinagar). This particular variant of motionless being used most often only in East U.P. Similarly, the word गौवंशीय (Gaushiyā: cow and related animals) is mostly used in North-Western UP (Rampur, Pilibhit, Jyotiba Phule Nagar (Amroha), Bijnor, Budaun, Bareilly, Moradabad). Three districts - Muzaffarnagar, Kanshiramnagar and Pratapgarh district constitute 81.5% occurrences of the word दंड (Dand: punishment). These districts are, however, spread across UP. An important thing to note is that words corresponding to specific districts/areas are colloquial and not part of the standard Hindi lexicon. This makes it difficult for prediction model to generalize across districts (§7).

Corpus of Bail Documents: Bail is the provisional release of a suspect in any criminal offence on payment of a bail bond and/or additional restrictions. Bail cases form a large majority of cases in the lower courts, as seen in Table 1. Additionally, they are very time-sensitive as they require quick decisions. For HLDC, the ratio of bail documents to total cases in each district is shown in Figure 3. As a use-case for the corpus, we further investigated the subset of the corpus having only the bail application documents (henceforth, we call it Bail Corpus).

Bail Document Segmentation: For the bail documents, besides the header and body, we further segmented the body part into more subsections (Figure 4). Body is further segmented into **Facts and Arguments**, **Judge’s summary** and **Case Result**. Facts contain the facts of the case and the defendant and prosecutor’s arguments. Most of the bail documents have a concluding paragraph where the judge summarizes their viewpoints of the case, and this constitutes the judge’s summary sub-section. The case result sub-section contains the final decision given by the

judge. More details about document segmentation are in Appendix D.

Bail Decision Extraction: Decision was extracted from Case Result Section using a rule based approach (Details in Appendix E).

Bail Amount Extraction: If bail was granted, it usually has some bail amount associated with it. We extracted this bail amount using regex patterns (Details in Appendix F).

We verified each step of the corpus creation pipeline (Detailed analysis in Appendix G) to ensure the quality of the data. We initially started with 363,003 bail documents across all the 71 districts of U.P., and after removing documents having segmentation errors, we have a Bail corpus with 176,849 bail documents. The bail corpus has a total of 2,342,073 unique tokens, and on average, each document has 614 tokens. A sample document segmented into various sections is shown in Appendix I.

4 HLDC: Ethical Aspects

We create HLDC to promote research and automation in the legal domain dealing with under-researched and low-resource languages like Hindi. The documents that are part of HLDC are in the public domain and hence accessible to all. Given the volume of pending cases in the lower courts, our efforts are aimed towards improving the legal system, which in turn would be beneficial for millions of people. Our work is in line with some of the previous work on legal NLP, e.g., legal corpora creation and legal judgement prediction (section 2). Nevertheless, we are aware that if not handled correctly, legal AI systems developed on legal corpora can negatively impact an individual and society at large. Consequently, we took all possible steps to remove any personal information and biases in the corpus. We anonymized the corpus (section 3) with respect to names, gender information, titles, locations, times, judge’s name, petitioners and appellant’s name. As observed in previous work (Malik et al., 2021b), anonymization of a judge’s name is important as there is a correlation between a case outcome and a judge

name. Along with the HLDC, we also introduce the task of Bail Prediction. Bail applications constitute the bulk of the cases (§3), augmentation by an AI system can help in this case. The bail prediction task aims not to promote the development of systems that replace humans but rather the development of systems that augment humans. The bail prediction task provides only the facts of the case to predict the final decision and avoids any biases that may affect the final decision. Moreover, the Bail corpus and corresponding bail prediction systems can promote the development of explainable systems (Malik et al., 2021b), we leave research on such explainable systems for future work. The legal domain is a relatively new area in NLP research, and more research and investigations are required in this area, especially concerning biases and societal impacts; for this to happen, there is a need for corpora, and in this paper, we make initial steps towards these goals.

5 Bail Prediction Task

To demonstrate a possible applicability for HLDC, we propose the *Bail Prediction Task*, where given the facts of the case, the goal is to predict whether the bail would be granted or denied. Formally, consider a corpus of bail documents $\mathcal{D} = b_1, b_2, \dots, b_i$, where each bail document is segmented as $b_i = (h_i, f_i, j_i, y_i)$. Here, h_i, f_i, j_i and y_i represent the header, facts, judge’s summary and bail decision of the document respectively. Additionally, the facts of every document contain k sentences, more formally, $f_i = (s_i^1, s_i^2, \dots, s_i^k)$, where s_i^k represents the k^{th} sentence of the i^{th} bail document. We formulate the bail prediction task as a binary classification problem. We are interested in modelling $p_\theta(y_i|f_i)$, which is the probability of the outcome y_i given the facts of a case f_i . Here, $y_i \in \{0, 1\}$, i.e., 0 if bail is denied or 1 if bail is granted.

6 Bail Prediction Models

We initially experimented with off-the-shelf pre-trained models trained on general-purpose texts. However, as outlined earlier (§1), the legal domain comes with its own challenges, viz. specialized legal lexicon, long documents, unstructured and noisy texts. Moreover, our corpus is from an under-resourced language (Hindi). Nevertheless, we experimented with existing fine-tuned (pre-trained) models and finally propose a multi-

task model for the bail prediction task.

6.1 Embedding Based Models

We experimented with classical embedding based model Doc2Vec (Le and Mikolov, 2014) and transformer-based contextualized embeddings model IndicBERT (Kakwani et al., 2020). Doc2Vec embeddings, in our case, is trained on the train set of our corpus. The embeddings go as input to SVM and XgBoost classifiers. IndicBERT is a transformer language model trained on 12 major Indian languages. However, IndicBERT, akin to other transformer LMs, has a limitation on the input’s length (number of tokens). Inspired by Malik et al. (2021b); Chalkidis et al. (2019), we experimented with fine-tuning IndicBERT in two settings: the first 512 tokens and the last 512 tokens of the document. The fine-tuned transformer with a classification head is used for bail prediction.

6.2 Summarization Based Models

Given the long lengths of the documents, we experimented with prediction models that use summarization as an intermediate step. In particular, an extractive summary of a document goes as input to a fine-tuned transformer-based classifier (IndicBERT). Besides reducing the length of the document, extractive summarization helps to evaluate the salient sentences in a legal document and is a step towards developing explainable models. We experimented with both unsupervised and supervised extractive summarization models.

For unsupervised approaches we experimented with TF-IDF (Ramos, 2003) and TextRank (a graph based method for extracting most important sentences) (Mihalcea and Tarau, 2004). For the supervised approach, inspired by Bajaj et al. (2021), we propose the use of sentence salience classifier to extract important sentences from the document. Each document ($b_i = (h_i, f_i, j_i, y_i)$, §5) comes with a judge’s summary j_i . For each sentence in the facts of the document (f_i) we calculate it’s cosine similarity with judge’s summary (j_i). Formally, salience of k^{th} sentence s_i^k is given by: $salience(s_i^k) = \cos(h_{j_i}, h_{s_i^k})$. Here h_{j_i} is contextualized distributed representation for j_i obtained using multilingual sentence encoder (Reimers and Gurevych, 2020). Similarly, $h_{s_i^k}$ is the representation for the sentence s_i^k . The cosine similarities provides ranked list of sentences and we select top

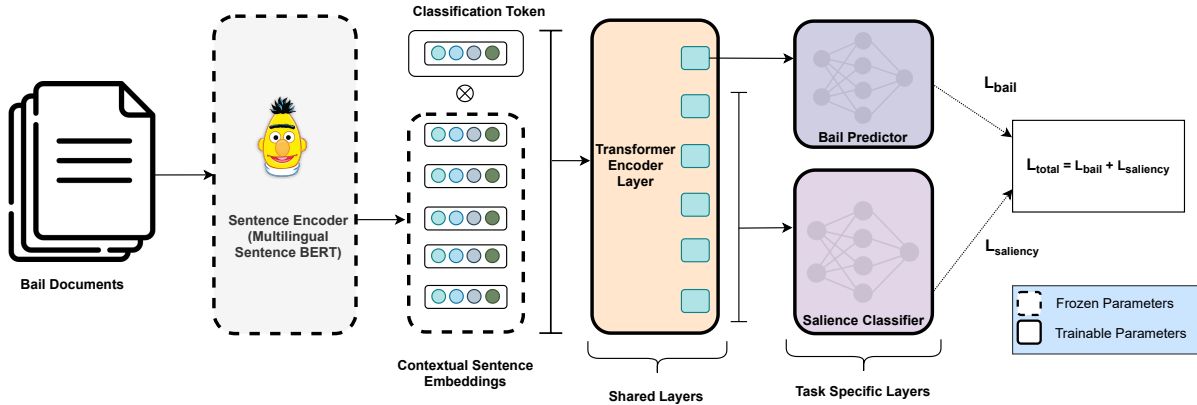


Figure 5: Overview of our multi-task learning approach.

50% sentences as salient. The salient sentences are used to train (and fine-tune) IndicBERT based classifier.

6.3 Multi-Task Learning (MTL) Model

As observed during experiments, summarization based models show improvement in results (§7). Inspired by this, we propose a multi-task framework (Figure 5), where bail prediction is the main task, and sentence salience classification is the auxiliary task. The intuition is that predicting the important sentences via the auxiliary task would force the model to perform better predictions and vice-versa. Input to the model are sentences corresponding to the facts of a case: $s_i^1, s_i^2, \dots, s_i^k$. A multilingual sentence encoder (Reimers and Gurevych, 2020) is used to get contextualized representation of each sentence: $\{h_i^1, h_i^2, \dots, h_i^k\}$. In addition, we append the sentence representations with a special randomly initialized CLS embedding (Devlin et al., 2019) that gets updated during model training. The CLS and sentence embeddings are fed into standard single layer transformer architecture (shared transformer).

6.3.1 Bail Prediction Task

A classification head (fully connected layer MLP) on the top of transformer CLS embedding is used to perform bail prediction. We use standard cross-entropy loss (L_{bail}) for training.

6.3.2 Saliency Classification Task

We use the saliency prediction head (MLP) on top of sentence representations at the output of the shared transformer. For training the auxiliary task, we use sentence saliency scores obtained via cosine similarity (these come from supervised summarization based model). For each sentence, we

		Granted	Dismissed	Total
All Districts	Train	77010 (62%)	46732 (38%)	123742
	Test	21977 (62%)	13423 (38%)	35400
	Validation	11067 (63%)	6640 (37%)	17707
District Wise	Train (44 districts)	77220 (62%)	47121 (38%)	124341
	Validation (10 districts)	9563 (60%)	6366 (40%)	15929
	Test (17 districts)	23271 (64%)	13308 (36%)	36579

Table 2: Number of documents across each split

use binary-cross entropy loss ($L_{saliency}$) to predict the saliency.

Based on our empirical investigations, both the losses are equally weighted, and total loss is given by $L = L_{bail} + L_{saliency}$

7 Experiments and Results

7.1 Dataset Splits

We evaluate the models in two settings: all-district performance and district-wise performance. For the first setting, the model is trained and tested on the documents coming from all districts. The train, validation and test split is 70:10:20. The district-wise setting is to test the generalization capabilities of the model. In this setting, the documents from 44 districts (randomly chosen) are used for training. Testing is done on a different set of 17 districts not present in train set. The validation set has another set of 10 districts. This split corresponds to a 70:10:20 ratio. Table 2 provides the number of documents across splits. The corpus is unbalanced for the prediction class with about 60:40 ratio for positive to negative class (Table 2). All models are evaluated using standard accuracy and F1-score metric (Appendix H.1).

Implementation Details: All models are trained using GeForce RTX 2080Ti GPUs. Models are

Model Name	District-wise		All Districts	
	Acc.	F1	Acc.	F1
Doc2Vec + SVM	0.72	0.69	0.79	0.77
Doc2Vec + XGBoost	0.68	0.59	0.67	0.57
IndicBert-First 512	0.65	0.62	0.73	0.71
IndicBert-Last 512	0.62	0.60	0.78	0.76
TF-IDF+IndicBert	0.76	0.74	0.82	0.81
TextRank+IndicBert	0.76	0.74	0.82	0.81
Saliency Pred.+IndicBert	0.76	0.74	0.80	0.78
Multi-Task	0.78	0.77	0.80	0.78

Table 3: Model results. For TF-IDF and TextRank models we take the sum of the token embeddings.

tuned for hyper-parameters using the validation set (details in Appendix H.2).

7.2 Results

The results are shown in Table 3. As can be observed, in general, the performance of models is lower in the case of district-wise settings. This is possibly due to the lexical variation (section 3) across districts, which makes it difficult for the model to generalize. Moreover, this lexical variation corresponds to the usage of words corresponding to dialects of Hindi. Another thing to note from the results is that, in general, summarization based models perform better than Doc2Vec and transformer-based models, highlighting the importance of the summarization step in the bail prediction task. The proposed end-to-end multi-task model outperforms all the baselines in the district-wise setting with 78.53% accuracy. The auxiliary task of sentence saliency classification helps learn robust features during training and adds a regularization effect on the main task of bail prediction, leading to improved performance than the two-step baselines. However, in the case of an all-district split, the MTL model fails to beat simpler baselines like TF-IDF+IndicBERT. We hypothesize that this is due to the fact that the sentence saliency training data may not be entirely correct since it is based on the cosine similarity heuristic, which may induce some noise for the auxiliary task. Additionally, there is lexical diversity present across documents from different districts. Since documents of all districts are combined in this setting, this may introduce diverse sentences, which are harder to encode for the saliency classifier, while TF-IDF is able to look at the distribution of words across all documents and districts to extract salient sentences.

7.3 Error Analysis

We did further analysis of the model outputs to understand failure points and figure out improvements to the bail prediction system. After examining the miss-classified examples, we observed the following. First, the lack of standardization can manifest in unique ways. In one of the documents, we observed that all the facts and arguments seemed to point to the decision of bail granted. Our model also gauged this correctly and predicted bail granted. However, the actual result of the document showed that even though initially bail was granted because the accused failed to show up on multiple occasions, the judge overturned the decision and the final verdict was bail denied. In some instances, we also observed that even if the facts of the cases are similar the judgments can differ. We observed two cases about the illegal possession of drugs that differed only a bit in the quantity seized but had different decisions. The model is trained only on the documents and has no access to legal knowledge, hence is not able to capture such legal nuances. We also performed quantitative analysis on the model output to better understand the performance. Our model outputs a probabilistic score in the range $\{0, 1\}$. A score closer to 0 indicates our model is confident that bail would be denied, while a score closer to 1 means bail granted. In Figure 6 we plot the ROC curve to showcase the capability of the model at different classification thresholds. ROC plots True Positive and False Positive rates at different thresholds. The area under the ROC curve (AUC) is a measure of aggregated classification performance. Our proposed model has an AUC score of 0.85, indicating a high-classification accuracy for a challenging problem.

We also plot (Figure 7) the density functions corresponding to True Positive (Bail correctly granted), True Negative (Bail correctly dismissed), False Positive (Bail incorrectly granted) and False Negatives (Bail incorrectly dismissed). We observe the correct bail granted predictions are shifted towards 1, and the correct bail denied predictions are shifted towards 0. Additionally, the incorrect samples are concentrated near the middle (≈ 0.5), which shows that our model was able to identify these as borderline cases.

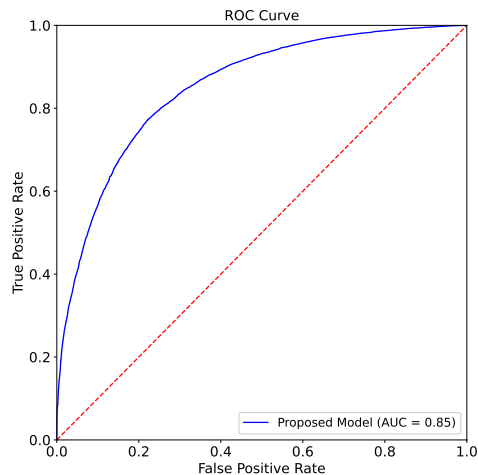


Figure 6: ROC curve for the proposed model. The total AUC (Area under curve) is 0.85.

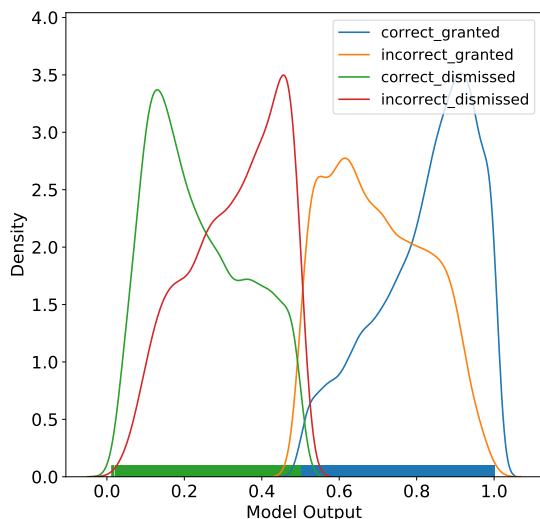


Figure 7: Kernel Density Estimate (KDE) plots of our proposed bail prediction model. The majority of errors (incorrectly dismissed / granted) are borderline cases with model output score around 0.5.

8 Future Work and Conclusion

In this paper, we introduced a large corpus of legal documents for the under-resourced language Hindi: Hindi Legal Documents Corpus (HLDC). We semi-structure the documents to make them amenable for further use in downstream applications. As a use-case for HLDC, we introduce the task of Bail Prediction. We experimented with several models and proposed a multi-task learning based model that predicts salient sentences as an auxiliary task and bail prediction as the main

task. Results show scope for improvement that we plan to explore in future. We also plan to expand HLDC by covering other Indian Hindi speaking states. Furthermore, as a future direction, we plan to collect legal documents in other Indian languages. India has 22 official languages, but for the majority of languages, there are no legal corpora. Another interesting future direction that we would like to explore is the development of deep models infused with legal knowledge so that model is able to capture legal nuances. We plan to use the HLDC corpus for other legal tasks such as summarization and prior case retrieval.

9 Acknowledgements

This paper is dedicated to T.H. Arjun, who contributed towards making this research possible, you will be remembered! We would like to thank Prof. Angshuman Hazarika and Prof. Shouvik Kumar Guha for their valuable suggestions and for guiding us regarding the technical aspects of the Indian legal system. The author Ashutosh Modi would like to acknowledge the support of Google Research India via the Faculty Research Award Grant 2021. This project was partially supported by iHub at IIIT Hyderabad, project O2-001.

References

- Ahsaas Bajaj, Pavitra Dangati, Kalpesh Krishna, Pradhiksha Ashok Kumar, Rheeya Uppaal, Bradford Windsor, Eliot Brenner, Dominic Dotterer, Rajarshi Das, and Andrew McCallum. 2021. [Long document summarization in a low resource setting using pretrained language models](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop*, pages 71–80, Online. Association for Computational Linguistics.
- Paheli Bhattacharya, Kaustubh Hiware, Subham Rajgaria, Nilay Pochhi, Kripabandhu Ghosh, and Saptarshi Ghosh. 2019. [A Comparative Study of Summarization Algorithms Applied to Legal Case Judgments](#). In Leif Azzopardi, Benno Stein, Norbert Fuhr, Philipp Mayr, Claudia Hauff, and Djoerd Hiemstra, editors, *Advances in Information Retrieval*, volume 11437, pages 413–428. Springer International Publishing, Cham.
- Ilias Chalkidis, Ion Androutsopoulos, and Nikolaos Aletras. 2019. [Neural legal judgment prediction in English](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4317–4323, Florence, Italy. Association for Computational Linguistics.

- Ilias Chalkidis, Manos Fergadiotis, and Ion Androutsopoulos. 2021. [MultiEURLEX - a multi-lingual and multi-label legal document classification dataset for zero-shot cross-lingual transfer](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6974–6996, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Xingyi Duan, Baoxin Wang, Ziyue Wang, Wentao Ma, Yiming Cui, Dayong Wu, Shijin Wang, Ting Liu, Tianxiang Huo, Zhen Hu, Heng Wang, and Zhiyuan Liu. 2019. [CJRC: A Reliable Human-Annotated Benchmark DataSet for Chinese Judicial Reading Comprehension](#). In Maosong Sun, Xuanjing Huang, Heng Ji, Zhiyuan Liu, and Yang Liu, editors, *Chinese Computational Linguistics*, volume 11856, pages 439–451. Springer International Publishing, Cham.
- Filippo Galgani, Paul Compton, and Achim Hoffmann. 2012. [Towards automatic generation of catchphrases for legal case reports](#). In *Proceedings of the 13th International Conference on Computational Linguistics and Intelligent Text Processing - Volume Part II, CICLing'12*, page 414–425, Berlin, Heidelberg. Springer-Verlag.
- Peter Jackson, Khalid Al-Kofahi, Alex Tyrrell, and Arun Vachher. 2003. [Information extraction from case law and retrieval of prior cases](#). *Artificial Intelligence*, 150(1):239–290. AI and Law.
- Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. [IndicNLPsuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4948–4961, Online. Association for Computational Linguistics.
- Prathamesh Kalamkar, Aman Tiwari, Astha Agarwal, Saurabh Karn, Smita Gupta, Vivek Raghavan, and Ashutosh Modi. 2022. [Corpus for automatic structuring of legal documents](#). *CoRR*, abs/2201.13125.
- Yoshinobu Kano, Mi-Young Kim, Randy Goebel, and Ken Satoh. 2017. Overview of coliee 2017. In *COLIEE@ICAIL*.
- Yoshinobu Kano, Mi-Young Kim, Masaharu Yoshioaka, Yao Lu, Juliano Rabelo, Naoki Kiyota, Randy Goebel, and Ken Satoh. 2019. [COLIEE-2018: Evaluation of the Competition on Legal Information Extraction and Entailment](#). In Kazuhiro Kojima, Maki Sakamoto, Koji Mineshima, and Ken Satoh, editors, *New Frontiers in Artificial Intelligence*, volume 11717, pages 177–192. Springer International Publishing, Cham.
- Justice Markandey Katju. 2019. Backlog of cases crippling judiciary. <https://tinyurl.com/v4xu6mvk>.
- Mi-Young Kim and Randy Goebel. 2017. [Two-step cascaded textual entailment for legal bar exam question answering](#). In *Proceedings of the 16th edition of the International Conference on Artificial Intelligence and Law*, pages 283–290, London United Kingdom. ACM.
- Jon Kleinberg, Himabindu Lakkaraju, Jure Leskovec, Jens Ludwig, and Sendhil Mullainathan. 2017. [Human Decisions and Machine Predictions*](#). *The Quarterly Journal of Economics*.
- Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32, ICML'14*, page II–1188–II–1196. JMLR.org.
- Pedro H. Luz de Araujo, Teófilo E. de Campos, Renato R. de Oliveira, Matheus Stauffer, Samuel Couto, and Paulo Bermejo. 2018. [LeNER-Br: a dataset for named entity recognition in Brazilian legal text](#). In *International Conference on the Computational Processing of Portuguese (PROPOR)*, Lecture Notes on Computer Science (LNCS), pages 313–323, Canela, RS, Brazil. Springer.
- Vijit Malik, Rishabh Sanjay, Shouvik Kumar Guha, Shubham Kumar Nigam, Anghshuman Hazarika, Arnab Bhattacharya, and Ashutosh Modi. 2021a. [Semantic segmentation of legal documents via rhetorical roles](#). *CoRR*, abs/2112.01836.
- Vijit Malik, Rishabh Sanjay, Shubham Kumar Nigam, Kripabandhu Ghosh, Shouvik Kumar Guha, Arnab Bhattacharya, and Ashutosh Modi. 2021b. [ILDC for CJPE: Indian legal documents corpus for court judgment prediction and explanation](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4046–4062, Online. Association for Computational Linguistics.
- Rada Mihalcea and Paul Tarau. 2004. Texttrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411.
- National Judicial Data Grid. 2021. National judicial data grid statistics. <https://www.njdg.ecourts.gov.in/njdgnew/index.php>.

- PopulationU. 2021. Population of uttar pradesh. <https://www.populationu.com/in/uttar-pradesh-population>.
- Juan Enrique Ramos. 2003. Using tf-idf to determine word relevance in document queries.
- Nils Reimers and Iryna Gurevych. 2020. **Making monolingual sentence embeddings multilingual using knowledge distillation**. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online. Association for Computational Linguistics.
- Yunqiu Shao, Jiabin Mao, Yiqun Liu, Weizhi Ma, Ken Satoh, Min Zhang, and Shaoping Ma. 2020. **BERT-PLI: Modeling Paragraph-Level Interactions for Legal Case Retrieval**. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pages 3501–3507, Yokohama, Japan. International Joint Conferences on Artificial Intelligence Organization.
- Benjamin Strickson and Beatriz De La Iglesia. 2020. **Legal Judgement Prediction for UK Courts**. In *Proceedings of the 2020 The 3rd International Conference on Information Science and System*, pages 204–209, Cambridge United Kingdom. ACM.
- Vu Tran, Minh Le Nguyen, and Ken Satoh. 2019. **Building legal case retrieval systems with lexical matching and summarization using a pre-trained phrase scoring model**. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law, ICAIL '19*, page 275–282, New York, NY, USA. Association for Computing Machinery.
- Wikipedia contributors. 2021. **Devanagari — Wikipedia, the free encyclopedia**. [Online; accessed 10-November-2021].
- WorldData. 2021. World data info: Hindi. <https://www.worlddata.info/languages/hindi.php>.
- Chaojun Xiao, Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Zhiyuan Liu, Maosong Sun, Yansong Feng, Xianpei Han, Zhen Hu, Heng Wang, et al. 2018. **Cail2018: A large-scale legal dataset for judgment prediction**. *arXiv preprint arXiv:1807.02478*.
- Chaojun Xiao, Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Zhiyuan Liu, Maosong Sun, Tianyang Zhang, Xianpei Han, Zhen Hu, Heng Wang, and Jianfeng Xu. 2019. **Cail2019-scm: A dataset of similar case matching in legal domain**. *ArXiv*, abs/1911.08962.
- Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. 2020a. **How does NLP benefit legal system: A summary of legal artificial intelligence**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5218–5230, Online. Association for Computational Linguistics.
- Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. 2020b. **Jecqa: A legal-domain question answering dataset**. In *Proceedings of AAAI*.

Appendix

A Data Statistics

District	Number of Bail Applications
Muzaffarnagar	17234
Moradabad	16219
Budaun	14533
Sitapur	14478
Saharanpur	10838

Table 4: Top 5 districts with most number of bail applications in UP.

B Data Cleaning and Filtering

1,221,950 documents were scraped from Ecourts website and 309,382 documents were removed in the cleaning and filtering process. Following rules were used to remove documents.

- Removed blank documents (whose length is less than 32 bytes)
- Removed duplicate documents
- Removed too long and too short documents (>8096 bytes or <2048 bytes).
- Removed document where majority text was in English language.

This resulted in 912,568 filtered case documents that constitute the Hindi Legal Document Corpus.

C NER Removal

For removing names and locations, lookup was done in lists containing NER. Libraries like HindiNLP⁵ (which uses SequenceTagger from flair library⁶ which is based on an RNN model) were run on a subset of the data to find additional NER that were added to the lists. Since the Sequence-Tagger model is quite slow in processing documents, directly tagging large HLDC is not efficient. If a word was found in one of these lists then it was replaced with a <नाम> (<name>) tag. Phone numbers were replaced with <फ़ोन - नंबर> (<phone-number>) tag using the following regex

```
((\+*)(0[-]*|((91 )*)))(\d{12})+|(\d{10})+))\d{5}([- ]*)\d{6}
```

Phone numbers written in Hindi were also considered by using the same regex as above with English digits replaced with Hindi ones.

⁵<https://github.com/avinsit123/HindiNLP>

⁶<https://github.com/flairNLP/flair>

D Document Segmentation

Out of 912,568 documents in HLDC, 340,280 were bail documents, these were further processed to obtain the Bail Document corpus. Bail documents were structured into different sections. We extracted these sections from the bail documents. Details are mentioned below. An example of document with different sections is shown in Table 10.

D.1 Header

Header refers to the meta data related to the case, for example, धारा (IPC (Indian Penal Code) sections), थाना (police station), case number, date of hearing, accused name, etc. Header is present at the top of the document. Header mostly ended with धारा (IPC) or थाना (police station) details. Hence, in order to cut the document to get header, we first find the indices of धारा (IPC) and थाना (police station), and from these indices we find the finishing word of the header. We then segment the document at the finishing word. We also include the first line of upcoming paragraph in header as it also didn't contain case arguments but contained data like if this is the first bail application or not.

D.2 Case Result

Case Result refers to the end of the document where judge writes their decision. Judge either accepts the bail application or rejects it. If the judge had accepted the bail document then this section mostly also contains bail amount and bail terms for accused.

We observed that result section mostly began along the following line, मामले के समस्त तथ्यों को देखकर (looking at all facts of the case), the keyword तथ्यों (facts) was very common around the start of the result section. Hence, we iterated over the indices of keyword तथ्यों (facts) in reverse order and checked if the division at that index is correct. To check if the division is correct we look for bail result in lower half of the division, if the bail result is present, we classify that division as correct else we move to next index of तथ्यों (facts).

D.3 Body

The remaining portion of the document after removing header and result section was called body. Body section was further divided, as described below.

D.3.1 Judge’s summary

Most of the bail documents have a concluding paragraph where the judge summarizes their viewpoints of the case. To extract this, we first constructed certain regex which often precedes judge’s summary, defendant’s and prosecutor’s arguments (described in Table 5). Since the document might have intermingling of different arguments and opinions, we opted for sentence level annotation of these labels using the regex pattern. The sentences not matching any criteria are given a tag of None. Next we try to replace the None by extending the tags of the sentences to paragraph level as long as no other tag is encountered. As the judge’s opinion mostly occurs at the end, we start iterating from end and start marking the None as judge’s opinion. If a label which is neither None nor judge’s opinion is encountered, the document is discarded as we cannot extract the judge’s opinion from the document using the process defined. If the judge’s opinion label is found in reverse iteration, then we claim that judge’s opinion can be extracted. Finally, all sentences labelled as judge’s opinion either during reverse iteration or during paragraph level extension are extracted out as judge’s summary and rest of the sentences form facts and opinions for further modelling. Using the above process, following are some cases where the judge’s opinion cannot be extracted:

1. Certain characters were mis-identified in the OCR pipeline and hence do not match the regex.
2. The segmentation of document into header, body and result caused a significant portion of the body and thus judge’s opinion to move to result section.
3. The document was written from judge’s perspective and hence judge’s summary also contains the prosecutor’s and defendant’s arguments.
4. The regex didn’t have 100% coverage.

D.3.2 Facts and Arguments

This section comprised of facts related to case, arguments from defendant and prosecutor. Mostly, this corresponds to the portion of the body after removing judge’s summary.

E Extracting Bail Decision from Result

To extract the bail decision we searched for keywords in result section. Keywords like खारिज

Field	Hindi phrases	English Translations
Judge’s Summary	उभय पक्ष की बहस सुनने, पत्रावली के अवलोकन, केस डायरी में उपलब्ध साक्ष्य के अनुसार, मामले के तथ्यों व परिस्थितियों में पूरी तरह से स्पष्ट है, प्रथम सूचना रिपोर्ट, पुलिस प्रपत्र ...परीशिलन किया	Hearing the arguments of the parties, perusal of the records, as per the evidence available in the case diary, fully clear from the facts and circumstances of the case, First Information Report, Police Forms ...perused
Prosecutor	जमानत का विरोध करते हुये अभियोजन की ओर से तर्क दिया गया है, जमानत प्रार्थनापत्र के विरुद्ध आपत्ति	Opposing the bail, it has been argued on behalf of the prosecution, the objection against the bail application
Defendant	अभियुक्त के विद्वान अधिवक्ता का तर्क है, में झूठा एवं रंजिशन फंसाया गया	The learned counsel for the accused has argued, has been falsely and enmity implicated in this case

Table 5: Phrases used to construct regular expression for extracting judge’s opinion. The list is just an indicative of the various phrases and variants used; the entire list can be found in code

(dismissed) and निरस्त (invalidated) identified rejection of bail application and words like स्वीकार (accepted) identified acceptance of bail application. Table 6 lists all the tokens used for extraction.

F Extracting Bail Amount from Result

In case of granted bail decision, the judge specifies bail amount. We saw that the bail amount mostly comprises of personal bond money and surety money. There can be multiple personal bonds and sureties. The bail amount we extracted refers to the sum of all the personal bond money. Bail amount was present in two forms in result section, numerical and Hindi-text. Numerical bail amount was extracted by regex matching and text bail amount was extracted by creating a mapping for it. Table 8 shows few examples of bail mapping.

G HLDC Pipeline Analysis

We used a validation set (0.1% of data) to evaluate our regex based approaches, the results are in Table 7. Note that metrics used for evaluation are

Field	Tokens
Bail granted tokens	स्वीकार किया जाता (is being accepted) स्वीकार करते हुये (by accepting) स्वीकार किये जाते (accepted) रिहा किए जाने का आदेश दिया जाता (ordered to be released) स्वीकार किये जाने योग्य है (deserves to be accepted) पर्याप्त आधार प्रतीत होता है (sufficient evidence found) पर्याप्त आधार पाता हूँ (I see sufficient evidence) आधार पर्याप्त है (sufficient evidence there) पर्याप्त आधार दर्शाते हैं (we see sufficient evidence) रिहा किये जाने का आदेश दिया जाता (ordered to be released) रिहा किया जाये (should be released) रिहा किया जाए (should be released) मुक्त किया जाता (should be released) रिहा कर दिये जायें (should be released) रिहा किया जाता है (is being released) रिहा किया जाये (should be released) रिहा कर दिया जाये (is being released)
Bail denied tokens	निरस्त किया जाता (is canceled) निरस्त किये जाते (are canceled) निरस्त किए जाते (are canceled) खण्डित किया जाता (is broken) खण्डित किये जाते (are broken) पर्याप्त आधार नहीं है (insufficient evidence) पर्याप्त आधार प्रतीत नहीं होता (insufficient evidence) खारिज किया जाता (is rejected) अस्वीकार (denied)

Table 6: Bail decision tokens

quite strict and hence the results are much lower for Judge’s summary part. The segmentation and Judge’s opinion were strictly evaluated and even a single sentence in the wrong segment reduces the accuracy. We also see that the main binary label of outcome detection (bail granted or denied) had an almost perfect accuracy of 99.4%. Nevertheless, in future we plan to improve our pipeline further by training machine learning models.

Process	Accuracy
Header, Body and Case Result Segmentation	89.7%
Judge’s Opinion and Facts extraction	85.7%
Bail Decision Extraction	99.4%

Table 7: Evaluation results of bail document division and bail decision extraction pipeline.

Text Amount	In Value Form
5-5 हजार	10000
बीस-बीस हजार	40000
पचीस-पचीस हजार	50000
तीस-तीस हजार	60000
...	...

Table 8: Text bail amount mapping example

H Model Details

H.1 Evaluation Metrics

To evaluate the performance of all the models, we use Accuracy, and F1-score, which are considered

Model	Hyper-Parameters (L=Learning Rate), (E=Epochs), (D=Embedding Dimension(Default 200)), (W= Weight Decay), (E=Epochs(Default 15))	
	District-wise Split	All Districts Split
Doc2Vec + SVM	E=100	E=100
Doc2Vec + XGBoost	E=100, D=300	E=100, D=300
IndicBert - (First 512 Tokens)	L=3.69 × 10 ⁻⁶ , W=2.6 × 10 ⁻²	L=1.58 × 10 ⁻⁶ , W=4.8 × 10 ⁻²
IndicBert - (Last 512 Tokens)	L=5.60 × 10 ⁻⁵ , W=1.0 × 10 ⁻²	L=2.18 × 10 ⁻⁵ , W=4.3 × 10 ⁻²
TF-IDF + IndicBert	L=1.11 × 10 ⁻⁵ , W=1.9 × 10 ⁻²	L=9.84 × 10 ⁻⁶ , W=4.9 × 10 ⁻²
TextRank + IndicBert	L=3.17 × 10 ⁻⁶ , W=3.1 × 10 ⁻²	L=3.99 × 10 ⁻⁶ , W=1.5 × 10 ⁻²
Saliency Pred. + IndicBert	L=1 × 10 ⁻⁵ , W=3.2 × 10 ⁻²	L=4.2 × 10 ⁻⁶ , W=1.7 × 10 ⁻²
Multi-Task	E=30, L=5 × 10 ⁻⁵	E=30, L=1 × 10 ⁻⁵

Table 9: Listing of Hyper-Parameters for Training of Models

standard evaluation metrics while performing classification experiments. These are mathematically described as the follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$F_1 \text{ Score} = \frac{2 * Precision * Recall}{Precision + Recall}$$

where TP, FP, TN, and FN denote True Positives, False Positives, True Negatives, and False Negatives, respectively. The mathematical formulation for *Precision* and *Recall* is given as follows:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

H.2 Hyperparameter Tuning

We used Optuna⁷ for hyperparameter optimisation. Optuna allows us to easily define search spaces, select optimisation algorithms and scale with easy parallelization. We run parameter tuning on 10% of the data to identify the best parameters before retraining the model with the best parameters on the entire dataset. The best parameters are listed in Table 9.

⁷<https://github.com/optuna/optuna>

I Sample Segmented Document

Field	Example	Translation
<p>Header: This chunk of the document contains meta information related to the case like court hearing date, IPC sections attached, police station of complain, etc.</p>	<p>न्यायालय विशेष न्यायाधीश (एस०सी०, / एस०टी० एक्ट) इलाहाबाद । जमानत प्रार्थना पत्र संख्या-4438/ 2020 C.N.R.No. UPAD01008173-2020 शैलेन्द्र <नाम> बनाम उत्तर प्रदेश राज्य मुकदमा अपराध संख्या-773/ 2020 धारा-376, 377, 504 व 506 भा०दं०सं० व <नाम> 3(2)V एस०सी० / एस०टी० एक्ट थाना-सोरांव, प्रयागराज 04.03.2021: यह जमानत प्रार्थना पत्र प्रार्थी / अभियुक्त <नाम> <नाम> पुत्र <नाम> <नाम> निवासी-धामापुर अब्दालपुर, थाना सोरांव, जनपद प्रयागराज की ओर से अपराध संख्या-773/ 2020, धारा-376, 377, 504 व 506 भारतीय दण्ड <नाम> व धारा-3(2)५ एस०सी०/ एस०टी० एक्ट, थाना-सोरांव, प्रयागराज में <नाम> गया है जो अभियुक्त के पैरोकार उसके पिता <नाम> <नाम> के शपथपत्र से समर्थित है।</p>	<p>Court Special Judge (SC, / ST Act) Allahabad. Bail Petition Number - 4438/2020 C. N. R. No. UPAD01008173-2020 NER Unknown Uttar Pradesh State Lawsuit Offense Number -773/2020 Article 376, 377, 504 & 506 Indian Penal Code & Article 3(2)V SC /ST Act Station - Sorav, Prayagraj. 04.03.2021 This bail petition has been given of behalf of petitioner/accused NER son NER, resident of Dhamapur Abdalpur, Station – Sorav, District – Prayagraj Offense number 773/2020 Article 376, 377, 504 & 506 Indian Penal Code & Article 3(2) 5 SC /ST Act in Station - Sorav, Prayagraj which is endorsed by the affidavit of the pairokar father NER of the accused.</p>
<p>Facts and Arguments: This chunk of the document contains case facts related to the case and arguments from defendant and prosecutor.</p>	<p>संक्षेप में अभियोजन तथ्य इस प्रकार है कि वादिनी का सम्पक <नाम> <नाम> पुत्र <नाम> नाश <नाम> के साथ 06 वर्ष पूर्व हुआ। <नाम> <नाम> वादिनी के घर दूध पहुंचाने आता था। ... वादिनी लोक लाजके <नाम> शान्त रही। वादिनी मुकदमा की तहरीर <नाम> मुकदमा पंजीकृत किया गया। अभियुक्त की ओर से यह तर्क <नाम> गया कि प्रार्थी का कभी कोई भीआपराधिक इतिहास नहीं है। ... सामान्यआरोप है जो झूठ व फर्जी है। ... धन <नाम> <नाम> ब्लेकमेल करने के लिए पीड़िताद्वारा झूठा एवं फर्जी मुकदमा दर्ज कराया गया है। तहरीर में वादिनी के हस्ताक्षर वतारीख का उल्लेख नहीं है। प्रार्थी निर्दोष है और उसे झूठा फँसाया गया है। प्रार्थीदिनांक 07.10.2020 से जेल में निरुद्ध है। प्रार्थी जमानत देने को तैयार है। अतः प्रार्थीको जमानत <नाम> <नाम> किया जाय । अभियोजन पक्ष की ओर से विशेष लोक अभियोजक ने जमानत का विरोधकरते हुए कहा है कि ... अभियुक्त द्वाराकारित अपराध गम्भीर <नाम> का है। अतः अभियुक्त का जमानत आवेदन निरस्तकिया जाय।</p>	<p>In short, prosecution facts are such that, the litigant (female) came in contact with NER son of NER 6 years ago. NER used to come to litigant's (female) house for delivering milk. Paraphrase ...Litigant (female) kept quiet due to public shame. Complaint was registered on basis of case record of litigant (female). It was argued on behalf of the accused that the petitioner has no previous crime record. Paraphrase ... it is a complete accusation which is untrue and fictitious. Paraphrase... With the intention of getting money, in order to blackmail, a false and fictitious complaint was filed by the victim (female). There is no mention of the signature of the litigant (female) and date on record. Petitioner is innocent and is being falsely framed. Petitioner is lodged in jail since 07.10.2020. Petitioner is ready to give bail. Hence petitioner be released on bail. On behalf of the prosecution party, refuting the bail application, special public prosecutor has said that... paraphrase ... the crime committed by the accused is of serious nature. So the bail petition of the accused be overruled.</p>

Continued on next page

Table 10 – continued from previous page

Field	Example	Translation
<p>Judge’s Opinion: This refers to the few lines present in the middle portion of the document where judge writes their opinion of the case.</p>	<p>मैंने अभियुक्त की ओर से उपस्थित विद्वान अधिवक्ता एवं राज्य की ओर से उपस्थित विशेष लोक अभियोजक को सुना एवं समस्त प्रपत्रों का अवलोकन किया। आवेदक / अभियुक्त द्वारा पीड़िता को शादी का झॉसा देकर उसके साथ बलात्कार एवं अप्राकृतिक कृत्य किया गया तथा जातिसूचक गाली देते हुए <नाम> से मारने की धमकी दी। यह अपराध एक महिला और समाज के <नाम> है, जो गम्भीर <नाम> का है।</p>	<p>I heard the learned advocate representing the accused and the special public prosecutor representing the State and examined all documents. The victim (female) was raped and she was subjected to unnatural acts under the pretext of marriage and was subjected to caste-slurred abuse and threatened to be killed by the petitioner/accused. This crime is against a woman and society which is of serious nature.</p>
<p>Result: This chunk of the document contains decision made by judge on the case.</p>	<p>अतः मामले के समस्त तथ्य, परिस्थितियों एवं अपराध की गम्भीरता को देखते हुए प्रार्थी / अभियुक्त को जमानत <नाम> छोड़े जाने का आधार पर्याप्त नहीं पाता हूँ। अभियुक्त का जमानत प्रार्थना पत्र निरस्त किए जाने <नाम> है। आदेश अभियुक्त शैलेन्द्र <नाम> पुत्र <नाम> <नाम> का जमानत प्रार्थना पत्र निरस्त किया जाता है। (रामकेश) विशेष न्यायाधीश (एस0सी0/ एस0टी0 एक्ट) इलाहाबाद JCO Code- UP5902</p>	<p>So, looking at all the facts of the matter, the circumstances and the seriousness of the crime, (I) do not find enough basis for releasing the petitioner/accused on bail. The bail petition of the accused is fit to be overruled. Ruling the bail petition of the accused NER son NER is overruled. (Ramkesh) Special Judge (SCST Act) Allahabad J C Code 5902</p>

Table 10: A sample segmented document