IL-TUR: Benchmark for Legal Text Understanding and Reasoning

Anonymous ACL submission

Abstract

Legal systems worldwide are inundated with exponential growth in cases and documents. To streamline the legal system, there is an imminent need to develop NLP and ML techniques for automatically processing and understanding legal documents. However, evaluating and comparing various NLP models designed specifically for the legal domain is challenging. This paper addresses this challenge by proposing **IL-TUR**: Benchmark for Indian Legal Text Understanding and Reasoning. **IL-TUR** contains monolingual (English, Hindi) and multi-lingual (9 Indian languages) domain-specific tasks that address different aspects of the legal system from the point of view of understanding and reasoning over Indian legal documents. We present baseline models (including LLM-based) for each task, outlining the gap between models and the ground truth. We will release a public leaderboard where the research community can upload and compare legal text understanding systems on various metrics, thus fostering research in the legal domain.

"Justice delayed is justice denied" - Legal Maxim

1 Introduction

Besides several other purposes, legal systems have been established in various countries to ensure, at the very minimum, order and fairness in society and to safeguard fundamental human rights. However, legal systems worldwide struggle with exponentially growing legal cases in various courts. It is even more pronounced in populous countries; for example, in India, there are about 43 million pending cases in multiple courts (National Judicial Data Grid, 2023). Such a massive backlog of cases goes against the fundamental human right of fair access to justice.

Documents in different natural languages are the backbone of various legal processes. Natural Language Processing (NLP) based techniques could be helpful in various legal processes involving fundamental tasks related to information extraction, document understanding, and prediction. In this paper, we introduce **IL-TUR**, a benchmark for *Indian Legal Text Understanding and Reasoning*. The purpose of **IL-TUR** is twofold. First, it aims to foster research in the Legal-NLP (L-NLP) domain and plans to address the pain points associated with processing legal texts (see below); second, it provides a platform for comparing different models and further advancing the L-NLP domain.

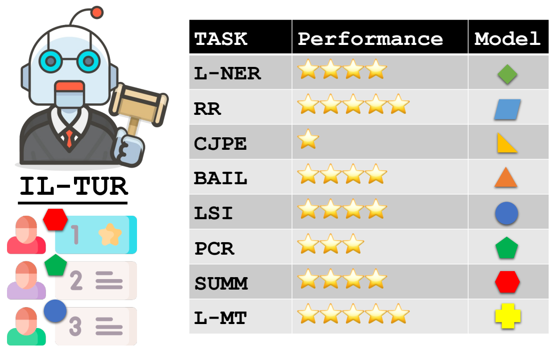

Figure 1: **IL-TUR**: A consolidated benchmark covering a wide range of legal text understanding and reasoning tasks with a publically available leaderboard.

Why a separate benchmark for the legal domain? The legal text involves natural language but differs from the regular text used to train NLP models. 1) Many of the terms used in legal documents are domain-specific. For example, some words used in everyday language have specialized meanings in legal parlance. The presence of a different lexicon posits a need for specialized NLP tools to handle legal texts. 2) Legal documents are typically very long compared to regular texts. For example, the average length of a legal document from the Supreme Court of India (SCI) is 4000 words (Malik et al., 2021). It poses a challenge for existing NLP models (e.g., LLMs) as the information is spread throughout the document and must be linked together for reasoning. Moreover, many of the existing language models (e.g., BERT (Devlin et al., 2019)) have limitations on the length (512 tokens) of the input. It requires developing specialized models for processing and handling long legal documents. 3) Legal documents are highly unstructured and sometimes noisy (for example, in the Indian setting, most documents are typed manually in the courts and prone to grammatical mistakes and typos). The absence of structure in the documents makes extracting semantically relevant information from large chunks of text difficult. 4) The legal domain is further subdivided into specialized subdomains; for example, criminal law differs from civil law, and both differ from banking and insurance law. Even though some fundamental legal principles are shared across various laws, models trained on a particular law (e.g., civil law) may not work on another (e.g., banking and insurance law). Hence, domain adaptation is a challenge. 5) Lastly, many existing state-of-the-art (SOTA) NLP models are black boxes; however, explainability is not a second-class citizen for the legal domain. For models to be widely usable by legal practitioners, these need to be explainable. Due to the above reasons, a separate set of models/systems is required for processing and understanding legal documents. Given the huge backlog of cases, NLP-based technologies could come to our rescue and help streamline the legal workflow. Even a small technical intervention can have a considerable impact. Hence, a benchmark is needed to promote the development of models in this area. In a nutshell, we make the following contributions:

- We introduce **IL-TUR**: a benchmark for Indian Legal Text Understanding and Reasoning. The benchmark has eight tasks (in English and 9 Indian languages) requiring different types of legal knowledge and skills to solve. Moreover, the list of tasks is not exhaustive, and we plan to keep adding more tasks to **IL-TUR**. Currently, there are various L-NLP-specific tasks; however, these occur in isolation, making it difficult to keep track of progress made in the field. Similar to existing NLP benchmarks (e.g., GLUE (Wang et al., 2018a)), we consolidate and harmonize some of the existing L-NLP tasks and create some new tasks to come up with a unified benchmark and platform to compare models.
- We report baseline model results on each of the tasks. We also experiment with various LLMs (§4), and results show that LLMs are far from solving the tasks and hence point towards the need to develop better models.
- We will release the dataset and baseline models associated with each task. Further, we plan to release a leaderboard where anyone can upload their model and test against the baselines and other proposed systems (e.g., Fig. 1).

2 Related Work

Over the past few years, L-NLP has been a fertile area for research. Researchers have explored different aspects of the legal domain via various tasks such as Prior Case Retrieval (Joshi et al., 2023; Jackson et al., 2003a), Case Prediction (Malik et al., 2021; Chalkidis et al., 2019; Strickson and De La Iglesia, 2020; Kapoor et al., 2022), Summarization (Moens et al., 1999), Semantic Segmentation of Legal Documents (Malik et al., 2022; Kalamkar et al., 2022b; Bhattacharya et al., 2019), and Information Extraction and Retrieval (Tran et al., 2019; Lagos et al., 2010). On the modeling side, various techniques have been proposed, ranging from classical ML-based methods such as SVM (Al-Kofahi et al., 2001; Jackson et al., 2003b) to recent transformer-based models (Chalkidis et al., 2019; Malik et al., 2021). Researchers have also proposed legal domain-specific language models such as LegalBERT (Chalkidis et al., 2020), CaseLawBERT (Zheng et al., 2021) and InLegalBERT and InCaseLawBERT (Paul et al., 2023). However, legal LLMs have shown limited success and have not demonstrated generalization and transfer learning capabilities (Chalkidis, 2023; Malik et al., 2021; Joshi et al., 2023).

Comparison with Existing Benchmarks: Benchmarks have played a crucial role in the development of better techniques and models in almost every domain, such as computer vision (Deng et al., 2009; Guo et al., 2014; Wu et al., 2013), NLP, and reinforcement learning (Laskin et al., 2021; Cobbe et al., 2020; Zhang et al., 2018). In particular, in the NLP domain, various benchmarks have been proposed, for example, GLUE (Wang et al., 2018a), Super-GLUE (Wang et al., 2019a), XTREME (Hu et al., 2020), CLUE (Xu et al., 2020), GLGE (Liu et al., 2020), and IndicNLPSuite (Kakwani et al., 2020). However, these benchmarks focus on the general NLP domain, and models developed for the generic domains do not perform well for the legal domain (Malik et al., 2022; Joshi et al., 2023). Similar attempts have thus been made for the Legal domain; for example, Chalkidis et al. (2022a)

Dataset	Jurisdictions	System	Task types	Languages
LexGLUE	U.S., E.U.	Predominantly Civil Law	Classification	English
LEXTREME	E.U., Brazil	Predominantly Civil Law	Classification	E.U.
FAIRLEX	E.U., U.S., China, Switzerland	Predominantly Civil Law	Fairness evaluation on Classification	E.U., Chinese
LBOX	Korea	Civil Law	Classification, Generation	Korean
LEGALBENCH	Multiple	Common & Civil Law	Generation	English
LAWBENCH	China	Civil Law	Classification, Generation, Extraction	Chinese
IL-TUR (ours)	India	Common & Civil Law	Classification, Retrieval, Generation, Extraction	English, Indian

Table 1: Comparison of different L-NLP benchmarks.

have proposed LexGLUE, a specialized English language benchmark (restricted to EU and US legal systems) for evaluating legal NLP models. The authors created the benchmark by consolidating existing datasets for various tasks. LexGLUE introduces six main (all classification-based) tasks, like violated article identification, case issue classification, concept identification, contract topic prediction, unfair contractual terms identification, and case holding identification. Niklaus et al. (2023) have proposed LEXTREME, a multi-lingual (24 EU languages) legal NLP benchmark (all tasks classification-based) restricted to EU and Brazilian jurisdictions. Chalkidis et al. (2022b) have introduced FAIRLEX, a multi-lingual benchmark consisting of cases from 5 languages and 4 jurisdictions, to test the fairness of different models on legal judgment and topic prediction. Hwang et al. (2022) have introduced LBOX benchmark for the Korean legal system. The benchmark targets tasks related to classification and summarization; the documents are in Korean. Recently, Guha et al. (2023) released LegalBench, a large, collaborative legal benchmark (restricted to US legal system) consisting of 162 tasks (in English) to test the reasoning abilities of LLMs. The tasks belong to six different categories of legal reasoning and address various stages in the pipeline of the litigation process. LegalBench is primarily focused on testing the ability of LLMs to handle legal processes at various stages of litigation, consequently, the tasks involve shorter texts (avg. length $\sim$ 200 words). To benchmark LLMs for Chinese law, Fei et al. (2023) released LawBENCH, a benchmark consisting of 20 tasks (in Chinese) to evaluate the capability of LLMs to memorize and understand legal knowledge. Most of these tasks consist of longer texts compared to LegalBench (avg. length $\sim$ 300 words).

IL-TUR differs from the existing benchmarks (see Table 1). First, **IL-TUR** focuses on multiple tasks that are not restricted to classification but also involve information retrieval, generation, and explanation. Second, via **IL-TUR**, we introduce tasks that are grounded in the actual legal workflow and, consequently, are more complex and involve actual long legal documents (average length 4000 words). In contrast to some of the popular benchmarks, **IL-TUR** is not introduced to test the law understanding capability of LLMs but rather to address the problems plaguing the judiciary. In the future, if LLMs are replaced by some other class of machine learning models, **IL-TUR** would still be relevant. In fact, as shown in our experiments, we observe that long legal documents are challenging for LLMs. Third, **IL-TUR** is based on Indian legal documents. Given that India is the most populous country in the world (population of $\sim$ 1.4 billion (United Nations, 2023)) and there is a backlog of almost 43 million cases, it is imminent to develop benchmarks and datasets for the Indian legal system. From the language perspective, **IL-TUR** benchmark covers English and 9 major Indian languages. Although **IL-TUR** is India-specific, the models developed for **IL-TUR** could also be adapted and further developed for the legal systems of other countries. Lastly and most importantly, **IL-TUR** covers tasks related to the common-law system as well as the civil law system. India has a predominantly common-law system, which implies that a judge in a higher court can overrule existing precedents, so the decision may not always be as per the rule book (written statutes and laws). It introduces some subjectivity into the decision-making process and must be backed by solid reasoning, making the tasks in **IL-TUR** much more difficult. Additionally, India also has a civil law system in certain matters (e.g., banking and insurance). In the proposed benchmark, we cover both settings. Moreover, the legal domain has various areas (following common or civil systems) of laws such as criminal, civil, and banking; via the benchmark, we want to test the cross-area generalization capabilities of the models i.e., how well the models developed on data from one area generalize across other areas. In contrast, Korea, China, (and, to a large extent, the EU) mainly follow civil law where a decision is as per the rule book. **IL-TUR** aims to fill the voids in the Legal NLP for the India settings by introducing some of the foundational tasks that can be useful for various legal applications. Table 1 compares different Legal-NLP benchmarks.

3 IL-TUR: Legal-NLP Benchmark

Table 2 summarizes various tasks proposed in **IL-TUR**. The tasks cover multiple aspects of the legal domain and require specialized skills and knowledge to solve them.

3.1 Design Philosophy

We want to develop technology that enables automated semantic and legal understanding of legal documents and processes. We created **IL-TUR** with the following principles in mind.

1) Legal Understanding and World Knowledge: The tasks should cater exclusively to the legal domain. Solving a task should require in-depth knowledge and understanding of the law and its associated areas. Further, the tasks should not be restricted to only classification but should also involve retrieval, generation, and explanation. The proposed tasks address the pain points of processing legal texts (§1). Moreover, solving legal tasks should require knowledge about the law as well as commonsense knowledge about the world (e.g., facts in a particular case). **2) Difficulty Level:** The difficulty level should be such that these are not solvable by a layperson (having minimal knowledge and expertise in legal matters). It ensures that general language learners are not easily able to solve the tasks, and the tasks would be sufficiently challenging for the current state-of-the-art models (e.g., LLMs). **3) Language:** Since India is a multi-lingual society, the tasks should cater to the most frequent languages used in the courts. We cover tasks in English and 9 other Indian languages. **4) Evaluation:** The tasks should be automatically evaluable, and the metrics used should align with human judgments. **5) Public Availability:** The data used for the tasks should be publicly available so anyone can use it for research purposes without licensing or copyright restrictions. Further, a leaderboard should be available to compare different systems and models. We will release the data via a Creative Common Attribution-NonCommercial-ShareAlike (CC BY-NC-SA) license and create a public leaderboard.

3.2 IL-TUR Tasks

Based on the design philosophy, in this version of **IL-TUR**, we selected eight different tasks as described next.

Task	Dataset	Avg. #Words	Task Type	Key Skills Required
L-NER	105 docs (650k words)	6,180	Sequence Classification	Foundational task, legal understanding
RR	21,184 sentences	25,796	Multi-Class Classification	Foundational task, legal knowledge and legal semantics understanding
CJPE	ILDC (34k Docs)	3,336	Classification, Extraction	Legal understanding and reasoning
BAIL	HLDC (900k Docs)	-	Classification	Legal understanding (in Hindi) and reasoning
LSI	ILSI (65k samples)	2,406	Multi-Label Classification	Understanding of the statutes and their applicability in various factual situations, commonsense knowledge and reasoning
PCR	IL-PCR (7,070 Docs)	8,096	Retrieval	Understanding of facts (commonsense + legal knowledge) and statutes, concept of legal relevance
SUMM	In-Abs (7,130 Docs)	4,376	Generation	Legal understanding and generation
L-MT	MILPaC (17,853 text pairs)	49	Generation	Parallel understanding of legal text in English and 9 Indian languages

Table 2: Summary of Tasks introduced in **IL-TUR**.

3.2.1 Legal Named Entity Recognition (L-NER)

Task Motivation and Description: Named Entity Recognition (NER) is a foundational task in NLP (Yadav and Bethard, 2019). However, in the legal domain, the types of named entities one may be interested in differ (e.g., judge, petitioner (appellant), and respondent), which may not be identified by a standard NER system. Hence, a separate task is needed to identify the legal named entities in the documents. Note that L-NER is very different from the standard NER; the standard NER (identifying person/organization/location names) requires a language understanding; in contrast, identifying the roles of entities involved in a legal case (L-NER) requires an understanding of the legal terminologies. Hence, we develop a gold-standard dataset for L-NER annotated with the help of law students (details in App. A). Moreover, the set of legal entities and corresponding definitions are formulated with the help of legal academicians (experts). **Formally, given a legal document, the task of Legal Named Entity Recognition is to identify entities (set of 12 entity types), namely, Appellant, Respondent, Judge, Appellant Counsel, Respondent Counsel, Court, Authority, Witness, Statute, Precedent, Date, and Case Number.** We provide detailed definitions of each named entity in the App. A. App. Fig. 2 shows an example of an L-NER task.

Dataset: We collected a total of 105 case documents in English (a total of 650K words and 12.5K entities) (Dataset and annotation details in App. A).

Task Evaluation: We use standard metrics of *strict* macro-averaged precision, recall, and F1 score for evaluation. The *strict* score assumes a correct match only if *both* the entity boundary and entity type are correctly predicted.

3.2.2 Rhetorical Role Prediction (RR)

Task Motivation and Description: As pointed out earlier, legal documents are typically long (avg. length 4000 words) and highly unstructured, with legal information spread throughout the document. Segmenting the long documents into topically coherent units (such as facts, arguments, precedent, statute, etc.) not only helps highlight the relevant information but also reduces human effort when going through a long list of documents. These topically coherent units are termed as *Rhetorical Roles* (RR). **Given a legal document, the task of RR prediction involves assigning RR label(s) to each sentence.** The definition of each RR label is given in the App. A. App. Fig. 3 shows an excerpt from a legal document annotated with RR labels. RR Prediction is a foundational task that helps structure the information and thus aids downstream applications related to document understanding, information extraction, summarization, and retrieval.

Dataset: For this task, we use the dataset developed by Malik et al. (2022) primarily due to the large number of annotations by several Law academicians and public availability. The dataset consists of $21,184$ sentences from legal documents (in English) about banking and competition law. The sentences are annotated with 13 RRs by as many as six legal experts (from a reputed Indian law school). The 13 RR labels are: *Fact, Issue, Arguments (Respondent), Argument (Petitioner), Statute, Dissent, Precedent Relied Upon, Precedent Not Relied Upon, Precedent Overruled, Ruling By Lower Court, Ratio Of The Decision, Ruling By Present Court, None.* (details in App. A).

Task Evaluation: The task is evaluated using the standard metric of macro-F1 score.

3.2.3 Court Judgment Prediction with Explanation (CJPE)

Task Motivation and Description: The task of Court Judgment Prediction with Explanation (CJPE) aims to augment a judge in the judicial decision-making process by predicting the final outcome of the case. Note that the idea behind this task is *not* to replace human judges but to aid them. Furthermore, the task requires the system to explain its decision so that it is interpretable for a human using it. **Formally, the task of Court Judgment Prediction with Explanation (CJPE) involves predicting the final judgment (appeal accepted or denied, i.e., the binary outcome of 0 or 1) for a given judgment document (having facts and other details) and providing the explanation for the decision.** The explanations, in this case, are in the form of the crucial sentences appearing in the input text that lead to the decision.

Dataset: For the CJPE task, we use the `Indian Legal Document Corpus (ILDC)` (Malik et al., 2021). ILDC is a corpus of 35K legal judgment documents (in English) from the Supreme Court of India. Each document is annotated with the ground truth (actual decision given by the judge); further, a small subset of the documents are annotated with explanations by legal experts (details in App. A). This makes it a suitable dataset to consider for a legal understanding benchmark as it covers both judgement as well as relevant explanations annotated by human experts. Regarding ethical concerns, we follow Malik et al. (2021) who took various steps, such as normalizing the dataset concerning named entities to remove any biases in the data (also check the Ethical Considerations section).

Task Evaluation: The Prediction part of the CJPE task is evaluated using standard accuracy and macro-F1 score metric. The explanation part is evaluated using ROUGE scores (Lin, 2004).

3.2.4 Bail Prediction (BAIL)

Task Motivation and Description: A large fraction of the pending cases in India are from the district-level courts, and have to do with bail applications (`https://en.wikipedia.org/wiki/Bail`) (Kapoor et al., 2022). Many of the district courts in India use Hindi as their official language (also refer to the Limitations section). Given the importance of Hindi (the most frequently spoken/written language in India), the task of Bail Prediction for Hindi legal documents is of immense importance, incorporating both language diversity and wider applicability in the Indian legal system. **Formally, given a legal document (having the facts of the case), the task of Bail Prediction involves predicting if the accused should be granted bail or not (i.e., a binary decision of 0 and 1).**

Dataset: For this task, we use the `Hindi Legal Document Corpus (HLDC)` dataset. (Kapoor et al., 2022). HLDC is a corpus of 900K Hindi legal documents from district courts of a north Indian state. HLDC corpus creation process involves various pre-processing steps to take care of possible ethical consequences (also check the Ethical Considerations section) that may creep in due to different types of biases. Bail documents are annotated with ground truth bail decisions. More details about the dataset creation and ethical considerations are given in App. A.

Task Evaluation: Since the Bail prediction task is essentially a binary prediction task, it is evaluated using the standard macro-F1 score metric.

3.2.5 Legal Statute Identification (LSI)

Task Motivation and Description: One of the first steps in the judicial process is finding the applicable statutes/laws based on the facts of the current situation. Manually rummaging through multiple legislation and laws to find out the relevant statutes can be time-consuming, making the LSI task important for reducing the workload, helping improve the efficiency of the judicial system. **The task of Legal Statute Identification (LSI) is formally defined to automatically identify the relevant statutes given the facts of a case.** An example of the LSI task is presented in the App. Table 9.

Dataset: For LSI, we use the `Indian Legal Statute Identification (ILSI)` dataset (Paul et al., 2022). The dataset consists of fact portions of 65K court case documents (derived from criminal court cases from the Supreme Court of India (SCI) and 6 High Courts of India). The Indian Penal Code (IPC) comprises most criminal statutes and procedures in India; the 100 most frequently occurring statutes in the IPC were chosen as the target statutes. The original ILSI dataset released by Paul et al. (2022) contains named entities. In line with recent works in legal NLP (Malik et al., 2021), we anonymize the dataset by masking entities of types 'PERSON' and 'ORGANIZATION' to remove any possible bias (details in App. A).

Task Evaluation: LSI is formulated as a multi-label text classification task. We use standard classification metrics such as macro-averaged precision, recall, and F1 score for evaluation. In principle, the LSI task can also be considered a retrieval task instead of a multi-label classification task, i.e., the task is to retrieve relevant statutes from a dynamic set of statutes, given the fact (query). However, in the current version of the benchmark, we follow the classification setting proposed in previous works (Wang et al., 2018b, 2019b; Chalkidis et al., 2019, 2021; Paul et al., 2022).

3.2.6 Prior Case Retrieval (PCR)

Task Motivation and Description: When framing a legal document, legal experts (judges and lawyers) use their expertise to cite previous cases to support their arguments/reasoning. Legal experts have relied on their expertise to cite previous cases; however, with an exponentially growing number of cases, it becomes practically impossible to recall all possible cases. **Given a query document (without citations), the task of Prior Case Retrieval (PCR) is to retrieve the legal documents from the candidate pool that are relevant (and hence can be cited) in the given query document.** Automating this process directly impacts the justice delivery logistics. Moreover, including this task in the benchmark incorporates the retrieval aspects and understanding of legal similarity (as opposed to semantic similarity), opening research directions for retrieval systems in the legal domain.

Dataset: For the task of PCR we use the `Indian Legal Prior Case Retrieval (IL-PCR)` corpus (Joshi et al., 2023) (details in App. A). To the best of our knowledge, IL-PCR is the largest publicly available retrieval dataset for the Indian judicial system, making it a suitable candidate to be added to the benchmark.

Task Evaluation: Micro-averaged F1 score is used as the evaluation metric. Each candidate is assigned a relevance score by the model based on the given query case. The prediction (i.e., whether a candidate is cited) is based on the Top-ranked candidates.

3.2.7 Summarization (SUMM)

Task Motivation and Description: Summarization is a standard task in NLP; however, as mentioned in §1, summarizing legal documents requires legal language understanding and reasoning. **The task of summarization involves generating a gist (of a legal document) that captures the critical aspects of the case.** Summarization could be extractive (selecting the important sentences) or abstractive (generating the gist). In our setting, summarization is an abstractive generation task.

Dataset: For the summarization task, it is necessary to have a large dataset with gold summaries. Consequently, we use the `In-Abs` dataset (Shukla et al., 2022), created from judgment documents from the Supreme Court of India. The dataset consists of 7130 case documents with abstractive summaries (also called "headnotes").

Task Evaluation: We use standard metrics for summarization such as ROUGE-1, ROUGE-2, ROUGE-L F1-scores and BERT-SCORE (Zhang et al., 2019) (details in App. A).

3.2.8 Legal Machine Translation (L-MT)

Task Motivation and Description: In the Indian legal setting, when a case is transferred (due to re-

Task	Best Result	Metric	Model Details
L-NER	80.57%	strict mF1	InLegalBERT + CRF
RR	69.01%	mF1	MTL-BERT
CJPE	76.55% 0.42 0.18	mF1 ROUGE-L BLEU	XLNet + BiGRU
BAIL	81%	mF1	TF-IDF + IndicBERT
LSI	28.08%	mF1	LeSICiN (Graph-based Model)
PCR	39.15%	μF1@K	Event-based Model
SUMM	0.33 0.86	ROUGE-L BERTScore	Legal-LED
L-MT	0.28 0.32 0.57	BLEU GLEU chrF++	MSFT

Table 3: Summary of Models and Results for tasks.

Task	0-Shot	1-Shot	2-Shot	Metric
L-NER	30.59%	23.68%	32.84%	strict mF1
RR	30.95%	30.05%	30.31%	mF1
CJPE	54.17% 0.39 0.02	51.46% 0.29 0.03	56.74% 0.36 0.03	mF1 ROUGE-L BLEU
BAIL	51.04%	46.35%	61.0%	mF1
LSI	21.55%	22.61%	21.43%	mF1
SUMM	0.27 0.85	0.16 0.83	0.19 0.85	ROUGE-L BERTScore
L-MT	0.23 0.28 0.42	0.25 0.28 0.43	0.26 0.29 0.43	BLEU GLEU chrF++

Table 4: Performance of Open-AI-GPT (gpt-3.5-turbo-16k) model on various tasks for zero-shot, one-shot and two-shot settings.

appeal) from a district court to a High court, the corresponding document (typically in a regional language) needs to be translated to English. Additionally, since a large majority of the Indian population is not proficient in English, High Court / Supreme Court documents often need to be translated from English to Indian languages for a better understanding of the involved parties. In both scenarios, such translations, if done by humans, become a primary reason for delay in administering justice. Machine translation (MT) can augment human translators who could post-edit the translated document rather than translating from scratch. As outlined in §1, legal documents have different lexicons and styles; hence, existing MT systems do not perform well (Mahapatra et al., 2023). Given that many Indian languages are low-resource, MT becomes even more challenging, requiring specialized models for translating legal documents in low-resource Indian languages. **The task of Legal Machine Translation (L-MT) is to translate text in English to Indian languages and vice-versa.**

Dataset: For this task, we use the Multilingual Indian Legal Parallel Corpora (MILPaC) (Mahapatra et al., 2023), which comprises of a total of 17,853 parallel text pairs across English and 9 Indian languages, namely, Bengali (BN), Hindi (HI), Gujarati (GU), Malayalam (ML), Marathi (MR), Telugu (TE), Tamil (TA), Punjabi (PA) and Oriya (OR) (details in App. A).

Task Evaluation: We use standard metrics such as BLEU, GLEU, and chrF++ (details in App. A).

3.3 Relevance of tasks to Litigation Process

In general, considering the pipeline of a litigation process for a case, all the tasks in the IL-TUR benchmark help formulate various ways in which automatic legal language processing can augment legal practitioners. Among the tasks, LSI is considered one of the first steps in the judicial process – right after identifying the facts, legal personnel must find out the violated statutes of the law. Since India follows a mixture of civil and common law systems, identifying the statutes is not the sole basis of legal reasoning; precedent cases must also be considered (PCR task). Subsequently, the final step in the litigation process is to decide the outcome of the case; the CJPE and BAIL tasks are relevant in this case, and human judges can use corresponding models to get suggestions/recommendations.

The tasks L-NER, RR, and SUMM, though not directly required for the judicial process, significantly help Legal practitioners (e.g., lawyers conducting legal research to argue an ongoing case) to get a quick understanding of the documents. Sometimes, a case gets re-appealed in a higher court, and consequently, the case document (in a regional language) in the lower court needs to be translated into English (L-MT Task).

4 Models, Experiments and Results

We extensively experimented with various models for each of the proposed tasks, including transformer-based language models. Table 3 summarizes various baseline models and results for different tasks. Due to space limitations, we provide only the top-performing models here; details of experiments and other models are in App. B. In general, results indicate that tasks are far from being solved, and more research is required. We experimented with both generic BERT model (Devlin et al., 2019) and legal domain-specific BERT models: LegalBERT (Chalkidis et al., 2020), CaseLawBERT (Zheng et al., 2021), and InLegalBERT (Paul et al., 2023). For L-NER, InLegalBERT (with CRF on top) shows the best performance, possibly because of in-domain data pre-training. For the RR task, vanilla BERT (or other transformers) and Legal-BERT do not work well; hence, RR prediction is posed as a sequence prediction problem (at the sentence level), and the Multi-Task Learning (MTL) model based on BERT developed by Malik et al. (2022) shows the best performance. Since legal documents are long, and BERT has a limitation of 512 tokens in the input, for the CJPE task, hierarchical XLNet (XLNet and Bi-GRU on top of that) (Malik et al., 2021) works best. For BAIL prediction, since the documents are in Hindi, IndicBERT (Kakwani et al., 2020), a BERT model trained on Indian languages, was used. A pre-filtering of salient sentences, followed by IndicBERT, works best (Kapoor et al., 2022). For the LSI task, we conduct experiments with hierarchical LegalBERT and InLegalBERT, along with LeSICiN, a graph-based method proposed by Paul et al. (2022). We observe that LeSICIN outperforms the BERT-based methods. For the PCR task, an event-based model works the best (Joshi et al., 2023). An event refers to an action/activity (in the form of a predicate (typically a verb) and corresponding arguments) mentioned in the document. For SUMM, Legal-LED (HuggingFace, a) performs the best, and the commercially available Microsoft Azure Cognitive Services Translation API works best for L-MT tasks. In general, across all tasks (except PCR, SUMM, and L-MT) BERT (or its variant) performs the best.

We also conducted some initial (zero/one/two-shots) experiments with LLMs. In particular, we experimented with large models (in terms of the number of parameters) like Open-AI GPT (`gpt-3.5-turbo-16k`) and smaller models like GPT-Neo (Black et al., 2021) family of three models (GPT-Neo-125M, GPT-Neo-1.3B, GPT-Neo-2.7B), GPT-J-6B (Wang and Komatsuzaki, 2021), Llama-2-7b-chat-hf (Touvron et al., 2023), and Mistral-7B-v0.1 (Jiang et al., 2023). Table 4 shows the results for the Open-AI GPT model (details about prompts, other settings, and models are provided in App. C). We could not experiment with GPT for PCR since it requires a comparison between the query document and a pool of candidate documents, and passing the content of all the documents to GPT exceeds its token length limit (16,000 tokens). As observed, the GPT model performs worse than the SOTA models for each of the tasks. This is possibly because the tasks are quite complex, and require reasoning across long contexts, and also, for some tasks like L-NER and RR, it can be hard to come up with output formats that the model can understand in a zero-shot setting. Results for one-shot and two-shot show a similar trend. In some cases, one-shot performance is worse than zero-shot performance (also observed in other works (Brown et al., 2020). Experiments with smaller models (GPT-Neo-125M, GPT-Neo-1.3B, GPT-Neo-2.7B, GPT-J-6B, Llama-2-7b-chat-hf, and Mistral-7B-v0.1) showed similar trends (details in App. C).

Tasks in **IL-TUR** are quite varied, requiring different types of knowledge and skills. Developing systems for the Legal domain is not easy. The legal domain has inherent challenges (§1). Moreover, legal datasets are expensive to annotate; consequently, these are usually relatively small in size, and hence, learning in low resource setting is challenging. Experiments indicate that transformers fine-tuned on legal texts have shown limited success in the legal domain. Further, LLMs like Chat-GPT, which have demonstrated SOTA results in other domains (and have been shown to pass the bar exam (Chalkidis, 2023)), have not performed well on the IL-TUR benchmark. Hence, the trends indicate the need for further research in the legal domain.

5 Conclusion and Future Directions

This paper presented **IL-TUR**, a benchmark for Indian Legal Text Understanding and Reasoning. The benchmark has eight tasks requiring different types of legal skills to solve. Results indicate that the tasks are far from solved using state-of-the-art transformer-based models and LLMs. The list of tasks in **IL-TUR** is not exhaustive, and we are working towards expanding the list of tasks in the future; for example, we are working on developing foundational tasks like **Legal Coreference Resolution (L-Coref)** that is required for various applications such as information extraction and knowledge graph creation. Although such tasks have been addressed well in general NLP, our initial experiments show that using off-the-shelf NLP toolkits do not perform well on legal texts. Due to the usage of specialized terms, new models must be developed for the legal domain. On the modeling side, in the future, we plan to develop one model that generalizes and works across all the tasks (e.g., mT5 (Xue et al., 2020) and Multi-task Adapters (Pfeiffer et al., 2020)). Overall, we are hopeful that **IL-TUR** and its successive versions would create excitement in the Legal-NLP community and lead to the development of new technologies that could benefit society immensely and facilitate fair access to justice, a fundamental human right.

Limitations

IL-TUR is a first step towards creating a benchmark for the Indian Legal domain, which desperately needs technological solutions. The benchmark is not perfect and has certain limitations.

Given the dynamic nature of the legal domain, new cases and precedents keep getting added. Hence we plan to keep updating **IL-TUR** in the future.

The Legal domain is very wide and covers various areas such as criminal law, civil law, banking, insurance, etc. In **IL-TUR**, we could not cover each of the sub-domains in each of the tasks as it is a time-consuming and expensive affair to annotate a large number of documents. One of our goals for **IL-TUR** is to test the cross-area generalization abilities of models, nevertheless, we would expand the datasets of each of the tasks in the future.

IL-TUR is multi-lingual only with respect to the L-MT task. Additionally, the BAIL task is in Hindi. All the High Courts and the Supreme Court in India use English as the official language. Hindi is the prominent language used in the district courts in the majority of the north Indian states. Nevertheless, India is a multi-lingual society, and legal models for other languages should also be developed for more tasks in the legal domain. We plan to extend the benchmark in the future and include some more tasks in Indian languages. The main challenge in doing so is a scarcity of legal data in regional languages in digital format at the district court level.

Dataset of some of the tasks (e.g., LSI) uses ML-based trained models (that may not be perfect) in the dataset creation process (fact extraction in the case of LSI). Extracting facts manually at a large scale is an expensive and time-consuming effort, nevertheless, in the future, we plan to employ legal professionals and create a more refined dataset.

Regarding LLM experiments, some of the tasks such as BAIL, and CJPE require the entire document to be a part of the model's input. Obtaining LLM predictions overall test set samples can be challenging in terms of expense and computation. Hence, we evaluated over a smaller subset assuming that it is a good proxy of LLM performance.

Lastly, the benchmark has only eight tasks. Creating legal tasks is time-consuming and expensive since it requires legal experts' help. Nevertheless, as explained earlier, **IL-TUR** is a work in progress, and we will keep growing by adding more tasks. In this work, we presented different models for various tasks; although many of the models (e.g., BERT) are common across all tasks, it would be nice to have a single model that could solve all the tasks (e.g., mT5), in future, we plan to explore such model. Developing such models is a computationally expensive process.

Ethical Considerations

We use publicly available and open-source datasets for the tasks; no copyright is infringed. To the best of our knowledge, the five of the proposed tasks (L-NER, RR, LSI, PCR, and Summ) do not have any direct ethical consequences since the proposed tasks are mainly related to information retrieval and summarization. Moreover, the tasks are meant to encourage the development of systems that would lead to streamlining the legal workflow and will not directly affect the life of any personnel.

Two of the tasks (CJPE and BAIL) have ethical considerations. Given a large quantum of pending cases in lower courts (district courts), these tasks aim to develop systems that augment a judge and not replace them; consequently, the systems are meant to provide recommendations, and a human judge takes the final decision. We follow all the steps as done by Malik et al. (2021); Kapoor et al. (2022) to avoid any bias in the data. For example, we removed cases (documents) related to sensitive issues like rape and sexual violence. For all the tasks, the documents are selected randomly (and anonymized) to avoid bias towards any entity, organization, or law.

Please note we do not endorse the use of the benchmark data for non-research (commercial and real-life) applications, and the primary motivation for creating the IL-TUR benchmark is to consolidate all the research happening in parallel for the Indian Legal domain. Hence we will release the benchmark and datasets under the Creative Common Attribution-NonCommercial-ShareAlike (CC BY-NC-SA) license. Moreover, we believe providing a platform by maintaining a common leaderboard for multiple tasks will advance the field with more transparency and reproducibility.

Since Legal-NLP is a relatively new area, to the best of our abilities, we have taken all steps concerning ethical considerations and privacy. Via these tasks, we want to encourage more research in this area so that any hidden factors that could not have been thought of beforehand are also brought to light.

References

Khalid Al-Kofahi, Alex Tyrrell, Arun Vachher, and Peter Jackson. 2001. A Machine Learning Approach to Prior Case Retrieval. In *Proceedings of the Eighth International Conference on Artificial Intelligence and Law (ICAIL)*.

Paheli Bhattacharya, Shounak Paul, Kripabandhu Ghosh, Saptarshi Ghosh, and Adam Wyner. 2019. Identification of Rhetorical Roles of Sentences in Indian Legal Judgments. *CoRR*, abs/1911.05405.

Sid Black, Gao Leo, Phil Wang, Connor Leahy, and Stella Biderman. 2021. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow. If you use this software, please cite it using these metadata.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Ilias Chalkidis. 2023. Chatgpt may pass the bar exam soon, but has a long way to go for the lexglue benchmark. *arXiv preprint arXiv:2304.12202*.

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(ACL)*.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online. Association for Computational Linguistics.

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

Ilias Chalkidis, Abhik Jana, Dirk Hartung, Michael Bommarito, Ion Androutsopoulos, Daniel Katz, and Nikolaos Aletras. 2022a. LexGLUE: A benchmark dataset for legal language understanding in English. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4310–4330, Dublin, Ireland. Association for Computational Linguistics.

Ilias Chalkidis, Tommaso Pasini, Sheng Zhang, Letizia Tomada, Sebastian Schwemer, and Anders Søgaard. 2022b. FairLex: A multilingual benchmark for evaluating fairness in legal text processing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4389–4406, Dublin, Ireland. Association for Computational Linguistics.

Karl Cobbe, Chris Hesse, Jacob Hilton, and John Schulman. 2020. Leveraging procedural generation to benchmark reinforcement learning. In *International conference on machine learning*, pages 2048–2056. PMLR.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

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 (NAACL-HLT)*.

Zhiwei Fei, Xiaoyu Shen, Dawei Zhu, Fengzhe Zhou, Zhuo Han, Songyang Zhang, Kai Chen, Zongwen Shen, and Jidong Ge. 2023. Lawbench: Benchmarking legal knowledge of large language models. *arXiv preprint arXiv:2309.16289*.

Joseph L Fleiss, Bruce Levin, and Myunghee Cho Paik. 2013. *Statistical methods for rates and proportions*. john wiley & sons.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The Pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Alexios Gidiotis and Grigorios Tsoumakas. 2020. A divide-and-conquer approach to the summarization of academic articles. *CoRR*, abs/2004.06190.

Neel Guha, Julian Nyarko, Daniel E. Ho, Christopher Ré, Adam Chilton, Aditya Narayana, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel N. Rockmore, Diego Zambrano, Dmitry Talisman, Enam Hoque, Faiz Surani, Frank Fagan, Galit Sarfaty, Gregory M. Dickinson, Haggai Porat, Jason Hegland, Jessica Wu, Joe Nudell, Joel Niklaus, John Nay, Jonathan H. Choi, Kevin Tobia, Margaret Hagan, Megan Ma, Michael Livermore, Nikon Rasumov-Rahe, Nils Holzenberger, Noam Kolt, Peter Henderson, Sean Rehaag, Sharad Goel, Shang Gao, Spencer Williams, Sunny Gandhi, Tom Zur, Varun Iyer, and Zehua Li. 2023. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models.

Yulan Guo, Jun Zhang, Min Lu, Jianwei Wan, and Yanxin Ma. 2014. Benchmark datasets for 3d computer vision. In *2014 9th IEEE Conference on Industrial Electronics and Applications*, pages 1846–1851. IEEE.

Prakhar Gupta, Matteo Pagliardini, and Martin Jaggi. 2019. Better word embeddings by disentangling contextual n-gram information. In *NAACL-HLT (1)*, pages 933–939. Association for Computational Linguistics.

Zhanying He, Chun Chen, Jiajun Bu, Can Wang, Lijun Zhang, Deng Cai, and Xiaofei He. 2012. Document summarization based on data reconstruction. In *Proc. AAAI Conference on Artificial Intelligence*, pages 620–626.

Montani Ines Honnibal Matthew and Boyd Adriane Van Landeghem Sofie. 2020. spaCy: Industrial-Strength Natural Language Processing in Python.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *International Conference on Machine Learning*, pages 4411–4421. PMLR.

HuggingFace. a. Legal LED. `https://huggingface.co/nsi319/legal-led-base-16384`. [Online].

HuggingFace. b. Legal Pegasus. `https://huggingface.co/nsi319/legal-pegasus`. [Online].

Wonseok Hwang, Dongjun Lee, Kyoungyeon Cho, Hanuhl Lee, and Minjoon Seo. 2022. A multi-task benchmark for korean legal language understanding and judgement prediction. *arXiv preprint arXiv:2206.05224*.

Peter Jackson, Khalid Al-Kofahi, Alex Tyrrell, and Arun Vachher. 2003a. Information extraction from case law and retrieval of prior cases. *Artificial Intelligence*, 150(1-2):239–290.

Peter Jackson, Khalid Al-Kofahi, Alex Tyrrell, and Arun Vachher. 2003b. Information Extraction from Case Law and Retrieval of Prior Cases. *Artificial Intelligence, Elsevier*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Abhinav Joshi, Akshat Sharma, Sai Kiran Tanikella, and Ashutosh Modi. 2023. U-CREAT: Unsupervised Case Retrieval using Events extrAcTion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, NC Gokul, 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.

Prathamesh Kalamkar, Astha Agarwal, Aman Tiwari, Smita Gupta, Saurabh Karn, and Vivek Raghavan. 2022a. Named entity recognition in Indian court judgments. In *Proceedings of the Natural Legal Language Processing Workshop 2022*, pages 184–193, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Prathamesh Kalamkar, Aman Tiwari, Astha Agarwal, Saurabh Karn, Smita Gupta, Vivek Raghavan, and Ashutosh Modi. 2022b. Corpus for Automatic Structuring of Legal Documents. In *Proceedings of the 13th Language Resources and Evaluation Conference -Association for Computational Linguistics (ACL-LREC)*.

Arnav Kapoor, Mudit Dhawan, Anmol Goel, Arjun T H, Akshala Bhatnagar, Vibhu Agrawal, Amul Agrawal, Arnab Bhattacharya, Ponnurangam Kumaraguru, and Ashutosh Modi. 2022. HLDC: Hindi Legal Documents Corpus. In *Findings of the Association for Computational Linguistics (ACL)*.

Nikolaos Lagos, Frederique Segond, Stefania Castellani, and Jacki O'Neill. 2010. Event extraction for legal case building and reasoning. In *International Conference on Intelligent Information Processing*, pages 92–101. Springer.

Michael Laskin, Denis Yarats, Hao Liu, Kimin Lee, Albert Zhan, Kevin Lu, Catherine Cang, Lerrel Pinto, and Pieter Abbeel. 2021. Urlb: Unsupervised reinforcement learning benchmark. *arXiv preprint arXiv:2110.15191*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Chao-Lin Liu and Kuan-Chun Chen. 2019. Extracting the gist of chinese judgments of the supreme court. In *Proc. International Conference on Artificial Intelligence and Law (ICAIL)*.

Dayiheng Liu, Yu Yan, Yeyun Gong, Weizhen Qi, Hang Zhang, Jian Jiao, Weizhu Chen, Jie Fu, Linjun Shou, Ming Gong, et al. 2020. Glge: A new general language generation evaluation benchmark. *arXiv preprint arXiv:2011.11928*.

Sayan Mahapatra, Debtanu Datta, Shubham Soni, Adrijit Goswami, and Saptarshi Ghosh. 2023. Improving access to justice for the indian population: A benchmark for evaluating translation of legal text to indian languages.

Vijit Malik, Rishabh Sanjay, Shouvik Kumar Guha, Angshuman Hazarika, Shubham Nigam, Arnab Bhattacharya, and Ashutosh Modi. 2022. Semantic Segmentation of Legal Documents via Rhetorical Roles. In *Proceedings of the Natural Legal Language Processing Workshop (NLLP) EMNLP*.

Vijit Malik, Rishabh Sanjay, Shubham Kumar Nigam, Kripabandhu Ghosh, Shouvik Kumar Guha, Arnab Bhattacharya, and Ashutosh Modi. 2021. 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 (ACL-IJCNLP)*.

Marie-Francine Moens, Caroline Uyttendaele, and Jos Dumortier. 1999. Abstracting of legal cases: the potential of clustering based on the selection of representative objects. *Journal of the American Society for Information Science*, 50(2):151–161.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Proc. AAAI Conference on Artificial Intelligence*.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Ranking sentences for extractive summarization with reinforcement learning. In *Proc. NAACL-HLT*, pages 1747–1759.

National Judicial Data Grid. 2023. National judicial data grid statistics. `https://www.njdg.ecourts.gov.in/njdgnew/index.php`.

Joel Niklaus, Veton Matoshi, Pooja Rani, Andrea Galassi, Matthias Stürmer, and Ilias Chalkidis. 2023. Lextreme: A multi-lingual and multi-task benchmark for the legal domain. *arXiv preprint arXiv:2301.13126*.

Shounak Paul, Pawan Goyal, and Saptarshi Ghosh. 2022. Lesicin: A heterogeneous graph-based approach for automatic legal statute identification from indian legal documents. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):11139–11146.

Shounak Paul, Arpan Mandal, Pawan Goyal, and Saptarshi Ghosh. 2023. Pre-trained language models for the legal domain: A case study on indian law.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. Mad-x: An adapter-based framework for multi-task cross-lingual transfer. *arXiv preprint arXiv:2005.00052*.

Seth Polsley, Pooja Jhunjhunwala, and Ruihong Huang. 2016. Casesummarizer: A system for automated summarization of legal texts. In *Proc. Iinternational conference on Computational Linguistics (COLING) System Demonstrations*.

Gowtham Ramesh, Sumanth Doddapaneni, Aravinth Bheemaraj, Mayank Jobanputra, Raghavan AK, Ajitesh Sharma, Sujit Sahoo, Harshita Diddee, Mahalakshmi J, Divyanshu Kakwani, Navneet Kumar, Aswin Pradeep, Srihari Nagaraj, Kumar Deepak, Vivek Raghavan, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh Shantadevi Khapra. 2022. Samanantar: The largest publicly available parallel corpora collection for 11 Indic languages. *Transactions of the Association for Computational Linguistics*, 10:145–162.

Joshua Robinson and David Wingate. 2023. Leveraging large language models for multiple choice question answering. In *The Eleventh International Conference on Learning Representations*.

Isabel Segura-Bedmar, Paloma Martínez, and María Herrero-Zazo. 2013. SemEval-2013 task 9 : Extraction of drug-drug interactions from biomedical texts (DDIExtraction 2013). In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 341–350, Atlanta, Georgia, USA. Association for Computational Linguistics.

Abhay Shukla, Paheli Bhattacharya, Soham Poddar, Rajdeep Mukherjee, Kripabandhu Ghosh, Pawan Goyal, and Saptarshi Ghosh. 2022. Legal case document summarization: Extractive and abstractive methods and their evaluation. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1048–1064, Online only. Association for Computational Linguistics.

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.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

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*, pages 275–282.

United Nations. 2023. India overtakes china as the world's most populous country. `https://www.un.org/development/desa/dpad/wp-content/uploads/sites/45/PB153.pdf`.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019a. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018a. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. `https://github.com/kingoflolz/mesh-transformer-jax`.

Pengfei Wang, Yu Fan, Shuzi Niu, Ze Yang, Yongfeng Zhang, and Jiafeng Guo. 2019b. Hierarchical Matching Network for Crime Classification. In *Proceedings of the 42nd International ACM Conference on Research and Development in Information Retrieval, (SIGIR)*.

Pengfei Wang, Ze Yang, Shuzi Niu, Yongfeng Zhang, Lei Zhang, and ShaoZhang Niu. 2018b. Modeling Dynamic Pairwise Attention for Crime Classification over Legal Articles. In *The 41st International ACM Conference on Research & Development in Information Retrieval (SIGIR)* .

Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. 2013. Online object tracking: A benchmark. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2411–2418.

Liang Xu, Hai Hu, Xuanwei Zhang, Lu Li, Chenjie Cao, Yudong Li, Yechen Xu, Kai Sun, Dian Yu, Cong Yu, et al. 2020. Clue: A chinese language understanding evaluation benchmark. *arXiv preprint arXiv:2004.05986*.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*.

Vikas Yadav and Steven Bethard. 2019. A survey on recent advances in named entity recognition from deep learning models. *arXiv preprint arXiv:1910.11470*.

Amy Zhang, Yuxin Wu, and Joelle Pineau. 2018. Natural environment benchmarks for reinforcement learning. *arXiv preprint arXiv:1811.06032*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

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

Appendix

Table of Contents

List of Tables

A Tasks and Dataset Details

We will release the baseline code along with a compiled list of task-specific datasets and evaluation scripts with the camera-ready version of the paper. The consolidated leaderboard website for the benchmark will be made public with the camera-ready release.

A.1 Legal Named Entity Recognition (L-NER)

As outlined in §3, we use a set of 12 Named Entity (NE) types for Legal NER. Fig. 2 shows an example. Table 5 shows the definition for 12 NE types/classes.

Dataset Details: Table 6 lists some important statistics about the NER dataset. The NE type label statistics are displayed along with the class descriptions in Table 5.

Annotation Details: For the L-NER task, we collected a total of 105 cases publicly available from the Supreme Court and a few High Courts of India by scrapping the website: `https://www.indiankanoon.org`. Please note that the IndianKanoon website allows free downloads of public documents. In discussion with legal experts, we decided on a comprehensive set of 12 NE (Named Entity) classes suited for the legal domain (Table 5). Two law students from a reputed law college in India were tasked with annotating the case documents. The annotation procedure involved the following steps:

- To ensure that entity spans are marked consistently, we discussed with both annotators how to mark every label. Such decisions involved leaving out prefixes/salutations such as 'Shri' (a polite way to address Mr. in the Indian context) and 'Smt.' (a polite way to address Ms. in the Indian context), 'Justice' (Honorific for a Judge), etc., from the entity names, including the (optional) precedent citations that follow case titles as part of the precedent (PREC) entities, and so on.

- We randomly chose a set of 25 documents, and each annotator worked on all 25 documents independently based on the rules devised in the previous step.

- We observed a high degree of agreement between the annotators for these 25 documents (Cohen's Kappa: 0.82, Krippendorff's Alpha: 0.85).

Broad Category	Label	Frequency	Description
Party	APPELLANT (APP)	660	Party filing an appeal to the court
	RESPONDENT (RESP)	516	Party against whom appeal has been filed
Legal Professional	JUDGE (JUD)	366	Judge of the current or prior/cited cases
	A.COUNSEL (AC)	288	Lawyer(s) on behalf of the appellant(s)
	R.COUNSEL (RC)	255	Lawyer(s) on behalf of the respondent(s)
Organizations	COURT (CRT)	1,572	Any court occurring in the document
	AUTHORITY (AUTH)	1,342	Any organization/body having administrative/legal authority
Other Person(s)	WITNESS (WIT)	312	Witness(es) who are testifying in the case
Legal References	STATUTE (STAT)	2,055	Citation to legal acts
	PRECEDENT (PREC)	1,804	Citation to prior cases
Legal Artefacts	DATE	2,316	Mention of any date in the case
	CASE NO. (CN)	1,102	Mention of any case number, including that of the current case

Table 5: Named Entity (NE) types used in the L-NER dataset

# Documents	105
# Labels	12
Total no. of words	648,937
Avg. Document Size (in #words)	6180.35
Total no. of entities (All occurrences)	12,588
Total no. of entities (Unique occ.)	5,658
Avg. no. of entities per doc (All occ.)	119.89
Avg. no. of entities per doc (Unique occ.)	53.89

Table 6: The dataset statistics for the L-NER task

- Both annotators worked together to resolve the disagreements to arrive at one single consolidated set of annotations for these 25 documents.
- Above steps performed over 25 documents calibrated the annotators and led to a high degree of agreement among them. Since annotation is an expensive and time-consuming process, the remaining 80 documents were split equally between the two annotators for annotation.

Task Evaluation: NER can be formulated as a sequence prediction task, where each word receives either of the labels {`B-X`, `I-X`, `O`} as per the popular 'B-I-O' scheme (Yadav and Bethard, 2019) ('X' represents any of the legal classes we are interested in). We use standard metrics of *strict* macro-averaged precision, recall, and F1 score for evaluation. The *strict* score assumes a correct match only if *both* the entity boundary and entity type are correctly predicted. L-NER evaluation F1 score metric is computed using `https://pypi.org/project/nervaluate/`. We use strict macro-averaged scores in our setup. The *strict* scoring mechanism ensures that a match is considered correct if the entity span and entity type are the same. In other words, if either the span is incorrect (the model predicts more/fewer tokens as part of the entity) or the predicted label type does not match the ground truth, the match is considered incorrect.

Comparison with existing L-NER datasets: Recently, Kalamkar et al. (2022a) released a dataset for L-NER over Indian legal documents. However, unlike our dataset, which comprises of *full-length documents* annotated with every occurrence of every NE, the dataset by Kalamkar et al. (2022a) consists of segments of documents and not full documents. This is a crucial difference since models trained on our data will be able to detect NEs even when provided with a snippet of a case document. There is also a slight variation in the set of NEs considered in our dataset as compared to those considered by Kalamkar et al. (2022b), although most common entity types have been covered in both datasets.

A.2 Rhetorical Role Prediction (RR)

In order to structure long legal documents, we consider Rhetorical Roles (RRs), where each sentence in the document is assigned one of 13 RRs (§3).

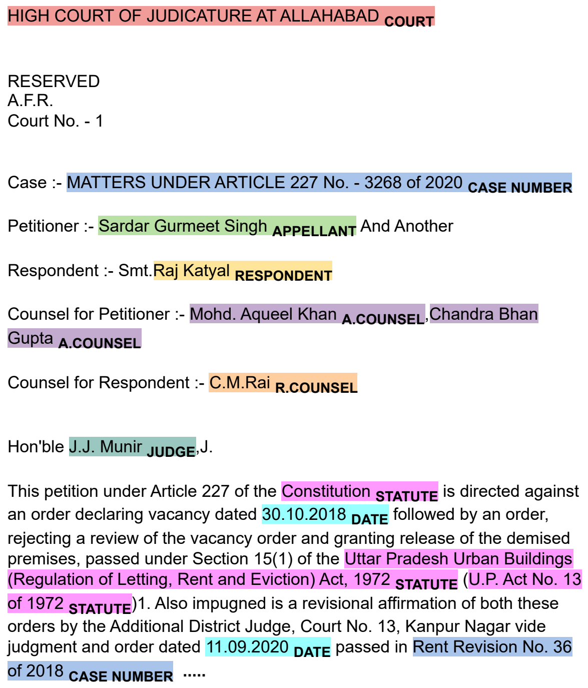

Figure 2: Example of L-NER

The definition of each of the Rhetorical Roles is provided in Table 7. We utilize the dataset and role definitions provided by prior work on structuring Indian legal documents (Malik et al., 2022).

Dataset Details: The RR dataset was created by scrapping (from IndianKanoon: https://indiankanoon.org/) publicly available documents from the Supreme Court of India, High Courts, and Tribunal courts. The documents pertain to Banking/Income Tax law (IT) and Competition Law (CL) (also called as Anti-Trust Law in the US). The dataset consists of $21,184$ sentences annotated with 13 RRs. Figure 4 shows the distribution of RR labels. The dataset is split randomly (at document level) into 80% train, 10% validation, and 10% test set.

Annotation Details: The dataset was annotated by six legal experts (graduate law student researchers), three annotated CL documents, and the remaining three annotated IT documents (Malik et al., 2022). The annotators showed a high degree of agreement. The Fleiss kappa (Fleiss et al., 2013) between the annotators is 0.65 for the IT domain and 0.87 for the CL domain, indicating a substantial agreement between annotators. Annotating RR is not a trivial task, and annotators can have disagreements. Several strategies were employed to resolve these disagreements. More details about annotation case studies can be found in Malik et al. (2022).

Evaluation: RR Prediction is evaluated using standard Macro F1 metric. Macro F1 is the average F1 score calculated per class.

A.3 Court Judgment Prediction with Explanation (CJPE)

Dataset Details: We use the ILDC-multi dataset (Malik et al., 2021) for CJPE, which consists of 34k cases from the Supreme Court of India (SCI). These cases consist of multiple appeals, which can contain corresponding decisions for each appeal. However, since the task has been posed as binary text classification, the final decision is considered as ACCEPT if *at least one* appeal is accepted, otherwise REJECT. The documents are stripped of the final decision given by the Judge with the help of regex-based matching. Table 8 provides details of the dataset.

Annotation Details: The explanation aspect of the CJPE task was annotated with the help of 5 legal experts (Malik et al., 2021). The annotators were graduate students and a law professor from a reputed law school. The annotators were not shown the final decision of the case. They were asked to predict the final decision and annotate the sentences (explanations) in the document that led to the final decision. More details about agreement among the annotators are provided in (Malik et al., 2021). In a nutshell, the average prediction F1 score of annotators w.r.t. to the ground truth judgment was 94.32%. This points towards the challenging nature of the CJPE task; as pointed out earlier, India has a common-law system, and hence, judges could override existing precedents. Disagreements among the annotators were mainly due to differences in the linguistic interpretation of the case and law. For the explanation part, similar trends are reported with the average agreement in terms of the BLEU score to be around 0.4.

Evaluation: The prediction part of the CJPE task is evaluated using standard F1 score metric, and the explanation part is evaluated using BLEU and ROUGE scores.

A.4 Bail Prediction (BAIL)

For the task of BAIL prediction, Kapoor et al. (2022) created a corpus of 900k Hindi Legal Documents (referred to as HLDC (Hindi Legal Document Copus)). The corpus is created by scrapping publicly available documents on the eCourts website (`https://ecourts.gov.in/ecourts_home/`). The documents are scrapped

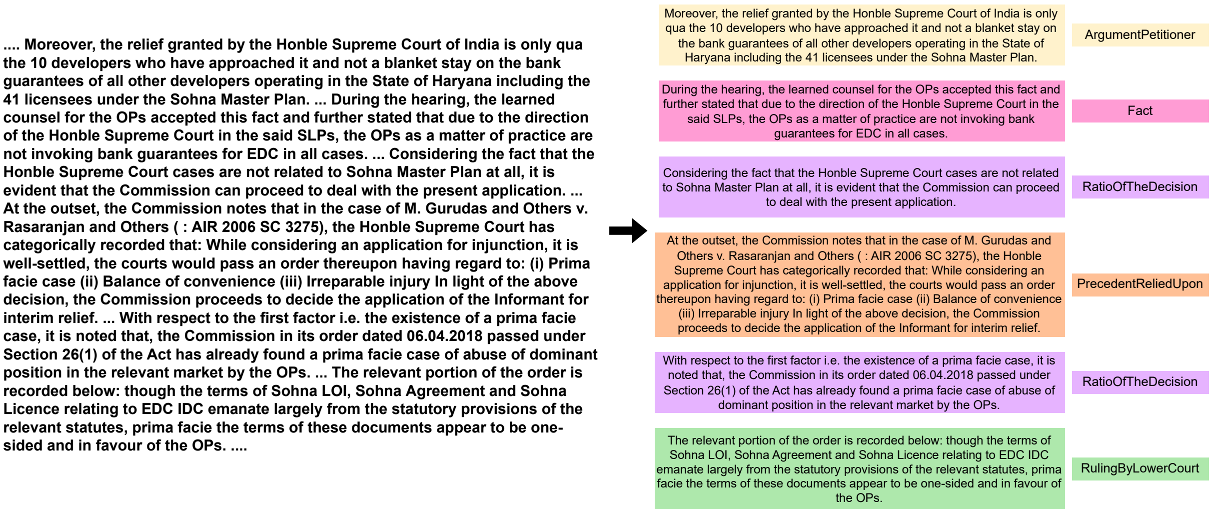

Figure 3: Example of the Rhetorical Role Prediction Task (Kalamkar et al., 2022b)

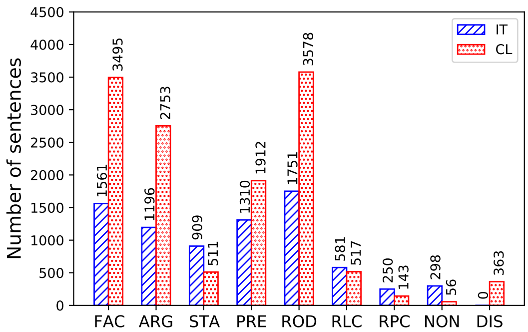

Figure 4: Distribution of RR labels in IT and CL documents (Malik et al., 2022).

from district courts of the state of Uttar Pradesh (a Hindi-speaking state in northern India). The data is anonymized to take care of biases and ethical aspects; please refer to (Kapoor et al., 2022) for more details. Bail cases in HLDC are pre-processed to remove the final decision (using regex) since we aim to predict this automatically. More details about the dataset are discussed in (Kapoor et al., 2022). For model training and evaluation, we divide the data into train, validation, and test split in the ratio of 70:10:20.

Evaluation: The BAIL prediction is a binary task; it is evaluated using the standard macro-F1 score metric.

A.5 Legal Statute Identification (LSI)

Table 9 shows an example of LSI task. We utilize the ILSI dataset for this task, which comprises of 100 target statutes from the Indian Penal Code (IPC), the main legislation codifying criminal laws in India.

Dataset Preprocessing: The LSI task requires the input to be *only* the facts of the case, and thus, an automated RR method (Bhattacharya et al., 2019) was employed to extract the facts. Since this method is not foolproof, some sentences containing statute citations may get mislabeled as facts. The version of the dataset released by Paul et al. (2022) contains some unmasked statute citations. Thus, we used an existing automated Legal NER method (Kalamkar et al., 2022a), which can identify both the act/law names and the statute/section references in the text, to mask all possible statute and act references (statutes from all acts were masked, not just IPC). To prevent model biases, we also masked all entities identified by the Legal NER method.

Dataset Details: Table 10 lists some important statistics about the ILSI dataset. In addition to the facts extracted from case documents and their corresponding statute mappings, Paul et al. (2022) also provided the statute descriptions as part of the dataset.

Evaluation: LSI is formulated as a multi-label text classification task. The facts, a functional segment of the entire case document, are provided as input. The expected output is one or more statutes from a list of target statutes relevant to the given fact portion. Standard classification metrics such as macro-averaged precision, recall, and F1 score are used for evaluation. In principle, the LSI task can also be considered a retrieval task instead of a multi-label classification task, i.e., the task is to retrieve from a dynamic set of statutes and provide a bigger pool of relevant candidates to be retrieved

Rhetorical Role Label	Definition
Fact (FAC)	These are the facts specific to the case based on which the arguments have been made and judgment has been issued. In addition to Fact, we also have the fine-grained label
Issues (ISS)	The issues which have been framed/accepted by the present court for adjudication.
Argument Petitioner (ARG-P)	Arguments which have been put forward by the petitioner/appellant in the case before the present court and by the same party in lower courts (where it may have been petitioner/respondent)
Argument Respondent (ARG-R)	Arguments which have been put forward by the respondent in the case before the present court and by the same party in lower courts (where it may have been petitioner/respondent)
Statute (STA)	The laws referred to in the case.
Dissent (DIS)	Any dissenting opinion expressed by a judge in the present judgment/decision.
Precedent Relied Upon (PRE-R)	The precedents which have been relied upon by the present court for adjudication. These may or may not have been raised by the advocates of the parties and amicus curiae.
Precedent Not Relied Upon (PRE-NR)	The precedents which have not been relied upon by the present court for adjudication. These may have been raised by the advocates of the parties and amicus curiae.
Precedent Overruled (PRE-O)	Any precedents (past cases) on the same issue that have been overruled through the current judgment.
Ruling By Lower Court (RLC)	Decisions of the lower courts which dealt with the same case.
Ratio Of The Decision (ROD)	The principle that has been established by the current judgment/decision which can be used in future cases. Does not include the obiter dicta which is based on observations applicable to the specific case only.
Ruling By Present Court (RPC)	The decision of the court on the issues that have been framed/accepted by the present court for adjudication.
None (NON)	any other matter in the judgment which does not fall in any of the above-mentioned categories.

Table 7: Definitions for different Rhetorical Roles

Corpus (Avg. tokens)	Number of docs (Accepted Class %) Train	Validation	Test
ILDC-multi (3231)	32305 (41.43%)	994 (50%)	1517 (50.23%)
ILDC-single (3884)	5082 (38.08%)		
ILDC-expert (2894)	56 (51.78%)		

Table 8: Statistics for the CJPE dataset (ILDC) (Malik et al., 2021)

for a particular query document having facts. However, as it is an initial phase of establishing the benchmark, we followed the classification setting proposed in previous works (Wang et al., 2018b, 2019b; Chalkidis et al., 2019, 2021).

A.6 Prior Case Retrieval (PCR)

The IL-PCR dataset (Joshi et al., 2023) is used for the PCR task. The IL-PCR corpus was created by scraping legal documents (available in the public domain) from the website IndianKanoon (https://indiankanoon.org/). The pool of documents is expanded by scraping documents cited by documents scraped previously. It was done to ensure sufficient citation links from the query to the candidate pool in the final dataset. Names of individuals and organizations were anonymized to the <NAME> and <ORG> tags, respectively, using a NER model (Honnibal Matthew and Van Landeghem Sofie, 2020) and a manually compiled gazetteer. This anonymization step is especially pertinent to the PCR task as it removes any biases in the judgment based on entity names. The ground truth labels mark all the candidate's cases relevant to each query case. Statistics for the IL-PCR corpus are shown in Table 11.

Evaluation: The PCR task uses micro-averaged F1@K score as the evaluation metric (as done in previous work: `https://sites.ualberta.ca/~rabelo/COLIEE2021/`). Prediction models predict a relevance score for each candidate for a given query. Top-K-ranked candidates are considered for prediction (i.e., whether a candidate is cited or not).

A.7 Summarization (SUMM)

We use the summarization dataset In-Abs (Shukla et al., 2022). As mentioned in Sec 3.2.7, the dataset comprises of 7,130 case documents from the Supreme Court of India. These documents were

Facts of the case	"On the fateful day at about 9.30 a.m. deceased accompanied by [PERSON1] (PW 4) and [PERSON2] (PW 7) was going from his village Talod to Alote. The accused persons were hiding behind bushes on the road near village Gharola. They were armed with lathies and farsies. When the deceased and the aforesaid two persons reached near the Khakhra, the respondents surrounded them and started attacking the deceased with weapons with which they were armed. His nose was cut. PWs. 4 and 7 tried to intervene, but they were also attacked by the accused persons as a result of which they also received injuries. The two witness rushed to the police station where PW 4 lodged the FIR (Exhibit P-10). The deceased in injured condition was taken to the hospital, and later he succumbed to the injuries. Post-mortem was conducted and large number of injuries were found on his body. During investigation the alleged weapons of the assailants were seized. After investigation charge sheet was placed."
IPC S.324	*Voluntarily causing hurt by dangerous weapons or means*
IPC S.302	*Punishment for murder*

Table 9: Example of the LSI task, fact section taken a High Court Document **"State Of Madhya Pradesh vs. Mansingh And Ors. on 13 August, 2003"**, along with the IPC Sections (324 and 302) that the case cites.

Dataset	**ILSI**
# Documents	66,090
# Labels	100
Train/Dev/Test Split	42,835/ 10,200/ 13,039
Avg. Document Size (in #words)	2406
Avg. no. of citations (#labels per doc)	3.78

Table 10: The table shows the dataset statistics for the ILSI dataset.

Dataset	**IL-PCR**
# Documents	7070
Avg. Document Size	8093.19
# query Documents	1182
Vocab Size	113340
Total Citation Links	8008
Avg. Citation Links per query	6.775
Language	English

Table 11: The table shows the IL-PCR dataset statistics

collected from the website of the Legal Information Institute of India (http://www.liiofindia.org/in/cases/cen/INSC/), which provides free and non-profit access to databases of Indian law. These documents are accompanied by additional notes called "headnotes", which enumerate the important issues and aspects of the case. Legal experts write these headnotes and can be considered abstractive summaries of the entire case document. Headnotes usually occur in the top part of the document, just below the document header (which contains party names, date, bench, etc.), and just above the main judgment. They are also usually preceded by the heading "HEADNOTE:". Shukla et al. (2022) used these cues, and additionally employed regular expression matching to extract the headnotes from the judgment. Table 12 provides some statistics of this dataset (more details in Shukla et al. (2022)).

Dataset	**In-Abs**
# Documents	7,130
Type of Summary	Abstractive
Language	English
Train/Test Split	7,030/100
Avg. Document size (in #words)	4376.98
Avg. Summary size (in #words)	842.52
Avg. Compression Ratio	0.235

Table 12: The table shows the statistics of the In-Abs dataset

Evaluation: Following Shukla et al. (2022), we use standard summarization metrics such as ROUGE-1, ROUGE-2, and ROUGE-L F1-scores (computed using https://pypi.org/project/py-rouge/, with *max_n* set to 2, parameters *limit_length* and *length_limit* not used, and other parameters kept as default), and BertScore (Zhang et al., 2019) (computed using https://pypi.org/project/

	EN	BN	HI	MR	TA	TE	ML	PA	OR	GU
EN	×	110	114	114	114	112	114	114	114	114
BN	*365*	×	110	110	110	108	110	110	110	110
HI	*365*	*365*	×	114	114	112	114	114	114	114
MR	*365*	*365*	*365*	×	114	112	114	114	114	114
TA	*365*	*365*	*365*	*365*	×	112	114	114	114	114
TE						×	112	112	112	112
ML							×	114	114	114
PA								×	114	114
OR									×	114
GU										×

Table 13: Number of parallel text units per language pair in (1) **MILPaC-IP** - black entries in upper triangular part, and (2) **MILPaC-CCI-FAQ** - blue italicized entries in lower triangular part. For both datasets, text units are QA-pairs, hence not tokenized into sentences (details in text).

	EN	BN	HI	MR	TA	TE	ML	PA	OR	GU
EN	×	739	706	578	418	319	443	261	256	316
BN		×	439	439	×	319	438	×	×	×
HI			×	578	×	319	443	262	256	×
MR				×	×	319	443	133	128	×
TA					×	×	×	×	×	×
TE						×	319	×	×	×
ML							×	×	×	×
PA								×	256	×
OR									×	×
GU										×

Table 14: Number of Parallel Text units per language pair in **MILPaC-Acts**. Text units are tokenized into sentences for this dataset.

bert-score/, version 0.3.4) that calculates the semantic similarity scores using the pre-trained BERT model.

A.8 Machine Translation (MT)

For this task, we use the Multilingual Indian Legal Parallel Corpora (MILPaC) (Mahapatra et al., 2023), which comprises the following 3 datasets, and a total of 17,853 parallel text pairs across English and 9 Indian languages, namely, Bengali (BN), Hindi (HI), Gujarati (GU), Malayalam (ML), Marathi (MA), Telugu (TE), Tamil (TA), Punjabi (PA) and Oriya (OR):

MILPaC-IP: Developed from a set of primers released by a society of law practitioners, this contains a set of approximately 57 question-answer pairs related to Indian Intellectual Property Laws, developed in EN and 9 Indian languages –BN, HI, MR, TA, GU, TE, ML, PA, OR. The details of the dataset are shown in Table 13

MILPaC-CCI-FAQ: is developed from a set of QA booklets released by the Competition Commission of India and contains 184 QA pairs on statutory rules based on competition issues in India. The parallel corpus has been developed for EN and 4 Indian languages — BN, HI, MA, and TA (see Table 13).

MILPaC-Acts: has been developed from 10 popular Indian Acts (statutory documents outlining laws of the country), for which official translations (from the Indian legislature) were available in English and the 9 Indian languages used in MILPaC-IP. For details, see Table 14.

The exact no. of pairwise samples are shown in Table 13 (MILPaC-IP and MILPaC-CCI-FAQ) and Table 14. For more details regarding the creation and curation of the dataset, refer to Mahapatra et al. (2023).

Evaluation: Following the evaluation strategies proposed by Mahapatra et al. (2023), we use the standard metrics for machine translation, such as BLEU (Bi-Lingual Evaluation Understudy), GLEU (Google BLEU) and chrF++. For all metrics, the IndicNLP tokenizer is first used to tokenize the texts in Indian languages. For BLEU and chrF++, we use the *SacreBLEU* package (https://pypi.org/project/sacrebleu/). In chrF++ calculation, the default order of character and word n-grams are set to 6 and 2 respectively. For GLEU, we use the *Huggingface evaluate* library for computation, and consider subsequences containing 1,2,3 and 4 tokens (https://huggingface.co/spaces/evaluate-metric/google_bleu).

B Tasks Models, Experiments and Results

In this section, we provide details for all baseline and SOTA models used for each of the tasks. Apart from these methods, we also conduct inference experiments with LLMs across all of these tasks except PCR, which we discuss in App. C.

B.1 Legal Named Entity Recognition (L-NER)

We perform NER based on token representations generated by BERT-based models. Since each document in the dataset does not come pre-segmented into sentences or paragraphs, we need to chunk documents before passing them to BERT, as case documents easily exceed the token limits of BERT. However, unlike other tasks like text classification, we need to devise a chunking strategy to avoid splitting true NEs into different chunks. For this, we choose to chunk at the last stopword (based

Method	Strict			Ent type		
	mP	mR	mF1	mP	mR	mF1
BERT	38.95	41.12	39.59	47.70	49.99	48.23
LegalBERT	43.98	48.06	45.58	53.19	58.33	55.21
CaseLawBERT	42.68	43.68	42.45	52.40	53.48	52.00
InLegalBERT	**47.83**	**50.33**	**48.58**	**57.45**	**60.40**	**58.30**
InCaseLawBERT	45.59	44.59	44.17	56.38	54.89	54.41

Table 15: Performance of BERT-based models over the L-NER dataset. All values are macro-averaged and in terms of percentage.

on NLTK's list of English stopwords), which satisfies the chunk size limit. The assumption is that these stopwords are not expected to be part of entity names.

We experiment with five different BERT encoders: (i) `bert-base-uncased` (Devlin et al., 2019), (ii) LegalBERT (Chalkidis et al., 2020), (iii) CaseLawBERT (Zheng et al., 2021), (iv) InLegalBERT (Paul et al., 2023) and (v) InCaseLawBERT (Paul et al., 2023). We applied a Conditional Random Field (CRF) on top of the BERT encoder due to the efficacy of CRFs in sequence labeling tasks.

Hyper-parameter Settings: We set the chunk limit to 512 tokens to maximize the input capability of BERT. We trained on a single Nvidia RTX A6000 (48 GB). We used a batch size of 40 during training and 24 during testing. The models were trained for a maximum of 20 epochs with early stopping. We used different learning rates for the different layers, viz., 3e-5 for the BERT layers and 1e-3 for the fully connected and CRF layers. We have used the PyTorch implementation of CRF provided in `https://pypi.org/project/pytorch-crf/`.

Model Result and Analysis: Since the dataset is small, we divide the 105 documents into three folds (by trying to maintain the class label frequency distribution across folds as much as possible). We perform 3-fold cross-validation and report the mean across folds. In addition to the strict scores, we also consider another type of scoring, called *ent-type* score (Segura-Bedmar et al., 2013). This scheme considers a match correct if the predicted label type is the same as that of the ground truth, *even if* the predicted span is not correct. Naturally, this scheme is more lenient than the strict mechanism. We report both strict and ent-type scores for all models in Table 15.

In terms of F1 scores, all the models perform relatively poorly. The L-NER dataset contains entire case documents, and evaluation is done over *every occurence* of every named entity. This means that models cannot always rely on the local context to infer the nature of an entity, and all these models are incapable of long range context modeling since the inputs are chunked before feeding to them. This could be a possible reason for the low results. For every model, the ent-type scores are around 20% higher than the strict scores, suggesting that these models also struggle to identify the NE boundaries correctly on quite a few occasions. Comparing among the models, we observe increasing performance with greater degree of domain familiarization. BERT performs the poorest, followed by LegalBERT and CaseLawBERT (which have been pre-trained on legal data from other countries). Counterparts for these models pre-trained on Indian legal text, viz., InLegalBERT and InCaseLawBERT, further outperform them.

Label Analysis: To further analyze the performance across different labels, we calculate the strict and ent-type F1 scores of every label of the best-performing model, InLegalBERT.

Labels like WITNESS, A.COUNSEL, and R.COUNSEL are straightforward to identify, possibly due to the presence of linguistic cues like "P.W." (abbreviated for "Prosecution Witness") and "learned counsel for the appellant/respondent" close to the entity mentions. Labels like COURT, AUTHORITY, and DATE are slightly more challenging to identify due to the large degree of variations possible in the way these entities are mentioned, e.g., "Delhi High Court" vs. "High Court of Judicature at New Delhi", or "14.06.2023" vs. "14/6/23" vs. "14th June 2023". We also observe very little difference in these classes' strict and ent-type scores.

Labels like APPELLANT, RESPONDENT, and JUDGE are more challenging to identify. There is an apparent confusion between APPELLANT and RESPONDENT roles since the entities belonging to these classes usually occur in the same context and play the same role in the court case (just opposing sides). However, the performance of the JUDGE class is lower, although JUDGE type entities are usually enclosed by prefixes such as "Honourable Justice" or suffixes such as "J.". The considerable difference in the strict and ent-type scores for the JUDGE class indicates that the model fails to detect the spans properly rather than the class.

Finally, for labels like STAT, PREC, and CASE NO., the spans can be challenging to identify even

Label	APP	RESP	JUD	AC	RC	CRT	AUTH	WIT	STAT	PREC	DATE	CN	Macro
Strict F1	22.72	11.70	57.33	61.27	53.32	69.16	44.37	29.32	63.45	36.99	81.52	51.76	48.58
Ent-type F1	34.14	18.01	71.30	67.18	58.80	76.09	50.13	34.21	72.43	64.49	85.06	67.73	58.30

Table 16: The table shows the results for each of the NER label.

for human readers since these entities are usually long, occur in multiple forms, and can have extended suffixes. For example, STAT can either be in the full form, such as “Indian Penal Code, 1860” or its abbreviated version “I.P.C.” or “I.P.C.”, while PREC entities can sometimes contain the case number of the particular precedent as a suffix. The considerable differences in strict and ent-type scores of these entities also point to this possibility.

B.2 Rhetorical Role Prediction (RR)

For the task of RR Prediction, we experiment with different approaches, such as passing each sentence individually to BERT and LegalBERT or applying hierarchical approaches to model the entire document together, such as BiLSTM-CRF with sent2vec (Gupta et al., 2019) or BERT embeddings. Malik et al. (2022) suggest an auxiliary task, Label Shift Prediction (LSP), which aims to predict, for sentence i in a document, whether the label changed from sentence $i-1$ to i. This is based on the intuition that RRs tend to maintain some inertia when going from one sentence to another, and changes in RR labels are not abrupt but smooth. BERT-SC is obtained by fine-tuning BERT for the LSP task *only* over the train set of the RR dataset. Finally, the Multi-task Learning (MTL) approach incorporates both RR (main task) and LSP (auxiliary task) prediction. For more details about LSP and MTL, check Malik et al. (2022).

Model	IT	CL	IT+CL
BERT	0.56	0.52	0.54
LEGAL-BERT	0.55	0.53	0.52
BiLSTM-CRF (sent2vec)	0.59	0.61	0.60
BiLSTM-CRF (BERT emb)	0.63	0.63	0.63
LSP(BERT-SC)	0.65	0.68	0.67
MTL(BERT-SC)	**0.70**	**0.69**	**0.70**

Table 17: RR Task Results: Macro-F1 values for both CL and IT datasets.

Results and Analysis: RR prediction is a challenging task; standard transformer-based models like BERT and Legal-BERT do not perform well. Posing the task as a sequence labeling problem, the hierarchical models employing BiLSTM-CRF show improvements. LSP plays a significant role in improving performance, which is seen in the performance of LSP(BERT-SC) over models that do not employ LSP. Harnessing the power of learning both RR and LSP prediction in an end-to-end setup, the MTL model performs the best. However, this is still quite far from human annotations, pointing towards significant scope for improvement.

B.3 Court Judgment Prediction with Explanation (CJPE)

We use the `ILDC-multi` split for judgment prediction and `ILDC-expert` for explanations. Different transformer-based models (BERT, RoBERTa and XLNet) have been tried for the CJPE task. Since these models cannot accommodate large documents, one approach is to make the prediction based on a chunk of 512 tokens. The last 512 tokens are chosen since these parts of the text are likely to contain more information for guiding the final decision (Malik et al., 2021). In other settings, a hierarchical approach is adopted by chunking the entire document into chunks of 512 tokens, passing these to the transformer, and collecting the [CLS] embeddings to be fed to a high-level encoder, such as BiGRU or BiGRU coupled with attention.

For the explanation part, an occlusion method is used by Malik et al. (2021). The primary idea behind this is to mask a chunk of text and then see the change in prediction probability. The prediction probability change indicates the salience of that particular chunk for making the prediction. The more the change in probability, the more salient the chunk.

Results and Analysis: From Table 18, it is evident that the hierarchical models perform better than their counterparts that take just the last 512 tokens (and thus suffer from loss of information). While adding the attn. The layer to the BiGRU module seems to help BERT and RoBERTa slightly, but the same is not true for XLNet. Overall, XLNet + BiGRU performs the best among these approaches.

The occlusion approach for extracting explanations can give positive or negative scores to each chunk; we choose the chunks that obtain positive

Model	Macro Precision (%)	Macro Recall (%)	Macro F1 (%)	Accuracy (%)
BERT	69.33	67.31	68.31	67.24
RoBERTa	72.25	71.31	71.77	71.26
XLNet	72.09	70.07	71.07	70.01
BERT + BiGRU	70.98	70.42	70.69	70.38
RoBERTa + BiGRU	75.13	74.30	74.71	74.33
XLNet + BiGRU	**77.80**	**77.78**	**77.79**	**77.78**
BERT + BiGRU-attn	71.31	70.98	71.14	71.26
RoBERTa + BiGRU-attn	75.89	74.88	75.38	74.91
XLNet + BiGRU-attn	77.32	76.82	77.07	77.01

Table 18: CJPE Prediction Results: Macro-P, R and F1 scores and accuracy scores for all models

scores. The text from these chunks is concatenated and compared with the expert-annotated chunks (5 different annotations for 5 experts). We only consider sentences ranked 1 or 2 (highly important) by the experts as gold-standard explanations. The best model, XLNet + BiGRU, gives 0.424 Rouge-L score and 0.176 BLEU score averaged across all experts. This demonstrates that explainability is still a big challenge, and the model's understanding of important sentences is quite far off from that of the experts.

Model	Accuracy	F1
IndicBert-First 512	0.73	0.71
IndicBert-Last 512	0.78	0.76
TF-IDF+IndicBert	**0.82**	**0.81**
TextRank+IndicBert	0.82	0.81
Salience Pred.+IndicBert	0.80	0.78
Multi-Task	0.80	0.78

Table 19: BAIL Task Results: Accuracy and macro-F1 scores for all models.

B.4 Bail Prediction (BAIL)

We use the `HLDC-all-districts` (Kapoor et al., 2022) split for all our experiments. For BAIL, we used the multi-lingual IndicBERT (Kakwani et al., 2020) to encode the facts and predict. Since the facts can be long, some unsupervised summarization-based approaches (such as TF-IDF ranking and TextRank) have been tried to shorten the inputs and remove noise. We also experiment with the salience prediction approach demonstrated by Kapoor et al. (2022) that aims to predict the important sentences via supervised learning of salience scores (the gold standard scores are decided by comparing each fact sentence with the final case summary written by the judge). Finally, we also an MTL approach by combining BAIL and salience prediction tasks is also carried out.

Results and Analysis: The results are reported in Table 19. As we observe, summarization of the input facts is a better approach than just taking the first or last 512 tokens for passing to IndicBERT. Surprisingly, TF-IDF shows the best performance with 81% macro-F1, even outperforming supervised salience prediction and MTL approaches. This could possibly be because of the large variation in the nature and dialect of text across the entire dataset.

B.5 Legal Statute Identification (LSI)

We chose some models from the BERT family – LegalBERT (Chalkidis et al., 2020) and InLegalBERT (Paul et al., 2023) as baselines for this task. Since fact descriptions (input for LSI) can be long, they may not fit within the maximum 512-token limit for BERT encoders, necessitating a hierarchical model. Examples from the ILSI dataset are pre-segmented into sentences. We pass each sentence individually through the BERT encoder and gather the [CLS] embeddings for each document. The sequence of [CLS]-embeddings are passed through an upper Bi-LSTM layer coupled with attention, yielding a single representation for the entire fact portion. It then passes through a fully connected layer with sigmoid activation to obtain label probabilities. Labels with a probability score > 0.5 are considered relevant. Apart from these two models, we also experiment with LeSICiN (Paul et al., 2022), a graph-based deep neural model that also utilizes sent2vec (Gupta et al., 2019) embeddings pre-trained on Indian legal data.
Results: The results are reported in Table 20. All models perform poorly, indicating the challenging nature of the ILSI dataset. Among the BERT-based methods, InLegalBERT outperforms LegalBERT since the former has been trained on Indian legal documents and is likely to have more inherent domain knowledge. While the BERT-based methods

Encoder Module	mP	mR	mF1
LegalBERT + LSTM-Attn	53.79	15.72	21.74
InLegalBERT + LSTM-Attn	**58.75**	19.29	26.23
LeSICiN	24.34	**36.58**	**28.08**

Table 20: Performance over the ILSI dataset for LSI. All reported values are macro-averaged and in terms of percentage.

utilize strong contextual representations to identify patterns in the fact text that highly correlate with certain labels (high precision), the low recall suggests that the model is not able to pick up more latent patterns. On the other hand, LeSICiN shows a comparatively better recall since it compares the fact text with the text of the statutes via a graph neural network but has poor precision. Overall, LeSICiN still manages to outperform the BERT-based methods.

B.6 Prior Case Retrieval

For the PCR task, a classical IR baseline BM-25, apart from some transformer-based approaches, is chosen. We follow the baselines proposed in (Joshi et al., 2023) and perform all the experiments, including the ones where a document is converted to a set of events.

Results and Analysis: The results are shown in Table 21. BM-25 seems to be a strong baseline, and BERT-based models fail to outperform this. In fact, the scores of transformer-based approaches are surprisingly low (less than 10% F1). Instead, the event-filtered doc approach works the best. Comparing the two event-based approaches, working directly with the atomic events works better for BM25 approaches with unigrams and bigrams, but for trigram onwards, the event-filtered doc approach outperforms this.

We have observed that the event-based models perform the best but still have a micro F1 score of 39.15, which is relatively low. Given the low scores, there is massive scope for developing better models for PCR.

B.7 Summarization (SUMM)

Although the IN-Abs dataset is meant for abstractive summarization, we can apply both extractive and abstractive methods (Shukla et al., 2022).

(i) **Extractive methods:** We try out approaches like CaseSummarizer (Polsley et al., 2016) (legal-specific, unsupervised), DSDR (He et al., 2012) (open domain, unsupervised), Gist (Liu and Chen, 2019) (legal-specific, supervised) and SummaRuNNer (Nallapati et al., 2017) (open domain, supervised). To adapt the abstractive gold-standard summaries for these extractive methods, we use the technique suggested by Narayan et al. (2018).

(ii) **Fine-tuned Abstractive methods:** We try out text generation models both from the open-domain like BART (Lewis et al., 2019), and legal domain like Legal-Pegasus (HuggingFace, b) and Legal-LED (HuggingFace, a). While Legal-LED can accommodate a large number of documents (16,384 token limit), the same is not true for the other models. To overcome this problem, we chunk the document into equal-sized chunks (each chunk size is lesser than the model length limit) and pass each chunk through the model. The summaries for each chunk are concatenated to form the final summary. To convert the overall document summary (gold standard) into chunk-wise summaries, we follow the approach given by Gidiotis and Tsoumakas (2020). All the models were fine-tuned on the summarization dataset.

Model Result and Analysis The results of all approaches are reported in Table 22. SummaRuNNer performs the best among the extractive approaches across three of the four metrics considered (Rouge-1 & 2, and BERTScore). The abstractive approaches show a general improvement over the extractive ones, possibly due to the gold-standard summaries also being abstractive. Despite being open-domain and requiring chunking, the BART model still comes close to or outperforms Legal-LED across different legal domain-specific metrics and can accommodate very long documents. Legal Pegasus beats BART in terms of R-2 and R-L but falls short in terms of R-1. Legal-LED outperforms every other model in terms of BERTScore.

B.8 Legal Machine Translation (L-MT)

For this task, we employed a host of systems, including Commercial systems such as Google Cloud Translation - Advanced Edition (v3) system[1] (**GOOG**) and the Translation API offered by Microsoft Azure Cognitive Services (v3)[2] (**MSFT**). We also used open-source models such as IndicTrans, which is a transformer-4x based multilingual

[1] https://cloud.google.com/translate/docs/samples/translate-v3-translate-text

[2] https://azure.microsoft.com/en-us/products/cognitive-services/translator

Model		K	Precision	Recall	F1
Word Level	BM25	5	17.11	11.64	13.85
	BM25 (Bigram)	7	29.30	27.91	28.59
Segmented-Doc Transformer (full document)	BERT	6	10.28	8.40	9.24
	BERT (finetuned)	6	8.79	7.18	7.90
	DistilBERT	7	17.02	16.21	16.61
	DistilBERT (finetuned)	5	9.70	6.60	7.86
	InCaseLawBERT	11	3.02	4.52	3.62
	InLegalBERT	12	6.10	9.96	7.56
Atomic Events	Jaccard Similarity	7	35.12	33.28	34.17
	BM25	7	37.69	35.90	36.77
	BM25 (Bigram)	6	35.39	28.89	31.81
	BM25 (Trigram)	6	30.71	25.07	27.61
Events Filtered Docs	BM25	5	24.26	16.50	19.64
	BM25 (Bigram)	6	33.69	27.50	30.28
	BM25 (Trigram)	6	**41.35**	33.76	37.17
	BM25 (Quad-gram)	7	40.12	**38.22**	**39.15**
	BM25 (Penta-gram)	7	39.57	37.70	38.61

Table 21: PCR Task Results: The table shows the K values, Precision, Recall, and F1 scores for each model.

Algorithm	ROUGE Scores			BERTScore
	R-1	R-2	R-L	
Extractive Methods (U: Unsupervised, S: Supervised)				
DSDR (U)	0.485	0.222	0.270	0.848
CaseSummarizer (U)	0.454	0.229	0.279	0.843
SummaRunner (S)	0.493	**0.255**	0.274	0.849
Gist (S)	0.471	0.238	0.308	0.842
Abstractive Methods				
BART	**0.495**	0.249	0.330	0.851
Legal-Pegasus	0.488	0.252	**0.341**	0.851
Legal-LED	0.471	0.235	0.332	**0.856**

Table 22: Results of Summarization Task: Document-wide ROUGE-L and BERTScores (Fscore) on the IN-Abs dataset, averaged over the 100 test documents.

NMT model[3] trained over the *Samanantar* dataset for translation among Indian languages (Ramesh et al., 2022).

Model Result and Analysis The performances of all the MT systems across the 3 datasets are presented in Table 23. We find that no single model performs the best in all scenarios. MSFT, GOOG, and IndicTrans are the 3 best models that generally perform the best in most scenarios. The scores for **MILPaC-Acts** are consistently lower than those for other datasets. This is expected since **MILPaC-Acts** has very formal legal language, which is challenging for all MT systems. Interestingly, though MSFT and GOOG perform the best over most datasets, IndicTrans performs better over **MILPaC-Acts** for several Indian languages (e.g., Malayalam & Gujarati). The superior performance of IndicTrans over **MILPaC-Acts** may stem partly from the fact that it was trained on some legal documents from Indian government websites (such as State Assembly discussions) according to Ramesh et al. (2022). However, it is *not* known publicly over what data commercial systems such as GOOG and MSFT are trained. By looking at the average scores across all 3 datasets and language pairs (see Table 24), we can establish that MSFT performs the best across all metrics.

[3]https://github.com/AI4Bharat/indicTrans

C Additional Experiments with LLMs

The wide generalization capability of large language models has shown tremendous performance across various Natural Language Understanding (NLU) tasks. To validate if the available LLMs generalize enough to domain-specific legal language, we perform a detailed set of experiments by prompting LLMs over the set of proposed tasks in **IL-TUR**. We design prompts based on the available task, the context length, and prior knowledge required for the task, like label definition, which is specific to the legal domain. In recent years, In-Context Learning (ICL) (Brown et al., 2020) has significantly improved LLMs performance on various tasks. Considering the performance boost due to the ICL prompt template, it becomes crucial to consider few-shot prompts. For our experiments with LLMs, we design a prompt template that is

EN → IN	Model	MILPaC-IP			MILPaC-Acts			MILPaC-CCI-FAQ		
		BLEU	GLEU	chrF++	BLEU	GLEU	chrF++	BLEU	GLEU	chrF++
EN → BN	GOOG	27.7	30.7	56.8	12.0	17.0	40.7	**52.0**	**53.6**	**74.8**
	MSFT	**31.0**	**33.8**	**59.4**	18.4	**23.1**	**45.6**	36.5	40.4	66.2
	IndicTrans	24.7	27.3	51.7	**18.6**	21.8	45.5	20.9	25.6	50.2
EN → HI	GOOG	36.6	35.3	53.8	21.2	26.7	47.1	46.0	48.4	67.3
	MSFT	**38.5**	**37.0**	**54.9**	**46.4**	**48.9**	**67.3**	45.5	48.2	**67.5**
	IndicTrans	27.0	28.1	45.1	45.7	48.2	66.6	**49.1**	**49.8**	67.1
EN → TA	GOOG	**39.3**	**41.8**	**69.4**	8.1	13.7	37.0	**41.4**	**44.0**	**70.7**
	MSFT	35.3	38.7	68.8	**12.1**	**17.6**	**46.3**	29.5	33.7	64.9
	IndicTrans	21.4	25.5	51.9	11.1	16.7	43.7	22.9	26.8	56.1
EN → MR	GOOG	**23.0**	**25.6**	**51.6**	8.6	14.6	37.5	**51.3**	**53.0**	**74.8**
	MSFT	19.4	22.8	49.6	**13.9**	**19.6**	**45.0**	34.1	38.3	65.8
	IndicTrans	16.0	19.6	44.0	12.9	18.5	42.1	28.2	32.0	56.7
EN → TE	GOOG	**22.4**	**23.2**	**48.9**	6.6	11.4	28.8	-	-	-
	MSFT	15.8	18.3	44.8	**12.0**	**16.9**	39.4	-	-	-
	IndicTrans	15.5	17.6	40.6	11.9	16.8	**40.4**	-	-	-
EN → ML	GOOG	22.3	27.7	57.5	7.3	12.4	32.2	-	-	-
	MSFT	**34.2**	**37.7**	**66.5**	10.8	17.0	46.2	-	-	-
	IndicTrans	19.8	24.5	48.9	**16.6**	**21.2**	**50.3**	-	-	-
EN → PA	GOOG	17.8	20.8	41.3	8.9	14.1	28.6	-	-	-
	MSFT	**30.2**	**30.5**	**51.3**	**40.1**	**42.4**	**62.5**	-	-	-
	IndicTrans	28.1	28.8	47.6	24.0	28.8	48.8	-	-	-
EN → OR	GOOG	2.4	6.5	29.0	4.1	8.2	26.3	-	-	-
	MSFT	5.5	9.0	33.7	7.6	13.3	37.3	-	-	-
	IndicTrans	4.9	8.6	30.5	8.9	15.0	**40.4**	-	-	-
EN → GU	GOOG	43.6	46.0	67.8	14.3	19.5	42.1	-	-	-
	MSFT	**47.3**	**49.2**	**70.6**	21.7	26.1	**51.9**	-	-	-
	IndicTrans	31.3	34.9	56.3	**22.9**	**27.0**	50.9	-	-	-
Average	GOOG	26.1	28.6	47.6	10.1	15.3	35.6	**47.7**	**49.8**	**71.9**
	MSFT	**28.6**	**30.8**	**55.5**	20.3	25.0	49.1	36.4	40.2	66.1
	IndicTrans	24.4	27.8	52.5	**21.7**	**26.8**	**53.3**	30.3	33.6	57.5

Table 23: Corpus-level BLEU, GLEU, and chrF++ scores for all MT systems, over three datasets. All values are averaged over all text pairs in a particular dataset. For each dataset and each English-Indian language pair, the best value of each metric is boldfaced.

Model	BLEU	GLEU	chrF++
GOOG	28.0	31.2	51.7
MSFT	**28.4**	**32**	**56.9**
IndicTrans	25.5	29.4	54.4

Table 24: Corpus-level BLEU, GLEU, and chrF++ scores for all MT systems. All values are averaged over all text pairs, across all languages, and across 3 datasets.

compatible with ICL, i.e., the same prompt template can be used to provide a few shot examples as a prompt to the language models. Primarily, we validate the performance of large proprietary LLMs as well as smaller non-commercial LLMs. As some of the tasks require the entire document to be a part of the model's input, evaluating the entire test sets becomes more challenging and time-consuming for tasks with large test sets. Since the primary design of the benchmark is not LLM specific, we perform the LLM validation to obtain a general proxy of LLM performance.

C.1 Experiments with Proprietary LLMs

For experiments with proprietary LLMs, we consider the widely used OpenAI's ChatGPT (`gpt-3.5-turbo-16k` version). This version is available to use and can accommodate an input length of up to 16k tokens, making it suitable for all tasks in IL-TUR except PCR. As explained in §4, PCR requires as input the texts of the source document as well as a set of candidate documents. Due to the size of legal documents, such a setup would exceed 16k tokens. We discuss the task-specific prompt design and evaluation strategies and the obtained findings in the subsections below. Table 25 shows the results for various tasks.

Task	#samples	0-Shot	1-Shot	2-Shot	Metric
L-NER	35	30.59%	23.68%	32.84%	strict mF1
RR	10	30.95%	30.05%	30.31%	mF1
CJPE	100 (Pred.) 56 (Explan.)	54.17% 0.39 0.019	51.46% 0.29 R-L 0.03	56.74% 0.36 0.03	mF1 R-L BLEU
BAIL	100	51.04%	46.35%	61.0%	mF1
LSI	100	21.55%	22.61%	21.43%	mF1
SUMM	100	0.27 0.85	0.16 0.83	0.19 0.85	R-L BERTScore
L-MT	110	0.23 0.28 0.42	0.25 0.28 0.43	0.26 0.29 0.43	BLEU GLEU chrF++

Table 25: Performance of Open-AI-GPT (gpt-3.5-turbo-16k) model on various tasks for zero-shot, one-shot and two-shot settings. R-L refers to ROUGE Longest Common Subsequence. The second column shows the number of samples chosen for LLM inference experiments.

C.1.1 Legal Named Entity Recognition (L-NER)

Prompt Design: Although the NER task is known to ChatGPT, LNER involves clearly understanding the meaning of the legal entities. Thus, we provide descriptions of the entities as part of our prompt (Table 26).

Data Selection: As discussed in App. B.1, we divided our entire data into 3 folds for testing the other models. In this experiment, we only choose the documents of one particular fold (Fold 1) for passing to GPT. For in-context learning, we randomly sample documents from Fold 2. In some cases, especially for 2-shot prompting, the input did not fit within 16k tokens even after choosing the shortest in-context (IC) examples. In these cases, we split the document into chunks, passed each chunk to the model along with IC examples, and collated the outputs from each chunk to produce the final output.

Verbalization: We expect the model's output to be precisely compatible with JSON. The generated JSON format was sometimes incomplete, and we used string processing to complete these strings for JSON compatibility.

Results: GPT returns a list of entities for each class. We mapped all character spans in the document corresponding to each entity and used these character span mappings to generate the BIO sequence used for evaluation. The results for the GPT are mentioned in Table 27. In terms of the strict scores, GPT performs much poorly compared to the SOTA models, demonstrating that it cannot understand the legal roles clearly without any fine-tuning. Observing the 1 and 2-shot results,

```
SYSTEM_PROMPT: You are a smart and intelligent Named Entity Recognition (NER) system. I will provide you with the definition of the entities you need to extract and the output format. I will also provide some examples of the task and the document from where you should extract the entities.
USER_PROMPT: Are you clear about your role?
ASSISTANT_PROMPT: Sure, I'm ready to help you with your NER task. Please provide me with the necessary information to get started.
INPUT_PROMPT:
Entity Definition:
1. APPELLANT: Name or abbreviation of the person(s) or organization(s) filing an appeal/petition to a court of law.
2. RESPONDENT: Name or abbreviation of a person(s) or organization(s) responding/defending to an appeal/petition filed against them in a court of law.
3. JUDGE: Name of the judge/justice presiding over the case in a court of law.
4. APPELLANT COUNSEL: Name of the lawyer representing the appellant/petitioner in a court of law.
5. RESPONDENT COUNSEL: Name of the lawyer representing the respondent in a court of law.
6. COURT: Name of the court of law
7. AUTHORITY: Name or abbreviation of any organization apart from a Court, which has administrative, legal or financial authority. This also includes regulatory and investigative agencies.
8. WITNESS: Name of a person appearing as witness or testifying to a case in a court of law.
9. STATUTE: Name or abbreviation of a statutory law or legal article.
10. PRECEDENT: Title of a prior court case.
11. DATE: Any format of date, even in natural language.
12. CASE NUMBER: Any format of prior case number or order numbers.
Important Instructions:
1. Salutations or prefixes/suffixes like Mr., Mrs., Smt., Justice, J., Dr., P.W., are not part of the named entity.
Output Format:
{"APPELLANT": [list of entities present], "RESPONDENT": [list of entities present], "JUDGE": [list of entities present], "APPELLANT COUNSEL": [list of entities present], "RESPONDENT COUNSEL": [list of entities present], "COURT": [list of entities present], "AUTHORITY": [list of entities present], "WITNESS": [list of entities present], "STATUTE": [list of entities present], "PRECEDENT": [list of entities present], "DATE": [list of entities present], "CASE NUMBER": [list of entities present]}
DO NOT REPEAT THE SAME ENTITY NAME MULTIPLE TIMES.
If no entities are presented in any category, keep an empty list for that category.
The above format should be a pure JSON format.
Examples:
Document 1: <In-context Document 1 goes here>
Output 1: <Gold-standard Labels for Document 1 goes here>
. . .
Document n+1: <Test Document goes here>
Output n+1:
```

Table 26: Prompt template for L-NER (for n in-context examples)

Method	Strict			Ent type		
	mP	mR	mF1	mP	mR	mF1
GPT 0-shot	48.57	24.58	30.59	65.23	**34.46**	**42.04**
GPT 1-shot	39.08	18.56	23.68	56.05	26.73	34.34
GPT 2-shot	**51.29**	**26.16**	**32.84**	**65.63**	32.80	41.54

Table 27: Performance of GPT over the L-NER dataset. All values are macro-averaged and in terms of percentage.

it is clear that providing a single IC example can mislead the model, and adding 2 examples provides a slight improvement over 0-shot. Finally, as observed for the BERT-based models, there is a significant difference between strict and ent-type scores.

C.1.2 Rhetorical Role Labeling (RR)

Prompt Design: The RR task can be considered a semantic role labeling task over the sentences. Such a variant of the task and the definition of the rhetorical roles themselves are probably not clearly known to GPT; hence, we give it explicit guidelines on how to carry out the labeling task. We tried out some initial prompts considering document-level inputs, i.e., passing the entire document (list of sentences) to GPT and asking it to generate a list of labels corresponding to each sentence. This approach had several challenges, such as GPT not returning the same number of labels as input sentences, random token generation, etc. This problem became more pronounced in the ICL setting. Further, input text and sample output for IC examples were becoming too long. Thus, for GPT, we frame the task as a simple sentence classification task, asking the model to predict the label of an individual sentence. The final prompt is shown in Table 28. We run GPT over all sentences in a document to get all corresponding label predictions.

```
SYSTEM_PROMPT: You are a smart and intelligent legal semantic role labeling system. In Indian Court judgment documents, each document sentence can be assigned a legal semantic role. Your task is, given a sentence from an Indian Court case document, to identify the given sentence's semantic role. I will provide you with the descriptions of the legal semantic roles. I will also provide you with some examples.

USER_PROMPT: Are you clear about your role?

ASSISTANT_PROMPT: Absolutely, I understand my role. You would like me to identify a sentence's legal semantic role label in an Indian court case document. Please provide me with the descriptions of the legal semantic roles to help guide me in accurately assigning the role to the given sentence.

INPUT_PROMPT:
Legal Semantic Role Descriptions:
1. Fact: The actual facts and events that led to the case.
2. Argument: Legal arguments which have been put forward by either lawyer.
3. RulingByLowerCourt: Decisions of the lower courts, if any.
4. Statute: References or citations to statutory laws and articles referred in the case.
5. Precedent: Sentences containing References or citations to precedents (prior cases).
6. RatioOfTheDecision: The reasoning which has been established by the judge in the current judgment.
7. RulingByPresentCourt: The final decision of the current court.
ANSWER ONLY WITH ONE OF THE ABOVE CHOICES, DO NOT PROVIDE ANY EXTRA OUTPUT.
Examples:
Sentence 1: <In-context Sentence 1 goes here>
Output 1: <Gold-standard Label for Sentence 1 goes here>
...
Sentence n+1: <Test Sentence goes here>
Output n+1:
```

Table 28: Prompt template for RR (for n in-context examples)

Model	CL	IT	CL + IT
GPT 0-shot	**0.25**	0.37	**0.31**
GPT 1-shot	0.24	0.36	0.30
GPT 2-shot	0.23	**0.38**	0.30

Table 29: Macro-F1 scores for RR datasets

Data Selection: We used all sentences from all documents in the CL and IT test sets (5 documents each). For in-context samples, we randomly choose sentences from all these documents except the document from which the test sentence (sentence for which we expect GPT prediction) is sampled.
Verbalizer: In most cases, GPT answers with the exact label name. In some cases, it can be accompanied by extra erroneous words. In case the prediction is a sequence of words, we iterate over the words and choose the first word that corresponds to an RR. If no such word is found, GPT prediction has failed, and we randomly choose a label to substitute its decision.
Results: The SOTA model achieves a macro-F1 of 70% over the combined (IT + CL) test set. In comparison, GPT can only achieve a macro-F1 of 31%, showing that it is not straightforward for the LLM to assign semantic labels to sentences. On manual inspection, we observed that the model was prone to assign the FAC label to all sentences with the model temperature set to 0. On increasing the temperature to 0.95 (temperature 1 was not giving stable results), we observe that the model is still prone to assigning labels like FAC, ARG-P, ARG-R, and RPC (frequent labels) to most sentences. Also, it seems that ICL has no positive impact on the model. It could be possible that just the description of the labels is not enough; GPT might need 1/2 examples from each class to clearly understand the meaning of the RRs. However, this approach is likely to increase the context length significantly.

C.1.3 Court Judgment Prediction and Explanation (CJPE)

Prompt Design: For the prediction aspect of this task, we ask GPT to read the content of the entire document and predict the final “accept”/“reject” decision (Table 30). For the explanation aspect, we modify the prompt, asking GPT first to predict the accept/reject decision and then extract important sentences of the text that led to its decision (Table 31).
Data Selection: For prediction, we divide the `ILDC-multi` test set into positive and negative ex-

```
SYSTEM_PROMPT: You are a smart and intelligent system, trained to
act like a judge in the Indian Supreme Court. A court case document
in the Indian Supreme Court can consist of one or more appeals by
a particular party. Your task is, given such a case document, to
predict whether the appeals will be accepted or rejected. For cases
containing multiple appeals, you will predict either 'accept' if at
least one of the appeals can be accepted or 'reject' if none of the
appeals can be accepted. PLEASE ANSWER ONLY WITH EITHER 'ACCEPT'
OR 'REJECT'. I will provide you with some examples of this task and
the case document you need to make the prediction for.

USER_PROMPT: Are you clear about your role?

ASSISTANT_PROMPT: Sure, I'm ready to help you with your court
judgment prediction task. Please provide me with the examples
and the case document I'm supposed to make the prediction for.

INPUT_PROMPT:
Examples:
Case Document 1: <In-context Document 1 goes here>
Output 1: <Gold-standard Label for Document 1 goes here>
. . .
Case Document n+1: <Test Document goes here>
Output:
```

Table 30: Prompt template for CJPE Prediction (for n in-context examples)

amples and randomly sample 50 positive and 50 negative examples. For ICL, we randomly sample examples from the remaining test set documents such that the final prompt is within the GPT token limit. For explanation, we use all 56 documents from `ILDC-expert` for prompting. We sample the IC examples from this set itself. The gold standard outputs, in this case, are the important sentences with rank 1 and 2, as per the ranking given by either expert 3 or expert 4, chosen randomly (since these experts had the highest agreement according to Malik et al. (2021)). In both cases, for 2-shot prompting, we sample one document each from the positive and negative classes.

Verbalizer: For prediction, the model always answers with either ACCEPT/REJECT. For explanation, there were a few variations in the output format, but all included a list of the important sentences, marked either with bullet points, numbering, or other delimiters. We used these cues to extract the exact sentences.

Results: For prediction, GPT always tends to predict "reject" in favor of "accept" for all cases. Only by tweaking the temperature up to as high as 0.98, we could observe more "accept" predictions. Despite this, GPT significantly underperforms compared to SOTA approaches, barely performing better than random choice (see Table 32). The result turns even worse with 1-shot prompting, possibly making the model biased towards the class of the IC example. 2-shot prompting gives the best result among these settings.

The explanation is a more difficult task than the prediction, and GPT again underperforms compared to the SOTA approach, especially consid-

```
SYSTEM_PROMPT: You are a smart and intelligent system, trained to
act like a judge in the Indian Supreme Court. A court case document
in the Indian Supreme Court can consist of one or more appeals
by a particular party. Your task is, given such a case document,
to predict whether the appeals will be ACCEPTED or REJECTED. You
will also have to explain your prediction by QUOTING VERBATIM the
important sentences of the input text that led to your decision.
I will provide you with examples of the task, and then the input
document.

USER_PROMPT: Are you clear about your role?

ASSISTANT_PROMPT: Sure, I'm ready to help you with your court
judgment prediction task. I will also quote verbatim the important
areas of the input text that led to my prediction. Please provide
me with the case document I'm supposed to make the prediction for.

INPUT_PROMPT:
IMPORTANT: For explaining your prediction, quote important
sentences verbatim from the input text. DO NOT PARAPHRASE OR
SUMMARIZE THESE SENTENCES.
Examples:
Case Document 1: <In-context Document 1 goes here>
Output 1: The appeals will be <Gold-standard ACCEPT/REJECT Label
for Document 1 goes here>. Here are the verbatim sentences that led
to this decision: <Gold-standard important sentences for Document
1 goes here>
. . .
Case Document n+1: <Test Document goes here>
Output:
```

Table 31: Prompt template for CJPE Explantion (for n in-context examples)

Model	ILDC$_{multi}$			ILDC$_{expert}$	
	mP	mR	mF1	R-L	BLEU
ChatGPT 0-shot	57.14	56.00	54.17	**0.386**	0.019
ChatGPT 1-shot	57.06	55.00	51.46	0.284	**0.027**
ChatGPT 2-shot	**65.79**	**60.42**	**56.74**	0.357	0.026

Table 32: Performance over the L-NER dataset. All reported values are macro-averaged and in terms of percentage.

ering the BLEU score (Table 32). In this case, we observe the positive impact of ICL, although there is a slight dip from 1-shot to 2-shot, possibly because in a 2-shot setting, the context becomes very large due to the added length of important sentences for IC examples.

C.1.4 Bail Prediction (BAIL)

Prompt Design: BAIL is a binary classification task, and in terms of understanding and format, it is very similar to the CJPE task, the only difference being that the HLDC dataset for BAIL contains Hindi text rather than English. We ask ChatGPT to read the application's content and provide the final decision, i.e., if the bail will be granted or dismissed (see Table 33).

```
SYSTEM_PROMPT: You are a smart and intelligent system, trained to act like a judge in a district court of India. Most criminal cases in district courts involve bail applications written in Hindi. The application can be 'granted' if the judge believes the applicant deserves relief or 'dismissed' if the crime is too grave to grant relief. Your task is, given such a bail application, to predict if the bail will be 'granted' or 'dismissed'. PLEASE ANSWER ONLY WITH EITHER 'GRANTED' OR 'DISMISSED'. I will provide you with some examples of this task and the application document you need to make the prediction for.
USER_PROMPT: Are you clear about your role?
ASSISTANT_PROMPT: Sure, I'm ready to help you with your bail application prediction task. Please provide me with the examples and the bail application I'm supposed to make the prediction for.
INPUT_PROMPT:
Examples:
Bail Application 1: <In-context Application 1 goes here>
Output 1: <Gold-standard Label for Application 1 goes here>
...
Bail Application n+1: <Test Application goes here>
Output:
```

Table 33: Prompt template for BAIL Prediction (for n in-context examples)

Data Selection: We divide the HLDC-all-districts test set into positive and negative examples and randomly sample 50 positive and 50 negative examples that can be accommodated in GPT's token length limit. For ICL, we sample examples at random from the rest of the test set. For the 2-shot setting, we always sample one example each from the positive and negative classes.

Verbalizer: The model outputs only GRANTED/DISMISSED, so we directly take the model output as the predicted label.

Results: GPT obtains a much inferior score compared to the SOTA baselines, and the performance is comparable to a random classifier for some settings. Similar to observations for CJPE, we observed that the model was more likely to predict DISMISSED than GRANTED. We adjusted the temperature to 0.95 to achieve some parity in the predicted labels. In ICL, the 1-shot setting actually performs worse than the random. However, a 2-shot produces an improved score.

Model	mP	mR	mF1
ChatGPT 0-shot	52.22	52.00	50.74
ChatGPT 1-shot	46.85	47.00	46.35
ChatGPT 2-shot	**63.37**	**62.00**	**61.00**

Table 34: Macro-averaged scores for BAIL

C.1.5 Legal Statute Identification (LSI)

Prompt Design: The Indian Penal Code (IPC) is already known to GPT since it can accurately answer when asked about the content of different Sections of IPC. In an initial setting, we asked GPT to just output the list of relevant Section numbers of IPC for a given input. We observed that GPT produced hallucinated outputs; in this case, the output often consisted of *non-existent* IPC Section numbers. Now, each IPC Section contains a corresponding title, which is a very short description of the entire statute. In a second setting, we asked GPT to output the section numbers and their corresponding titles. For instance, if, for a particular case, Section 302 of the IPC is relevant, ChatGPT was expected to output just "Section 302" in the first setting, whereas it was expected to answer "Section 302 — Punishment for murder" in the second setting. We observed that this second setting reduced the hallucination to a great extent. The prompt is shown in Table 35.

```
SYSTEM_PROMPT: You are an intelligent Legal Crime Classification system. In the Indian legal system, the Indian Penal Code (IPC) is an Act in the Indian legislature that contains many legal articles or 'Sections' that codify different laws. Your task is, given the facts or evidence of an Indian court case as input, to predict the relevant or violated 'Sections' of the IPC as output. I will provide you some examples of this task and the facts of the case to make predictions for.
USER_PROMPT: Are you clear about your role?
ASSISTANT_PROMPT: Yes, I understand my role as an intelligent Legal Crime Classification system for the Indian legal system. You can provide me with the facts of a court case, and I will identify the relevant or violated sections of the Indian Penal Code (IPC) based on the provided input and output format. Please go ahead and provide me with the examples and the necessary information for the case you'd like me to analyze.
INPUT_PROMPT:
Output Format:
List of relevant Sections and their titles
Examples:
Facts 1: <In-context Facts 1 go here>
Output 1: <Gold-standard Labels for Facts 1 go here>
...
Facts n+1: <Test Facts go here>
Output:
```

Table 35: Prompt template for LSI Prediction (for n in-context examples)

Data Selection: We randomly chose 100 documents (in this case, fact portions) from the ILSI

Model	mP	mR	mF1
ChatGPT 0-shot	21.60	**32.55**	21.55
ChatGPT 1-shot	**27.06**	22.07	**22.61**
ChatGPT 2-shot	25.35	21.53	21.40

Table 36: Macro-averaged scores for ILSI

test set, all of which satisfied the length constraints of GPT. For ICL, we sample other documents from the test set while satisfying the length constraints. Also, for IC examples, we collate the gold-standard Section numbers and their respective titles in the form *Section x — Title of Section x*, create a numbered list, and pass it to GPT.

Verbalizer: Due to the flexibility of the output format, GPT can output a lot of Sections from the IPC and even other acts. We filtered the outputs by considering if either the Section number OR the Section title matched with any of the 100 IPC Section numbers and the corresponding titles of the ILSI candidate statute set. The OR condition was necessary since we observed that even with the second setting, GPT still suffers from the hallucination problem, sometimes providing the correct Section titles with non-existent Section numbers. For instance, consider the GPT output *"Section 1565 of the Indian Penal Code (IPC) - Liability of abettor when one act abetted and different act done"*. This is a hallucinated output since IPC does not have more than 600 Sections. But, the title actually corresponds to a Section in IPC, namely Section 111.

Results: The ILSI dataset is quite challenging, as seen in the SOTA results. In such a comparison, GPT does not perform too badly, as compared to other tasks. ICL does not seem to help too much, with 0,1 and 2-shot settings showing very little difference in results.

C.1.6 Summarization (SUMM)

Prompt Design: GPT is known to be more conversant with the abstractive summarization task. Hence, we provide the model with the summary length limit and ask it to generate the summary (see Table 37). A large majority of the judgments (more than 95%) can be passed as a whole to GPT. For the rest of the (longer) documents, we break the documents into two chunks, summarize each chunk individually, and then append the chunk summaries to get the final summary.

Data Selection: We chose all 100 documents from the test set of In-Abs for passing to ChatGPT. For ICL, we sample from this set of documents itself. We try to sample the smallest samples for the longer input examples to fit the entire prompt within 16k tokens.

Verbalizer: The entire output returned by GPT is considered as the abstractive summary.

Results: GPT results are shown in Table 22. The performance gap between the SOTA approaches and GPT is much lesser as compared to most other tasks. 1 and 2-shot prompting lead to lower performance, possibly because the extended input (full documents along with their summaries) is confusing GPT.

```
SYSTEM_PROMPT: You are a smart and intelligent summarization system,
trained to read and understand Indian court case documents. Your
task is to, given a court case document, summarize the contents of
the document. The summary should contain the important aspects of
the case. I will provide you with some examples of this task and
the document to be summarized.

USER_PROMPT: Are you clear about your role?

ASSISTANT_PROMPT: Sure, I'm ready to help you with your court
judgment summarization task. Please provide me with the examples
and the case document I'm supposed to summarize.

INPUT_PROMPT:
Examples:
Case Document 1: <In-context Document 1 goes here>
Output 1: <Reference summary for Document 1 goes here>
. . .
Case Document n+1: <Test Document goes here>
Output:
```

Table 37: Prompt template for SUMM (for n in-context examples)

Model	R-1	R-2	R-L	BERTScore
ChatGPT 0-shot	**0.466**	**0.221**	**0.271**	**0.852**
ChatGPT 1-shot	0.261	0.102	0.162	0.830
ChatGPT 2-shot	0.305	0.146	0.190	0.846

Table 38: Rouge-1,2,L and BERTScore scores for SUMM

C.1.7 Legal Machine Translation (L-MT)

Prompt Design: GPT is known to perform translations effectively. Hence, we provide the model with just the input sentence (in English), and we ask the model to translate the sentence to the desired target language (see Table 39).

Data Selection: We randomly choose 5 samples from each target language from each MILPaC dataset. This gives us 45 documents each for MILPaC-IP and MILPaC-Acts (9 target languages), and 20 documents for MILPaC-CCI-FAQ (4 target languages), giving us a total of 110 samples. It should be noted that all datasets contain two types of samples – questions and answers. However, the answers from the MILPaC-CCI-FAQ dataset consist of just a single number corresponding to different choices in the MCQ setting. Thus, we do not choose the answer samples from MILPaC-CCI-FAQ. For ICL, we randomly choose samples from the same target language in the same dataset.

```
SYSTEM_PROMPT: You are a smart and intelligent machine translation system, trained to read Indian legal texts and translate them to Indian languages. Your task is, given an English language sentence from a legal document, translate it to the given target Indian language. I will provide you with the input/output format, target language and the sentence to be translated. I will also provide some examples of the task.
USER_PROMPT: Are you clear about your role?
ASSISTANT_PROMPT: Sure, I'm ready to help you with your legal translation task. Please provide me with the sentence and the target language I am supposed to translate to.
INPUT_PROMPT:
Examples:
Sentence 1 in English: <In-context Sentence 1 goes here>
Sentence 1 in <Target language goes here>: <Reference translation for Sentence 1 goes here>
...
Sentence n+1 in English: <Test Document goes here>
Sentence n+1 in <Target language goes here>:
```

Table 39: Prompt template for L-MT (for n in-context examples)

Verbalization: We directly take the entire GPT output as the translation.

Results: GPT produces decent results for L-MT as compared to SOTA approaches, possibly due to GPT's prior knowledge on this task. GPT produces comparable results on the IP and CCI-FAQ datasets, but there is a drop in performance for Acts, possibly due to the more complex nature of the text in the Acts dataset (Mahapatra et al., 2023). We see a gradual improvement across all metrics and all datasets with an increasing degree of ICL, with 2-shot prompting producing the best results.

Dataset	# Shots	BLEU	GLEU	chrF++
MILPac-IP	0	26.2	30.3	45.3
	1	27.8	**31.5**	**46.3**
	2	**27.9**	31.0	45.4
MILPaC-CCI-FAQ	0	24.1	28.2	43.9
	1	25.9	28.7	43.8
	2	**27.9**	**30.6**	**44.9**
MILPaC-Acts	0	18.2	23.1	36.0
	1	19.5	23.6	36.6
	2	**21.2**	**24.8**	**38.2**
Average	0	22.8	28.2	**43.9**
	1	24.4	27.9	42.3
	2	**25.6**	**28.8**	42.8

Table 40: Corpus-level BLEU, GLEU, and chrF++ scores for ChatGPT prompting with 0, 1 and 2 shot settings

C.2 Experiments with Smaller LLMs

In addition, we also experimented with other large language models with smaller parameter sizes. Specifically, we experimented with GPT-Neo (Black et al., 2021) family of three models (GPT-Neo-125M, GPT-Neo-1.3B, GPT-Neo-2.7B) trained on the Pile dataset (Gao et al., 2020), GPT-J-6B(Wang and Komatsuzaki, 2021), Llama-2-7b-chat-hf(Touvron et al., 2023), and recently released Mistral-7B-v0.1(Jiang et al., 2023) language models for our experiments. The primary challenge when validating the smaller language model is the prompt design. Following previous works (Brown et al., 2020; Robinson and Wingate, 2023), we pose the prompt in a multiple-choice question-answering format (a prompt sample for various tasks present in the benchmark can be found in the supplementary material) and validate the performance using the obtained log probability of the predicted tokens as highlighted in (Robinson and Wingate, 2023). Moreover, since the tasks are more complicated with larger context lengths, the generative models sometimes generate some irrelevant tokens. For those cases with random token generation, we consider it to be a failure case and use a random prediction as a proxy of predictions. Overall, we observed that all the language models perform poorly with near-random predictions over the proposed set of legal language understanding tasks.

We speculate two primary reasons for this finding. First, the language models we used are not explicitly designed to capture the question-answering format for a larger context. Since the context length of the task in the proposed benchmark is significantly higher than the other NLU tasks, it becomes more challenging for smaller language models to decode the question-answer format required for performing these tasks. Second, these models lack the instruction tuning strategies followed by larger models like GPT3.5, making it much harder to capture the context. Moreover, our experiments with GPT3.5 also suggest that if the context is large, even the larger models fail to capture the requested instructions present in the query prompt.