ILDC for CJPE: Indian Legal Documents Corpus for Court Judgment Prediction and Explanation

Vijit Malik[1] **Rishabh Sanjay**[1] **Shubham Kumar Nigam**[1]
Kripa Ghosh[2] **Shouvik Kumar Guha**[3] **Arnab Bhattacharya**[1]
Ashutosh Modi[1]
[1]Indian Institute of Technology Kanpur (IIT-K)
[2]Indian Institute of Science Education and Research Kolkata (IISER-K)
[3]West Bengal National University of Juridical Sciences (WBNUJS)
{vijitvm,rsan,sknigam}@iitk.ac.in
kripaghosh@iiserkol.ac.in shouvikkumarguha@nujs.edu
{arnabb,ashutoshm}@cse.iitk.ac.in

Abstract

An automated system that could assist a judge in predicting the outcome of a case would help expedite the judicial process. For such a system to be practically useful, predictions by the system should be explainable. To promote research in developing such a system, we introduce *ILDC (Indian Legal Documents Corpus)*. ILDC is a large corpus of 35k Indian Supreme Court cases annotated with original court decisions. A portion of the corpus (a separate test set) is annotated with gold standard explanations by legal experts. Based on ILDC, we propose the task of Court Judgment Prediction and Explanation (CJPE). The task requires an automated system to predict an explainable outcome of a case. We experiment with a battery of baseline models for case predictions and propose a hierarchical occlusion based model for explainability. Our best prediction model has an accuracy of 78% versus 94% for human legal experts, pointing towards the complexity of the prediction task. The analysis of explanations by the proposed algorithm reveals a significant difference in the point of view of the algorithm and legal experts for explaining the judgments, pointing towards scope for future research.

1 Introduction

In many of the highly populated countries like India, there is a vast number of pending backlog of legal cases that impede the judicial process (Katju, 2019). The backlog is due to multiple factors, including the unavailability of competent judges. Therefore, a system capable of assisting a judge by *suggesting* the outcome of an ongoing court case is likely to be useful for expediting the judicial process. However, an automated decision system is not tenable in law unless it is well explained in terms of how humans understand the legal process. Hence, it is necessary to *explain* the suggestion. In other words, we would like such a system to predict not only *what* should be the final decision of a court case but also *how* one arrives at that decision. In this paper, we introduce INDIAN LEGAL DOCUMENTS CORPUS (ILDC) intending to promote research in developing a system that could *assist* in legal case judgment prediction in an explainable way. ILDC is a corpus of case proceedings from the Supreme Court of India (SCI) that are annotated with original court decisions. A portion of ILDC (i.e., a separate test set) is additionally annotated with gold standard judgment decision explanations by legal experts to evaluate how well the judgment prediction algorithms explain themselves.

Based on ILDC, we propose a new task: COURT JUDGMENT PREDICTION AND EXPLANATION (CJPE). This task aims to predict the *final decision* given all the *facts and arguments* of the case and provide an explanation for the predicted decision. The decision can be either *allowed*, which indicates ruling in favor of the appellant/petitioner, or *dismissed*, which indicates a ruling in favor of the respondent. The explanations in the CJPE task refer to sentences/phrases in the case description that best justify the final decision. Since, we are addressing mainly the SCI cases, one might argue that the usefulness of the task may be limited since, the legislative provisions can always change with time. However, the legal principles of how to apply a given law to a given set of facts remain constant for prolonged periods.

Judgment prediction and explanation in the CJPE task are far more challenging than a standard text-classification task for multiple reasons. Firstly, the legal court case documents (especially

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 4046–4062
August 1–6, 2021.

in Indian context) are unstructured and are usually quite long, verbose, and noisy. There is no easy way of extracting and directly using the facts and arguments. Secondly, the domain-specific lexicon used in court cases makes models pre-trained on generally available texts ineffective on such documents. Consequently, the standard models need to be adapted to the legal domain for the proposed judgment prediction on court cases. Thirdly, explaining prediction in legal documents is considerably more challenging as it requires understanding the facts, following the arguments and applying legal rules, and principles to arrive at the final decision.

Our main contributions can be summarized as:

1. We create a new corpus, INDIAN LEGAL DOCUMENTS CORPUS (ILDC), annotated with court decisions. A portion of the corpus (i.e. a separate test set) is additionally annotated with explanations corresponding to the court decisions. We perform detailed case studies on the corpus to understand differences in prediction and explanation annotations by legal experts, indicative of the computational challenges of modeling the data.
2. We introduce a new task, COURT JUDGMENT PREDICTION AND EXPLANATION (CJPE), with the two sub-tasks: (a) Court Judgment Prediction (CJP) and (b) Explanation of the Prediction. While CJP is not a novel task per se; however, in combination with the explanation part, the CJPE task is new. Moreover, the requirement for explanations also puts restrictions on the type of techniques that could be tried for CJP. In the CJPE task, gold explanations are not provided in the train set; the task expects that the trained algorithms should explain the predictions without requiring additional information in the form of annotations during training.
3. We develop a battery of baseline models for the CJPE task. We perform extensive experimentation with state-of-the-art machine learning algorithms for the judgment prediction task. We develop a new method for explaining machine predictions since none of the existing methods could be readily applied in our setting. We compare model explainability results with annotations by legal experts, showing significant differences between the point of view of algorithms and experts.

ILDC is introduced to promote the development of a system/models that will *augment humans and not replace* them. We have covered the ethical considerations in the paper. Nevertheless, the community needs to pursue more research in this regard to fully understand the unforeseen social implications of such models. This paper takes initial steps by introducing the corpus and baseline models to the community. Moreover, we plan to continue to grow, revise and upgrade ILDC. We release the ILDC and code for the prediction and explanation models via GitHub[1].

2 Related Work

There has been extensive research on legal domain text, and various corpora and tasks have been proposed e.g., prior case retrieval (Jackson et al., 2003), summarization (Tran et al., 2019; Bhattacharya et al., 2019a), catchphrase extraction (Galgani et al., 2012), crime classification (Wang et al., 2019), and judgment prediction (Zhong et al., 2020).

Why ILDC? The task of Legal Judgment Prediction (LJP) and its corresponding corpora (Chalkidis et al., 2019; Zhong et al., 2020; Yang et al., 2019a; Xiao et al., 2018) are related to our setting. In the LJP task, given the *facts* of a case, *violations*, *charges* (e.g., theft) and *terms of penalty* are predicted. However, the ILDC and the CJPE task introduced in this paper differ from the existing LJP corpora and task in multiple ways. Firstly, we require prediction algorithms to explain the decisions in the CJPE task, to evaluate the explanations we provide a separate test set annotated with gold explanations. Secondly, in the LJP task, typically, the facts of a case are explicitly provided. However, in our case, only unannotated unstructured documents are provided. ILDC addresses a more realistic/practical setting, and consequently, CJPE is a much more challenging task. Moreover, the bare facts do not form the judgment premise of a case since facts are subject to interpretations. A court case description, in practice, has other vital aspects like *Ruling by Lower Court*, *Arguments*, *Statutes*, *Precedents*, and *Ratio of the decision* (Bhattacharya et al., 2019b) that are instrumental in decision making by the judge(s). Unlike LJP, we consider (along with the facts) the entire case (except the judgment), and we predict the judgment only. Work by Strickson and de la Iglesia (2020) comes close to our setting, where the authors prepared the test set on UK court cases by removing the final decision from rulings and employed classical machine learning models. Thirdly, to the best of our knowledge,

[1] https://github.com/Exploration-Lab/CJPE

we are the first to create the largest legal corpus (34,816 documents) for the Indian setting. It is important because India has roots in the common law system and case decisions are not strictly as per the statute law, with the judiciary having the discretion to interpret their version of the legal provisions as applicable to the case at hand; this can sometimes make the decision process subjective. Fourth, we do not focus on any particular class of cases (e.g., criminal, civil) but address publicly available generic SCI case documents.

Xiao et al. (2018) released the Chinese AI and Law challenge dataset (CAIL2018) in Chinese for judgment prediction, that contains more than 2.68 million *criminal cases* published by the Supreme People's Court of China. Chalkidis et al. (2019) released an English legal judgment prediction dataset, containing 11,478 cases from the European Court of Human Rights (ECHR). It contains facts, articles violated (if any), and an importance score for each case. ILDC contrasts with the existing LJP corpora, where mainly the civil law system and cases are considered. Though the proposed corpus focuses on Indian cases, our analysis reveals (§ 4.2) that the language used in the cases is quite challenging to process computationally and provides a good playground for developing realistic legal text understanding systems.

Several different approaches and corpora have been proposed for the LJP task. Chalkidis et al. (2019) proposed a hierarchical version of BERT (Devlin et al., 2019) to alleviate BERT's input token count limitation for the LJP task. Yang et al. (2019a) applied Multi-Perspective Bi-Feedback Network for predicting the relevant law articles, charges, and terms of penalty on Chinese AI and Law challenge (CAIL2018) datasets. Xu et al. (2020) proposed a system for distinguishing confusing law articles in the LJP task. Zhong et al. (2018) applied topological multi-task learning on a directed acyclic graph to predict charges like theft, traffic violation, intentional homicide on three Chinese datasets (CJO, PKU, and CAIL). Luo et al. (2017) proposed an attention-based model to predict the charges given the facts of the case along with the relevant articles on a dataset of Criminal Law of the People's Republic of China. Hu et al. (2018) used an attribute-attentive model in a few-shot setup for charge prediction from facts of the case. Long et al. (2019) predicts the decision of the case using a Legal Reading Comprehension technique on a Chinese dataset. Chen et al. (2019) used a deep gating network for prison term prediction, given the facts and charges on a dataset constructed from documents of the Supreme People's Court of China. Aletras et al. (2016) used linear SVM to predict violations from facts on European Court of Human Rights cases. Şulea et al. (2017) used SVM in the LJP task on French Supreme Court cases. Katz et al. (2017) presented a random forest model to predict the "Reverse", "Affirm", and "Other" decisions of US Supreme Court judges. We also experiment with some of these models as baselines for the CJPE task (§ 5).

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 1: ILDC Statistics

Explainability in a system is of paramount importance in the legal domain. Zhong et al. (2020) presented a QA based model using reinforcement learning for *explainable* LJP task on three Chinese datasets (CJO, PKU, and CAIL). The model aims to predict the appropriate crime by asking relevant questions related to the facts of the case. Jiang et al. (2018) used a rationale augmented classification model for the charge prediction task. The model selects as rationale the relevant textual portions in the fact description. Ye et al. (2018) used label-conditioned Seq2Seq model for charge prediction on Chinese legal documents, and the interpretation comprise the selection of the relevant rationales in the text for the charge. We develop an explainability model based on the occlusion method (§ 5.2).

3 Indian Legal Document Corpus

In this paper, we introduce the INDIAN LEGAL DOCUMENTS CORPUS (ILDC), a collection of case proceedings (in the English language) from the Supreme Court of India (SCI). For a case filed at the SCI, a decision ("accepted" v/s "rejected") is taken between the appellant/petitioner versus the respondent by a judge while taking into account the *facts of the case*, *ruling by lower Court(s)*, if any, *arguments*, *statutes*, and *precedents*. For every case filed in the Supreme Court of India (SCI), the judge

(or a bench) decides on whether the claim(s) filed by the appellant/petitioner against the respondent should be "accepted" or "rejected". The decision is relative to the appellant. In ILDC, each of the case proceeding document is labeled with the original decision made by the judge(s) of the SCI, which serve as the gold labels. In addition to the ground truth decision, a separate test set documents are annotated (by legal experts) with explanations that led to the decision. The explanations annotations are ranked in the order of importance.

ILDC Creation. We extracted all the publicly available SCI[2] case proceedings from the year 1947 to April 2020 from the website: https://indiankanoon.org. Case proceedings are unstructured documents and have different formats and sizes, have spelling mistakes (since these are typed during the court hearing), making it challenging to (pre-)process. We used regular expressions to remove the noisy text and meta-information (e.g., initial portions of the document containing case number, judge name, dates, and other meta information) from the proceedings. In practice, as pointed by the legal experts, the judge deciding the case and other meta information influence the final decision. In SCI case proceedings, the decisions are written towards the end of the document. These end section(s) directly stating the decision have been deleted from the documents in ILDC since that is what we aim to predict. Each case's actual decision label has been extracted from the deleted end sections of the proceeding using regular expressions. Another challenge with SCI case proceedings is the presence of cases with multiple petitions where, in a single case, multiple petitions have been filed by the appellant leading to multiple decisions. Consequently, we divided ILDC documents into two sets. The first set, called $\text{ILDC}_{\text{single}}$, either have documents where there is a single petition (and, thus, a single decision) or multiple petitions, but the decisions are the same across all those petitions. The second set, called $\text{ILDC}_{\text{multi}}$, is a superset of $\text{ILDC}_{\text{single}}$ and has multiple appeals leading to different decisions. Predicting multiple different decisions for cases with multiple appeals is significantly challenging. In this paper, we do not develop any baseline computational models for this setting; we plan to address this in future work. For the computational models for the CJPE task, in the case of $\text{ILDC}_{\text{multi}}$, even if a single appeal was accepted in the case having multiple appeals/petitions, we assigned it the label as accepted. Table 1 shows the corpus statistics for ILDC. Note that the validation and test sets are the same for both $\text{ILDC}_{\text{multi}}$ and $\text{ILDC}_{\text{single}}$.

Temporal Aspect. The corpus is randomly divided into train, validation, and test sets, with the restriction that validation and test sets should be balanced w.r.t. the decisions. The division into train, development, and test set was not based on any temporal consideration or stratification because the system's objective that may eventually emerge from the project is not meant to be limited to any particular law(s), nor focused on any particular period of time. On the contrary, the aim is to identify standard features of judgments pronounced in relation to various legislation by different judges and across different temporal phases, to be able to use the said features to decipher the judicial decision-making process and successfully predict the nature of the order finally pronounced by the court given a set of facts and legal arguments. While there would be a degree of subjectivity involved, given the difference in the thoughts and interpretations adopted by different judges, such differences are also found between two judges who are contemporaries of each other, as much as between two judges who have pronounced judgments on similar matters across a gap of decades. The focus is, therefore, to develop a system that would be equally successful in predicting the outcome of a judgment given the law that had been in vogue twenty years back, as it would in relation to the law that is currently in practice. The validity and efficacy of the system can therefore be equally tested by applying it to cases from years back, as to cases from a more recent period. In fact, if the system cannot be temporally independent, and remains limited to only successful prediction of contemporary judgments, then it is likely to fail any test of application because by the time the final version of the system can be ready for practical applications on a large scale, the laws might get amended or replaced, and therefore, the judgments that would subsequently be rendered by the court might be as different from one pronounced today, as the latter might differ from one pronounced in the twentieth century. Not acknowledging time as a factor during data sample choice, therefore, appears to be the prudent step in this case, especially

[2]Although IndianKanoon includes lower court cases as well, they do not have a common structural format and many of the case documents in lower courts may be in a regional Indian language. Hence, for now we only use SCI documents.

given the exponential rate at which legislation is getting amended today, as well as the fast-paced growth of technological development.

Legal Expert Annotations. In our case, the legal expert team consisted of a law professor and his students at a reputed national law school. We took a set of 56 documents ($ILDC_{expert}$) from the test set, and these were given to 5 legal experts. Experts were requested to (i) predict the judgment, and (ii) mark the sentences that they think are explanations for their judgment. Each document was annotated by all the 5 experts (in isolation) using the WebAnno framework (de Castilho et al., 2016). The annotators could assign ranks to the sentences selected as explanations; a higher rank indicates more importance for the final judgment. The rationale for rank assignment to the sentences is as follows. *Rank 1* was given to sentences immediately leading to the decision. *Rank 2* was assigned to sentences that contributed to the decision. *Rank 3* was given to sentences indicative of the disagreement of the current court with a lower court/tribunal decision. Sentences containing the facts of the case, not immediately, leading to decision making, but are essential for the case were assigned *Rank 4 (or lower)*. Note in practice, only a small set of sentences of a document were assigned a rank. Although documents were annotated with explanations in order of ranks, we did not have a similar mechanism in our automated explainability models. From the machine learning perspective, this is a very challenging task, and to the best of our knowledge, none of the state-of-the-art explainability models are capable of doing this. Annotation of explanations is a very specialized, time-consuming, and laborious effort. In the current version of ILDC we provide explanation annotations to only a small portion of the test set, this is for evaluating prediction algorithms for the explainability aspect. Even this small set of documents is enough to highlight the difference between the ML-based explainability methods and how a legal expert would explain a decision (§ 5.3). Nevertheless, we plan to continue to grow the corpus by adding more explainability annotations and other types of annotations. Moreover, we plan to include lower courts like Indian High Court cases and tribunal cases. The corpus provides new research avenues to be explored by the community.

Fairness and Bias. While creating the corpus, we took all possible steps to mitigate any biases that might creep in. We have not made any specific choice with regard to any specific law or any category of cases, i.e., the sampling of cases was completely random. As explained earlier, we took care of the temporal aspect. Importantly, the names of the judge(s), appellants, petitioners, etc., were anonymized in the documents so that no inherent bias regarding these creeps in. The anonymization with respect to judge names is necessary as legal experts pointed out that a judge's identity can sometimes be a strong indicator of the case outcome. It is noteworthy that according to the legal experts if we had not done the same, we could have had higher prediction accuracy. The subjectivity associated with judicial decision-making may also be controlled in this way since the system focuses on how consideration of the facts and applicable law are supposed to determine the outcome of the cases, instead of any individual bias on the judge's part. We also address the ethical concerns in the end.

4 Annotation Analysis

We performed a detailed analysis of case predictions and the explanations annotations. With assistance from a legal expert, we also performed detailed studies for some court cases to understand the task's complexity and possible reasons for deviations between the annotators.

4.1 Case Judgment Accuracy

We computed the case judgment accuracy of the annotators with respect to original decisions by judges of SCI. The results are shown in Table 2. Though the values are high, none of these are 100%. The accuracy indicates that no annotator agrees with the original judgment in all the cases. This possibly depicts the subjectivity in the legal domain with regard to decision making. The subjectivity aspect has also been observed in other tasks that involve human decision-making, e.g., sentiment and emotion analysis. We performed detailed case studies with the help of experts to further probe into this difference in judgment. Due to space limitations, we are not able to present the studies here; please refer to appendix A and GitHub repository for details. To summarize, the study indicated that the sources of confusion are mainly due to differences in linguistic interpretation (by the annotators) of the legal language given in the case document.

4.2 Inter-Annotator Agreements

Agreement in the judgment prediction: For the *quantitative evaluation*, we calculate pair-wise

Expert	Accuracy (%)
Expert 1	94.64
Expert 2	91.07
Expert 3	98.21
Expert 4	89.28
Expert 5	96.43

Table 2: Annotators' accuracy.

Agreement (%)	Expert 1	Expert 2	Expert 3	Expert 4	Expert 5
Expert 1	100.0	87.5	94.6	85.7	89.3
Expert 2	87.5	100.0	92.9	87.5	91.1
Expert 3	94.6	92.9	100.0	91.1	94.6
Expert 4	85.7	87.5	91.1	100.0	89.3
Expert 5	89.3	91.1	94.6	89.3	100.0

Table 3: Pairwise inter-annotator agreement for judgment prediction.

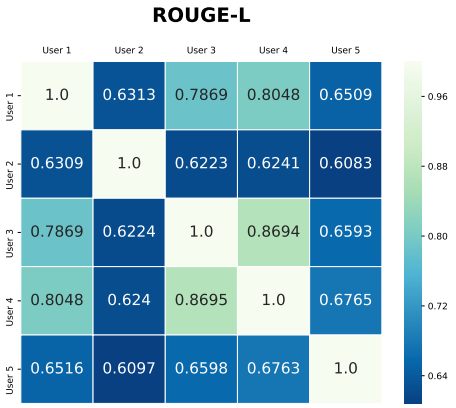

Figure 1: Explanation agreement among the annotators

agreement between the annotators as shown in Table 3. The highest agreement (94.6%) is between Experts 1-3 and 3-5. We also calculate Fleiss' kappa (Fleiss, 1971) as 0.820, among all the five annotators, which indicates high agreement.

Agreement in the explanation: There are no standard metrics for evaluating annotator agreements for textual annotations. For *quantitative evaluation* of agreements among the annotators for explanations, we took inspiration from machine translation community and used metrics like ROUGE-L, ROUGE-1, ROUGE-2 (Lin, 2004), BLEU (Papineni et al., 2002) (unigram and bigram averaging), METEOR (Lavie and Agarwal, 2007), Jaccard Similarity, Overlap Maximum and Overlap Minimum[3]. The result for ROUGE-L (averaged out over all documents)[4] is shown in Figure 1. The highest overlap across all the metrics is observed between Expert 3 and Expert 4. The highest value (0.9129) is between Expert 2 and Expert 4 for Overlap-Min. We also performed a *qualitative evaluation* of the agreements in the explanations. We observed that Expert 1, Expert 3, and Expert 4 consider holistic reasoning for the decision. They look at both Substantive (sections applicable) and Procedural (about the jurisdiction of a lower court) aspects of the case. The differences among them are largely due to consideration/non-consideration of the factual sentences. On the other hand, Expert 2 and Expert 5 often use bare-minimum reasoning leading to the final judgment instead of looking at the exhaustive set of reasons and did not always cover both Substantive and Procedural aspects of the case.

Analysis of annotations gives insights into the inherent complexity and subjectivity of the task. Legal proceedings are long, verbose, often challenging to comprehend, and exhibit interesting (and computationally challenging) linguistic phenomena. For example, in a case numbered "1962_47" (appendix A), sentence 17 of the case appears to refer to the Supreme Court having accepted a previous appeal for which a review has been requested (i.e., the current appeal). This amounted to the fact that the court actually rejected the present appeal while accepting the previous one. Such intricacies can confuse even legal experts.

5 CJPE Task

Given a case proceeding from the SCI, the task of COURT JUDGMENT PREDICTION AND EXPLANATION (CJPE) is to automatically predict the decision for the case (with respect to the appellant) and provide the explanation for the decision. We address the CJPE task via two sub-tasks in the following sequence: Prediction and Explanation.

Prediction: Given a case proceeding D, the task is to predict the decision $y \in \{0, 1\}$, where the label 1 corresponds to the acceptance of the appeal/petition of the appellant/petitioner.

Explanation: Given the case proceeding and the predicted decision for the case, the task is to explain the decision by predicting important sentences that lead to the decision. Annotated explanations are not provided during training; the rationale is that a model learned for prediction should explain the decision without explicit training on explanations, since explanation annotations are difficult to obtain.

[3]Overlap Max: Size of the intersection divided by the maximum size out of the two sample sets that are being compared. Overlap Min: Size of the intersection divided by the minimum size out of the two sample sets that are being compared

[4]Due to space constraints we are not able to show heatmaps corresponding to other metrics but they showed similar trends. For the heatmaps for other metrics please refer to our GitHub repository.

5.1 Case Decision Prediction

ILDC documents are long and have specialized vocabulary compared to typical corpora used for training text classification models and language models. We initially experimented with non-neural models based on text features (e.g., n-grams, tf-idf, word based features, and syntactic features) and existing pre-trained models (e.g., pre-trained word embeddings based models, transformers), but none of them were better than a random classifier. Consequently, we retrained/fine-tuned/developed neural models for our setting. In particular, we ran a battery of experiments and came up with four different types of models: classical models, sequential models, transformer models, and hierarchical transformer models. Table 4 summarizes the performance of different models. Due to space constraints, we are not able to describe each of the models here. We give a very detailed description of model implementations in appendix B.

Classical Models: We considered classical ML models like word/sentence embedding based Logistic Regression, SVM, and Random Forest. We also tried prediction with summarized legal (Bhattacharya et al., 2019a) documents; however, these resulted in a classifier no better than random classifier. As shown in Table 4, classical models did not perform so well. However, model based on Doc2vec embeddings had similar performance as sequential models.

We extensively experimented with dividing documents into chunks and training the model using each of the chunks separately. We empirically determined that sequential and transformer-based models performed the best on the validation set using the last 512 tokens[5] of the document. Intuitively, this makes sense since the last parts of case proceedings usually contain the main information about the case and the rationale behind the judgment. We also experimented with different sections of a document, and we observed last 512 tokens gave the best performance.

Sequence Models: We experimented with standard BiGRU (2 layers) with attention model. We tried 3 different types of embeddings: (i) Word level trained GloVe embeddings (Pennington et al., 2014), with last 512 tokens as input, (ii) Sentence level embeddings (Sent2Vec), where last 150 sentences were input[6], and (iii) Chunk level embeddings (trained via Doc2Vec). We also trained Hierarchical Attention Network (HAN) (Yang et al., 2016) model. GloVe embeddings with BiGRU and

[5]length of 512 was partly influenced by the maximum input token limit of BERT

[6]last 150 sentences covered around 90% of the documents

Model	Macro Precision (%)	Macro Recall (%)	Macro F1 (%)	Accuracy (%)
Classical Models on $ILDC_{multi}$ train set				
Doc2Vec + LR	63.03	61.00	**62.00**	60.91
Sent2vec + LR	57.19	55.55	56.36	55.44
Sequential Models on $ILDC_{multi}$ train set				
Sent2vec + BiGRU + att.	60.98	58.40	59.66	58.31
Doc2vec + BiGRU + att.	57.18	56.03	56.60	57.44
GloVe + BiGRU + att.	68.26	60.87	**64.35**	60.75
HAN	59.96	59.57	59.77	59.53
Sequential Models on $ILDC_{single}$ train set				
Sent2Vec + BiGRU+ att.	60.05	55.8	57.85	55.67
Doc2vec + BiGRU + att.	58.07	57.44	57.75	59.23
GloVe + BiGRU + att.	66.92	62.30	**64.53**	62.2
HAN	57.64	55.56	56.58	55.44
Catchphrases + Sent2Vec + BiGRU + att.	61.90	60.13	61.00	60.06
Transformer Models on $ILDC_{multi}$ train set				
BERT Base	60.56	57.64	59.06	57.65
BERT Base	67.54	62.22	64.77	62.10
BERT Base	67.24	63.85	65.50	63.74
BERT Base	66.12	60.58	63.23	60.45
BERT Base	69.33	67.31	68.31	67.24
DistillBERT	65.21	64.26	64.73	64.21
RoBERTa	72.25	71.31	**71.77**	71.26
XLNet	72.09	70.07	71.07	70.01
Hierarchical Models on $ILDC_{multi}$ train set				
BERT + BiGRU	70.98	70.42	70.69	70.38
RoBERTa + BiGRU	75.13	74.30	74.71 (±0.01)	74.33 (±1.99)
XLNet + BiGRU	77.80	77.78	**77.79**	**77.78**
BERT + CNN	71.68	70.17	70.92	70.12
RoBERTa + CNN	74.74	73.17	73.95	73.22
XLNet + CNN	77.84	77.21	77.53	77.24
Hierarchical Models on $ILDC_{single}$ train set				
BERT + BiGRU	65.28	63.95	64.27 (±0.0116)	63.89 (±1.10)
RoBERTa + BiGRU	73.24	72.93	73.09 (±0.0022)	72.95 (±0.25)
XLNet + BiGRU	75.11	75.06	**75.09 (±0.0043)**	75.06 (±0.42)
Hierarchical Models with Attention on $ILDC_{multi}$ train set				
BERT + BiGRU + att.	71.31	70.98	71.14 (±0.0011)	71.26 (±0.09)
RoBERTa + BiGRU + att.	75.89	74.88	75.38 (±0.0004)	74.91 (±0.11)
XLNet + BiGRU + att.	77.32	76.82	**77.07 (±0.0077)**	77.01 (±0.52)
Hierarchical Models with Attention on $ILDC_{single}$ train set				
BERT + BiGRU + att.	68.30	62.05	65.03 (±0.0084)	61.93 (±0.68)
RoBERTa + BiGRU + att.	73.39	72.66	73.02 (±0.0017)	72.69 (±0.29)
XLNet + BiGRU + att.	75.26	75.22	**75.25 (±0.0009)**	75.22 (±0.13)
Transformers Voting Ensemble				
RoBERTa	68.20	62.55	**65.26**	62.43
XLNet	67.84	60.07	63.72	59.92
Hierarchical concatenated model with attention on $ILDC_{single}$ train				
XLNet + BiGRU	76.85	76.31	**76.55 (±0.0140)**	76.32 (±2.43)

Table 4: Prediction Results using different models. Some of the transformer and hierarchical models vary in performance across runs, we average out performance across 3 runs (variance in the parenthesis).

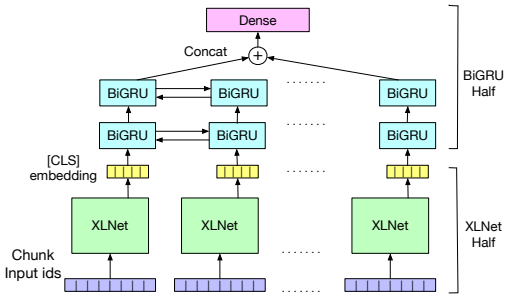

Figure 2: Hierarchical XLNet architecture (XLNet + BiGRU)

attention model gave the best performance (64% F1) among the sequential models. Sequential models trained on $\text{ILDC}_{\text{multi}}$ and $\text{ILDC}_{\text{single}}$ have similar performances

Transformer Models: We experimented with BERT (Devlin et al., 2019), DistilBERT (Sanh et al., 2019), RoBERTa (Liu et al., 2019), and XLNet (Yang et al., 2019b). Due to limitation on the number of input tokens to BERT and other transformer models, we experimented with different sections (begin tokens, middle tokens, end tokens, combinations of these) of the documents and as shown in Table 4, the last 512 tokens gave the best performance. In general, transformer models outperform classical and sequential models. RoBERTa gave the best performance (72% F1) and DistilBERT was the worst. We did not experiment with domain specific transformers like LEGAL-BERT (Chalkidis et al., 2020), since these have been trained upon US/EU legal texts, hence, they do not work well in the Indian setting as the legal systems are entirely different.

Hierarchical Transformer Models: Taking inspiration from hierarchical topic prediction model (Chitkara et al., 2019), we developed Hierarchical Transformer model architecture (Chalkidis et al., 2019). We divided each document into chunks using a moving window approach where each chunk was of length 512 tokens, and there was an overlap of 100 tokens. We obtained the $[CLS]$ representation of these chunks, which were then used as input to sequential models (BiGRU + attention) or feed-forward model (CNN (Kim, 2014)). We also tried an ensemble of individual transformer models on each of the chunks.

In general, all the hierarchical models outperform transformer models. The best performing model (78% F1) for predicting the case decision is XLNet with BiGRU on the top (Figure 2). Comparing best model accuracy with average annotator accuracy (78% vs. 94%) indicates the task's inherent complexity and motivates more research in this direction.

5.2 Case Decision Explanation

We experimented with a variety explainability algorithms as a post-prediction step. We experimented with the best judgment prediction model (Hierarchical Transformer (XLNet + BiGRU)) for all the explainable algorithms. We explored three class of explainability methods (Xie et al., 2020): attribution based, model agnostic, and attention-based.

In the class of attribution based methods, Layerwise Relevance Propagation (LRP) (Bach et al., 2015) and DeepLIFT (Shrikumar et al., 2017) methods did not work in our case. Due to the long length of documents, model agnostic explainability methods like LIME (Ribeiro et al., 2016) and Anchors (Ribeiro et al., 2018) were not applicable. We also experimented with attention-based methods, and Integrated Gradients (Sundararajan et al., 2017) method using the CAPTUM library (Kokhlikyan et al., 2019). However, these highlighted only a few tokens or short phrases. Moreover, attention-based scores are not necessarily indicative of explanations (Jain and Wallace, 2019).

To extract explanations, we propose a method inspired from Li et al. (2016) and Zeiler and Fergus (2014). The idea is to use the occlusion method at both levels of the hierarchy. For each document, for the BiGRU part of the model, we mask each complete chunk embedding one at a time. The masked input is passed through the trained BiGRU, and the output probability (masked probability) of the label obtained by the original unmasked model is calculated. The masked probability is compared with unmasked probability to calculate the chunk explainability score. Formally, for a chunk c, if the sigmoid outputs (of the BiGRU) are σ_m (when the chunk was not masked) and $\sigma_{m'}$ (when the chunk was masked) and the predicted label is y then the probabilities and chunk score $s_c = p_m - p_{m'}$ and

$$p_{m'/m} = \begin{cases} \sigma_{m'/m}, & y = 1 \\ 1 - \sigma_{m'/m}, & y = 0 \end{cases}$$

We obtain sentences that explain the decision from the transformer part of the model (XLNet) using the chunks that were assigned positive scores. Each chunk (length 512 tokens) is segmented into sentences using NLTK sentence splitter (Loper and Bird, 2002). Similar to BiGRU, each sentence is masked and the output of the transformer at the classification head (softmax logits) is compared

Metric	Explainability Model vs Experts				
	Expert				
	1	2	3	4	5
Jaccard Similarity	0.333	0.317	0.328	0.324	0.318
Overlap-Min	0.744	0.589	0.81	0.834	0.617
Overlap-Max	0.39	0.414	0.36	0.35	0.401
ROUGE-1	0.444	0.517	0.401	0.391	0.501
ROUGE-2	0.303	0.295	0.296	0.297	0.294
ROUGE-L	0.439	0.407	0.423	0.444	0.407
BLEU	0.16	0.28	0.099	0.093	0.248
Meteor	0.22	0.3	0.18	0.177	0.279

Table 5: Machine explanations v/s Expert explanations

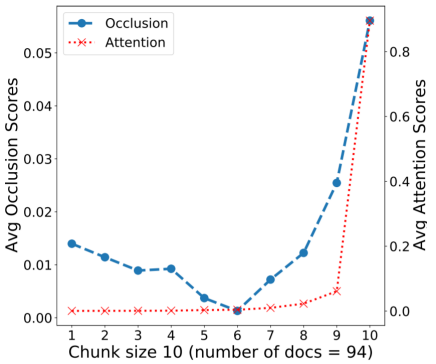

Figure 3: Averaged chunk scores for attention and occlusion

with logits of the label corresponding to original hierarchical model. The difference between the logits normalized by the length of the sentence is the explanation score of the sentence. Finally, top-k sentences ($\sim$ 40%) in each chunk are selected.

To understand and analyze which parts of the documents were contributing towards prediction, we examined the attention weights (scores) in the case of the XLNet+BiGRU+Attention model and the occlusion scores of the XLNet+BiGRU model. Plots for some of the documents are shown in Figure 3. Plots for different chunk sizes are provided in Data/images folder in our GitHub repository. We also provide the t-SNE visualization on the test set using the BERT and Doc2Vec embeddings. Token visualization heatmap using Integrated Gradient for document name 1951_33.txt for BERT model is also provided in GitHub. Plots of scores averaged out over the entire test set for each chunk size can be visualized in appendix B.2. Two things can be noted: firstly, the largest attention and occlusion scores are assigned to chunks corresponding to the end of the document; this is in line with our hypothesis that most of the important information and rationale for judgment is mainly towards the end of the document. Secondly, although attention scores are optimized (via loss minimization or accuracy maximization) to concentrate on the last chunks, this is not the case with occlusion scores. There is no optimization of occlusion scores; yet they still focus on the chunks at the end, which affirms our hypothesis.

5.3 Model Explainability versus Annotators

We compare the performance of occlusion method explanations with the expert annotators' gold explanations by measuring the overlap between the two. We used the same measures (§ 4.2) ROUGE-L, ROUGE-1, ROUGE-2, Jaccard Similarity, BLEU, METEOR, Overlap Maximum, and Overlap Minimum Table 5 compares machine explanations with the gold explanations. The highest overlap value (0.8337) is observed for the measure Overlap-Min with Expert 4. The values for Overlap-Min depict high agreements of the explainability model with all the experts. However, the values for the other evaluation measures, e.g., ROUGE-L, are in the low to medium range, the highest being 0.4445 for ROUGE-L and Expert 4. The results show the wide gap between how a machine would explain a judgment and the way a legal expert would explain it. The results motivate us for future research in this direction of developing an explainable model.

6 Conclusion

This paper introduces the ILDC corpus and corresponding CJPE task. The corpus is annotated with case decisions and explanations for the decisions for a separate test set. Analysis of the corpus and modeling results shows the complexity of legal documents that pose challenges from a computational perspective. We hope that the corpus and the task would provide a challenging and interesting resource for the Legal NLP researchers. For future work, we would like to train a legal transformer similar to LEGAL-BERT (Chalkidis et al., 2020) on our Indian legal case documents. Moreover, we would also like to focus upon using rhetorical roles Bhattacharya et al. (2019b) of the sentences to include structural information of the documents for CJPE task as well.

Acknowledgements

We would like to thank anonymous reviewers for their insightful comments. We would like to thank student research assistants Abin Thomas Alex, Amrita Ghosh, Parmeet Singh, and Unnati Jhunjhunwala from West Bengal National University of Juridical Sciences (WBNUJS) for annotating the documents. This work would not have been possible without their help.

Ethical Concerns

The corpus is created from publicly available data: proceedings of Supreme Court of India (SCI). The data was scraped from the website: www.indiankanoon.org. The website allows scrapping of the data and no copyrights were infringed. Annotators were selected randomly and they participated voluntarily.

The proposed corpus aims to promote the development of an explainable case judgment prediction system. The system intends to assist legal professionals in their research and decision-making and not replace them. Therefore, ethical considerations such as allowing legal rights and obligations of human beings to be decided and pronounced upon by non-human intelligence are not being breached by the system. The system proposes to provide valuable information that might be useful to a legal professional to make strategic decisions, but the actual decision-making process is still going to be carried out by the professional himself. Therefore, the system is not intended to produce a host of artificial lawyers and judges regulating human behavior. At the same time, the final expert human analysis of the systemic output should ensure that any existing flaw, absurdity, or overt or latent bias gets subjected to an additional layer of ethical scrutiny. In this way, the usual ethical concerns associated with the concept of case-law prediction also get addressed to a considerable extent since the system is not performing any judicial role herein nor deciding the legal rights or liabilities of human beings. Instead, the system is purported to be used primarily by legal professionals to make strategic decisions of their own, said decisions being still subjected to legal and judicial scrutiny performed by human experts. Nevertheless, the community needs to pursue more research in this regard to fully understand the unforeseen social implications of such system. This paper takes initial steps by introducing the corpus and baseline models to the community.

Care has been taken to select cases in a completely random manner, without any particular focus on the type of law or the identities or socio-politico-economic background of the parties or the judges involved. Specifically, the aforementioned identities have been deliberately anonymized so as to minimize or eliminate any possible bias in the course of prediction. The subjectivity that is associated with the judicial decision-making may also be controlled in this way, since the system is focusing on how consideration of the facts and applicable law are supposed to determine the outcome of the cases, instead of any individual bias on the judge's part; another judge might not share such bias, and therefore the only common point of reference that the two judges would have would be the relevant facts of the case and the laws involved. This also gets reflected in the objective methodology used in the selection of annotators and by eliminating any interaction between the annotators themselves while at the same time paying attention to the factors or observations common to the output from the various annotators.

The only specification with regard to the forum has been made by taking all the cases from the domain of the Supreme Court of India, owing to the propensity of the apex court of the land towards focusing on the legalities of the issues involved rather than rendering mere fact-specific judgments, as well as the binding nature of such decisions on the subordinate courts of the land. This would also allow the results to be further generalized and applied to a broader set of cases filed before other forums, too, since the subordinate courts are supposed to follow the reasoning of the Supreme Court's judgments to the greatest possible extent. As a result, the impact of the training and testing opportunities provided to the system by a few Supreme Court cases is likely to be much greater than the mere absolute numbers would otherwise suggest.

References

Nikolaos Aletras, Dimitrios Tsarapatsanis, Daniel Preotiuc-Pietro, and Vasileios Lampos. 2016. Predicting judicial decisions of the European Court of Human Rights:A Natural Language Processing perspective. *PeerJ Computer Science*, 2:93.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Paheli Bhattacharya, Kaustubh Hiware, Subham Rajgaria, Nilay Pochhi, Kripabandhu Ghosh, and Saptarshi Ghosh. 2019a. A comparative study of summarization algorithms applied to legal case judgments. In *European Conference on Information Retrieval*, pages 413–428. Springer.

Paheli Bhattacharya, Shounak Paul, Kripabandhu Ghosh, Saptarshi Ghosh, and Adam Wyner. 2019b. Identification of Rhetorical Roles of Sentences in Indian Legal Judgments. In *Legal Knowledge and Information Systems - JURIX 2019*, volume 322 of *Frontiers in Artificial Intelligence and Applications*, pages 3–12. IOS Press.

Richard Eckart de Castilho, Eva Mujdricza-Maydt, Seid Muhie Yimam, Silvana Hartmann, Iryna Gurevych, Anette Frank, and Chris Biemann. 2016. A web-based tool for the integrated annotation of semantic and syntactic structures. In *Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH)*, pages 76–84.

Ilias Chalkidis, Ion Androutsopoulos, and Nikolaos Aletras. 2019. Neural Legal Judgment Prediction in English. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4317–4323, Florence, Italy. Association for Computational Linguistics.

Ilias Chalkidis, Manos Fergadiotis, 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.

Huajie Chen, Deng Cai, Wei Dai, Zehui Dai, and Yadong Ding. 2019. Charge-Based Prison Term Prediction with Deep Gating Network. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6362–6367, Hong Kong, China. Association for Computational Linguistics.

Pooja Chitkara, Ashutosh Modi, Pravalika Avvaru, Sepehr Janghorbani, and Mubbasir Kapadia. 2019. Topic Spotting using Hierarchical Networks with Self Attention. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3755–3761, Minneapolis, Minnesota. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

J. L. Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 75(5).

Filippo Galgani, Paul Compton, and Achim Hoffmann. 2012. Towards automatic generation of catchphrases for legal case reports. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 414–425. Springer.

Zikun Hu, Xiang Li, Cunchao Tu, Zhiyuan Liu, and Maosong Sun. 2018. Few-Shot Charge Prediction with Discriminative Legal Attributes. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 487–498, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

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

Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556, Minneapolis, Minnesota. Association for Computational Linguistics.

Xin Jiang, Hai Ye, Zhunchen Luo, WenHan Chao, and Wenjia Ma. 2018. Interpretable Rationale Augmented Charge Prediction System. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 146–151, Santa Fe, New Mexico. Association for Computational Linguistics.

Justice Markandey Katju. 2019. Backlog of cases crippling judiciary. https://tinyurl.com/v4xu6mvk.

Daniel Martin Katz, Michael J. Bommarito, II, and Josh Blackman. 2017. A general approach for predicting the behavior of the Supreme Court of the United States. *PLOS ONE*, 12:1–18.

Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The Efficient Transformer. In *International Conference on Learning Representations*.

Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Jonathan Reynolds, Alexander Melnikov, Natalia Lunova, and Orion Reblitz-Richardson. 2019. Pytorch Captum. https://github.com/pytorch/captum.

Alon Lavie and Abhaya Agarwal. 2007. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of the second workshop on statistical machine translation*, pages 228–231.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220*.

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

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Shangbang Long, Cunchao Tu, Zhiyuan Liu, and Maosong Sun. 2019. Automatic judgment prediction via legal reading comprehension. In *China National Conference on Chinese Computational Linguistics*, pages 558–572. Springer.

Edward Loper and Steven Bird. 2002. NLTK: the natural language toolkit. *arXiv preprint cs/0205028*.

Bingfeng Luo, Yansong Feng, Jianbo Xu, Xiang Zhang, and Dongyan Zhao. 2017. Learning to Predict Charges for Criminal Cases with Legal Basis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2727–2736, Copenhagen, Denmark. Association for Computational Linguistics.

Arpan Mandal, Kripabandhu Ghosh, Arindam Pal, and Saptarshi Ghosh. 2017. Automatic catchphrase identification from legal court case documents. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 2187–2190.

Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. 2018. Unsupervised Learning of Sentence Embeddings Using Compositional n-Gram features. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 528–540, New Orleans, Louisiana. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why should I trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Anchors: High-Precision Model-Agnostic Explanations. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18)*, pages 1527–1535. AAAI Press.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In *International Conference on Machine Learning*, pages 3145–3153. PMLR.

Benjamin Strickson and Beatriz de la Iglesia. 2020. Legal Judgement Prediction for UK Courts. *ICISS 2020: The 3rd International Conference on Information Science and System, Cambridge, UK, March 19-22, 2020*, pages 204–209.

Octavia-Maria Şulea, Marcos Zampieri, Mihaela Vela, and Josef van Genabith. 2017. Predicting the Law Area and Decisions of French Supreme Court Cases. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 716–722, Varna, Bulgaria. INCOMA Ltd.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR.

Vu Tran, Minh Le Nguyen, and Ken Satoh. 2019. Building Legal Case Retrieval Systems with Lexical Matching and Summarization Using A Pre-Trained Phrase Scoring Model. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law*, ICAIL '19, page 275–282, New York, NY, USA. Association for Computing Machinery.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Pengfei Wang, Yu Fan, Shuzi Niu, Ze Yang, Yongfeng Zhang, and Jiafeng Guo. 2019. Hierarchical Matching Network for Crime Classification. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, pages 325–334.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Chaojun Xiao, Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Zhiyuan Liu, Maosong Sun, Yansong Feng, Xianpei Han, Zhen Hu, Heng Wang, et al. 2018. Cail2018: A large-scale legal dataset for judgment prediction. *arXiv preprint arXiv:1807.02478*.

Ning Xie, Gabrielle Ras, Marcel van Gerven, and Derek Doran. 2020. Explainable deep learning: A field guide for the uninitiated. *arXiv preprint arXiv:2004.14545*.

Nuo Xu, Pinghui Wang, Long Chen, Li Pan, Xiaoyan Wang, and Junzhou Zhao. 2020. Distinguish Confusing Law Articles for Legal Judgment Prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3086–3095, Online. Association for Computational Linguistics.

Wenmian Yang, Weijia Jia, Xiaojie Zhou, and Yutao Luo. 2019a. Legal Judgment Prediction via Multi-Perspective Bi-Feedback Network. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 4085–4091.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019b. XLNet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489.

Hai Ye, Xin Jiang, Zhunchen Luo, and Wenhan Chao. 2018. Interpretable Charge Predictions for Criminal Cases: Learning to Generate Court Views from Fact Descriptions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1854–1864, New Orleans, Louisiana. Association for Computational Linguistics.

Matthew D Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer.

Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Chaojun Xiao, Zhiyuan Liu, and Maosong Sun. 2018. Legal Judgment Prediction via Topological Learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3540–3549, Brussels, Belgium. Association for Computational Linguistics.

Haoxi Zhong, Yuzhong Wang, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. 2020. Iteratively Questioning and Answering for Interpretable Legal Judgment Prediction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(01):1250–1257.

Appendix

A Annotations and Case studies: Agreement in Judgment Prediction for Annotators

Annotation Assignment 1954_13: In this case, although the original decision is that the appeal has been rejected, Experts 1-4 have reached the decision that it has been accepted, while Expert 5 has decided that it has been rejected. This discrepancy appears to owe its origin to the very nature of the case and the issues considered by the court. There had been more than one such issue and separate arguments had been made by appellant in favour of each of such issue and associated prayer. The court appears to have agreed to some of the arguments and disagreed with the rest.

Annotation Assignment 1961_417: In this case, although the original decision is that the appeal has been rejected, Experts 2 and 4 have decided that it has been accepted. Expert 2 appears to have misconstrued certain positions of law and relied unduly upon one of the other cases being cited as precedent (but not considered relevant by the Supreme Court), which might account for the divergence. In case of Expert 4, however, the issue appears to be more of a linguistic matter. Expert 4 has referred to a particular statement made by the court, "The main question that arises in this appeal is whether an illegitimate son of a sudra vis-a-vis his self acquired property, after having succeeded to a half share of his putative fathers estate, will be entitled to succeed to the other half share got by the widow, after the succession opened out to his putative father on the death of the said widow." From this sentence, Expert 4 has drawn the inference that the appellant was the one asking to establish such entitlement. Since the court in subsequent comments agreed that such entitlement does exist, Expert 4 inferred that the appeal had been accepted. However, in reality, the appellant had been contesting such entitlement.

Annotation Assignment 1962_47: In this case, although the original decision is that the appeal has been rejected, Experts 2 and 5 have decided that it has been accepted. This discrepancy appears to owe its origin to both of them having been misled by Sentence 17 of the case, which appears to refer to the Supreme Court having accepted an appeal and merely giving reasons for such order in the present case. However, the case in point was actually arising from an application for review of the court's earlier judgment (acceptance of the appeal), and therefore, when the court was affirming its earlier judgment and giving reasons behind it, it was in reality rejecting this present application for review, that had been made by the party (respondent in the original appeal) aggrieved by the acceptance of such appeal by the court earlier. Experts 2 and 5 could not apparently distinguish the appeal from the review petition and that appears to have led to such discrepancy.

B Models Details

Table 6 summarizes hyperparameter settings for all the models. All the experiments were run on Google Colab[7] and used the default single GPU Tesla P100-PCIE-16GB, provided by Colab.

B.1 Case Prediction Model Details

Classical Models: We considered classical ML models like Logistic Regression, SVM, and Random Forest. We used sentence embeddings via Sent2Vec (Pagliardini et al., 2018) and document embeddings via Doc2Vec (Le and Mikolov, 2014) as input features. Both embeddings were trained on $\text{ILDC}_{\text{multi}}$ as our data is domain-specific. Legal proceedings are typically long documents, we tried out extractive summarization methods (as described in Bhattacharya et al. (2019a)) for gleaning relevant information from the documents and passing these as input to neural models. However, this approach also resulted in classifiers that were no better than random classifier.

We also experimented by using TF-IDF vectors with the classical models like Logistic Regression (LR), Random Forests (RF) and Support Vector Machines (SVM) from the scikit-learn library in python (Pedregosa et al., 2011). However, the results were no better than a random classifier, which, according to us, could be due to the huge length of the documents and they were not able to capture such long term dependencies well enough.

Results: Classical models based on logistic regression and Sent2Vec embeddings performed much worse than the one based on Doc2vec embeddings. It is interesting to see that Doc2Vec+LR has performance competitive to Sequential models. The simple word embedding based model has

[7] https://colab.research.google.com/

similar performance as the more complicated hierarchical attention network model (HAN). The best results are recorded in the Table 4, each for Sent2Vec and Doc2Vec.

Sequential Models: We experimented with standard BiGRU (2 layers) with attention model. We tried 3 different types of embeddings: (i) Word level trained GloVe embeddings (Pennington et al., 2014), with last 512 tokens as input, (ii) Sentence level embeddings (Sent2Vec), where last 150 sentences were input[8], and (iii) Chunk level embeddings (trained via Doc2Vec). Both Sequential models and HAN were trained on both $\text{ILDC}_{\text{multi}}$ and $\text{ILDC}_{\text{single}}$. All the models from here on were trained on Colab[9].

We extracted catchphrases (Mandal et al., 2017) from the $\text{ILDC}_{\text{single}}$ (we could not use this method on $\text{ILDC}_{\text{multi}}$ due to requirement of huge compute resources). After extracting these catchphrases we ranked the sentences from the documents accordingly and used upto 200 sentences only[10]. These top 200 sentences were then mapped to their Sent2Vec embeddings and passed through BiGRU as above.

Results: Sequential models trained on $\text{ILDC}_{\text{multi}}$ and $\text{ILDC}_{\text{single}}$ have similar performances. We also experimented with extracting key sentences from $\text{ILDC}_{\text{single}}$ documents with the help of catchphrases and using these sentences as input (via the Sent2Vec embeddings) to a sequence model. Extracting the key sentences performs better than the using all the sentences but the performance is worse (61% versus 64% F1) than using GloVe embeddings on last 512 words. GloVe embeddings with BiGRU and attention model gave the best performance (64% F1) among the sequential models. The GloVe embeddings (last 512 tokens) with BiGRU + Attention gave the best results among the models mentioned above.

Transformer Models: Recently, SOTA language models have been developed using Transformer Architectures (Vaswani et al., 2017). A number of transformer architectures have been introduced recently. We experimented with BERT (Devlin et al., 2019), DistilBERT (Sanh et al., 2019), RoBERTa (Liu et al., 2019), and XLNet (Yang et al., 2019b). We used HuggingFace library (Wolf et al., 2020) to fine tune BASE models of above transformers from HuggingFace (Wolf et al., 2020) on the last 512 tokens of $\text{ILDC}_{\text{multi}}$[11]. Due to high compute requirements we could not utilize Longformer (Beltagy et al., 2020) and Reformer (Kitaev et al., 2020) models developed especially for long documents.

For the other transformer models we used only the last 512 tokens as input.

Results: Among the combinations of input tokens, the best performance was obtained by using last 512 tokens as input to the BERT Base model. We can observe the trend that the more the tokens from the final parts of the document are taken as input, the better is the prediction performance. This observation agrees with the fact that there are more clues towards the correct prediction in the final parts of the document (since *Arguments*, *Ratio of the decision* etc. Bhattacharya et al. (2019b) most aligned to the judgment are expected to appear more towards the end, closer to the judgment). As for the comparison between different transformers, unsurprisingly, RoBERTa and XLNet perform better than BERT in the prediction sub-task. Similarly, among DistilBERT and BERT, the latter outperforms the other.

Hierarchical Models: In order to use transformers hierarchically, it was first necessary to fine-tune these models on the downstream task of classification. We use two different strategies to fine-tune these:

- On $\text{ILDC}_{\text{multi}}$: Using last 512 tokens only from the documents.
- On $\text{ILDC}_{\text{single}}$: We fine-tune the transformer by dividing each document into chunks of 512 with an overlap of 100 tokens, the label for each chunk is given as the whole document label.

Then we extracted the 768 dimension, $[CLS]$ token embeddings from the transformers for each chunk in all the documents. This was done on $\text{ILDC}_{\text{multi}}$ corpus irrespective of whether it was fine-tuned on $\text{ILDC}_{\text{multi}}$ or $\text{ILDC}_{\text{single}}$. As mentioned in (Devlin et al., 2019) we also experimented with concatenating the last 4 hidden layers of the $[CLS]$ token and taking that as the chunk embedding.

After getting the chunk embeddings we used two types of neural networks: BiGRU and CNN.

For some models, the results varied over multiple runs. For these we recorded their mean and variance on F1 and Accuracy in the table 4.

[8] last 150 sentences covered around 90% of the documents

[9] https://colab.research.google.com/

[10] These covered more than 90% of the $\text{ILDC}_{\text{single}}$.

[11] As shown in Table 4, we also experimented with different sections of documents and we observed last 512 tokens gave the best performance

Results: Information is lost in considering only the last portion of the case proceeding for prediction and this is reflected in the performance of hierarchical models. In general, all the hierarchical models outperform transformer models. Adding attention on top of BiGRU in the hierarchical model does not boost the performance significantly. However, adding a CNN (instead of BiGRU + Attention) on top gives a competitive performance. As for the comparison between the strategies of fine-tuning between $\text{ILDC}_{\text{multi}}$ and $\text{ILDC}_{\text{single}}$, the later seemed to perform worse on prediction. For the hierarchical concatenated model fine tuned on $\text{ILDC}_{\text{single}}$, there was a slight boost in performance.

B.2 Explanability Models and Results Details

To extract explanations from our best model (XLNet + BiGRU), we propose a method inspired from Li et al. (2016) and Zeiler and Fergus (2014). The idea is to use occlusion method at both levels of the hierarchy. For the BiGRU part of the model, for each document we mask each complete chunk embedding one at a time. The masked input is passed through the trained BiGRU and output probability (masked probability) of the label obtained by original unmasked model is calculated. The masked probability is compared with unmasked probability to calculate chunk explainability score. Formally, for a chunk c, if the sigmoid outputs (of the BiGRU) are σ_m (when the chunk was not masked) and $\sigma_{m'}$ (when the chunk was masked) and the predicted label is y then the probabilities and chunk score $s_c = p_m - p_{m'}$ and $p_{m'/m} = \begin{cases} \sigma_{m'/m}, & y = 1 \\ 1 - \sigma_{m'/m}, & y = 0 \end{cases}$

We obtain sentences that explain the decision from the transformer part of the model (XLNet) using the chunks that were assigned positive scores. Each chunk (length 512 tokens) is segmented into

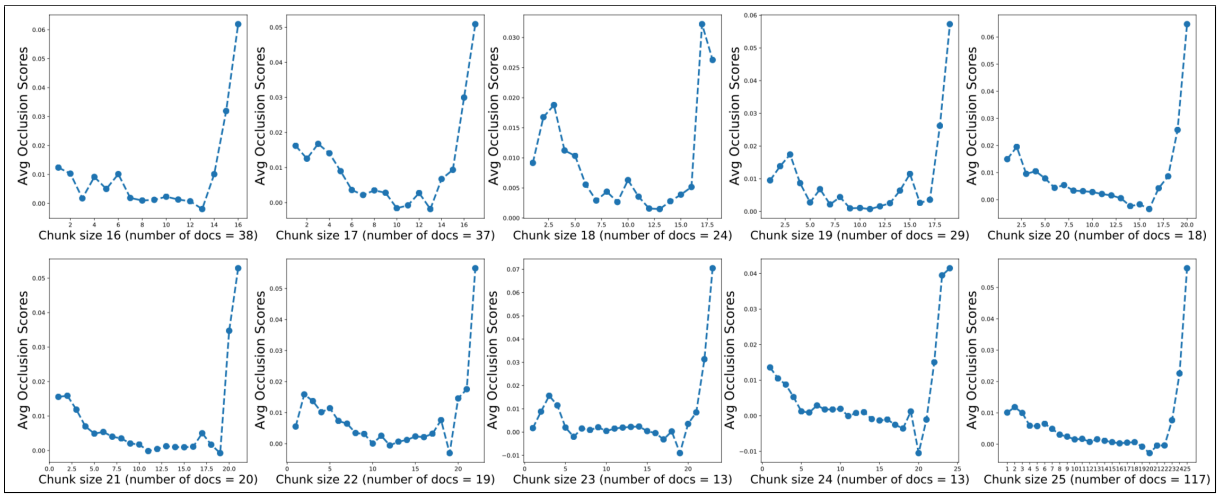

Figure 4: Visualization of Occlusion scores accross full Test set.

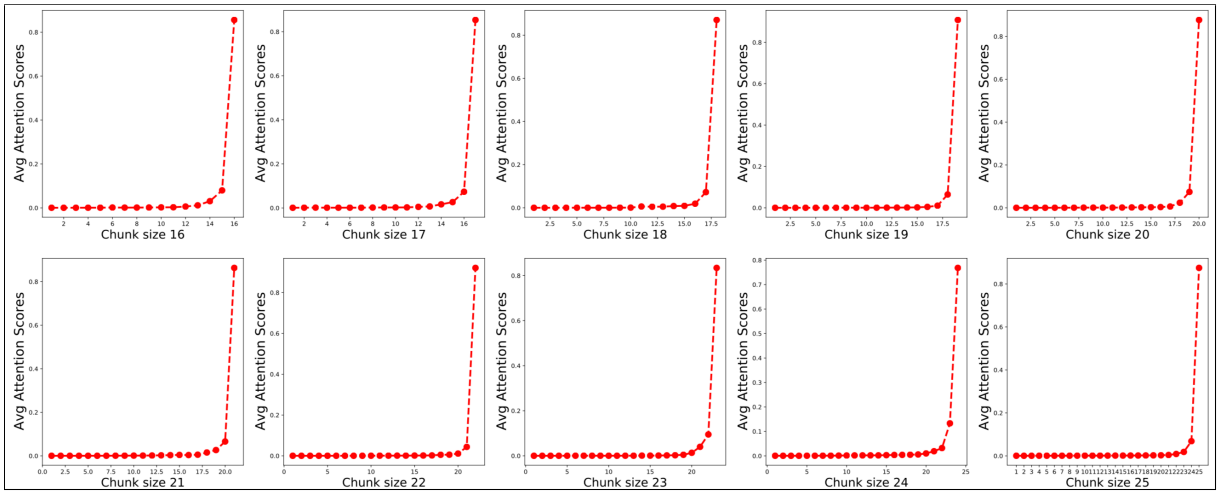

Figure 5: Visualization of Attention scores accross full Test set.

sentences using NLTK sentence splitter (Loper and Bird, 2002). Similar to BiGRU, each sentence is masked and the output of the transformer at the classification head (softmax logits) is compared with logits of the label corresponding to original hierarchical model. The difference between the logits normalized by the length of the sentence is the explanation score of the sentence. Finally, top-k sentences ($\sim$ 40%) in each chunk are selected.

In Figure 4 and Figure 5 we visualize the mean chunk importance scores. Out of the 1517 test documents we average out chunk scores of the documents having same number of chunks. As shown in Figure 5, the attention weights are biased towards the last chunks, thus giving negligible attention to the chunks before. However, in Figure 4, in some of the graphs, the last chunk is given the second-highest score and in 7 out of 10 graphs, it has the highest score. Due to space limitation, we are not providing the graphs for occlusion and attention scores for chunks 1 to 15. But we observed that for these chunks pattern matches for occlusion scores with attention scores. From these observations, we believe it is safe to say that both the methods of visualization affirm our hypothesis that *the most relevant syntactic and semantic information lies towards the end of the case.* Although attention scores are optimized (via loss minimization or accuracy maximization) to concentrate on last chunks, this is not the case with occlusion scores. There is no optimization of occlusion scores, yet they still focus on the chunks at the end which affirms our hypothesis. One might argue that this observation might be due to the transformer being trained on last 512 tokens only. To check this, we also visualized the hierarchical transformers trained on $\text{ILDC}_{\text{single}}$, but the results were similar as to what we have observed in this case.

Model	**Hyper-Parameters (E = Epochs), (Dim = Embedding Dimension), (L = Layers), (att. = attention), (default setting= 512 tokens with overlapping 100 tokens)**
Classical Models on $\text{ILDC}_{\text{multi}}$ train set	
Doc2Vec + LR	dim = 1000 , E = 20
Sent2vec + LR	dim=500, E = 20, Avg Pool
Sequential Models on $\text{ILDC}_{\text{multi}}$ train set	
Sent2vec + BiGRU + att.	dim = 200, E = 1, L = 2
Doc2vec + BiGRU + att.	dim = 1000, E = 2, L = 2
GloVe + BiGRU + att.	dim = 180, E = 3, L = 2
HAN	word dim = 100, sent dim = 100, E = 10
Sequential Models on $\text{ILDC}_{\text{single}}$ train set	
Sent2Vec + BiGRU+ att.	dim = 200, E = 1, L = 2
Doc2vec + BiGRU + att.	dim = 1000, E = 2, L = 2
GloVe + BiGRU + att.	dim = 180, E = 10, L = 2
HAN	word dim = 100, sent dim = 100, E = 10
Catchphrases + Sent2Vec + BiGRU + att.	dim =180, E =5, L = 2
Transformer Models on $\text{ILDC}_{\text{multi}}$ train set	
BERT Base	512 begin tokens, E = 3
BERT Base	256 begin, 256 end tokens, E = 3
BERT Base	256 mid, 256 end tokens, E = 3
BERT Base	128 begin, 256 mid, 128 end, E = 3
BERT Base	512 end tokens, E = 3
DistillBERT	512 end tokens, E = 5
RoBERTa	512 end tokens, E = 5
XLNet	512 end tokens, E = 3
Hierarchical Models on $\text{ILDC}_{\text{multi}}$ train set	
BERT + BiGRU	default setting, E = 5, L = 3
RoBERTa + BiGRU	default setting, E = 2, L = 3, runs = 3
XLNet + BiGRU	default setting, E = 5, L = 2
BERT + CNN	default setting, E = 3, L = 3 (Conv1D)
RoBERTa + CNN	default setting, E = 3, L = 3 (Conv1D)
XLNet + CNN	default setting, E = 3, L = 3 (Conv1D)
Hierarchical Models on $\text{ILDC}_{\text{single}}$ train set	
BERT + BiGRU	default setting, E = 1, L = 2, 3 runs
RoBERTa + BiGRU	default setting, E = 1, L = 2, 3 runs
XLNet + BiGRU	default setting, E = 2, L = 2, 3 runs
Hierarchical Models with Attention on $\text{ILDC}_{\text{multi}}$ train set	
BERT + BiGRU + att.	default setting, E = 2, L = 2, 3 runs
RoBERTa + BiGRU + att.	default setting, E = 2, L = 3, 3 runs
XLNet + BiGRU + att.	default setting, E = 3, L = 2, 3 runs
Hierarchical Models with Attention on $\text{ILDC}_{\text{single}}$ train set	
BERT + BiGRU + att.	default setting, E = 1, L = 2, 3 runs
RoBERTa + BiGRU + att.	default setting, E = 1, L = 3, 3 runs
XLNet + BiGRU + att.	default setting, E = 1, L = 2, 3 runs
Transformers Voting Ensemble	
RoBERTa	fine tuned on last 512 tokens, voting
XLNet	fine tuned on last 512 tokens, voting
Hierarchical concatenated model with att on $\text{ILDC}_{\text{single}}$ train	
XLNet + BiGRU	last 4 layers concat, E = 1, L = 2, 3 runs

Table 6: Hyper-parameters corresponding to every model.