IRIT_IRIS_C at SemEval-2023 Task 6: A Multi-level Encoder-based Architecture for Judgement Prediction of Legal Cases and their Explanation

Nishchal Prasad, Mohand Boughanem, Taoufiq Dkaki
Institut de Recherche en Informatique de Toulouse (IRIT), Toulouse, France
{Nishchal.Prasad, Mohand.Boughanem, Taoufiq.Dkaki}@irit.fr

Abstract

This paper describes our system used for sub-task C (1 & 2) in Task 6: LegalEval: Understanding Legal Texts. We propose a three-level encoder-based classification architecture that works by fine-tuning a BERT-based pre-trained encoder, and post-processing the embeddings extracted from its last layers, using transformer encoder layers and RNNs. We run ablation studies on the same and analyze its performance. To extract the explanations for the predicted class we develop an explanation extraction algorithm, exploiting the idea of a model's occlusion sensitivity. We explored some training strategies with a detailed analysis of the dataset. Our system ranked 2nd (macro-F1 metric) for its sub-task C-1 and 7th (ROUGE-2 metric) for sub-task C-2.

1 Introduction

With the increasing population, there is also an increase in the number of legal cases, and the shortage of capable judges to attend to them makes it more difficult to process them in time, hence creating backlogs. In such a scenario, a method to assist the judges and suggest an outcome (with an explanation) of a legal case becomes necessary. Such methods can also assist a legal professional to help people; unfamiliar with the legal terms and proceedings; to make informed decisions. In the past, there have been many such approaches using machine learning techniques. In Sub-task C of Task 6: LegalEval (Modi et al. (2023)) the organizers tried to tackle this problem in the context of Indian Legal Cases from the Supreme Court of India (SCI). Task C is divided into two sub-tasks, C-1 and C-2. C-1 is the judgment prediction of the court cases/documents, and C-2 is the explanation of the said prediction.

In this work, we propose a classification framework taking a few ideas from the hierarchical transformer approach by Pappagari et al. (2019) which builds upon BERT (Devlin et al. (2019)) as a backbone encoder model. We fine-tune a BERT-based pre-trained model and experiment with features extracted from its last layers, and use it for further classification (sub-task C-1) process, through a combination of transformer encoders and RNNs. We explored the effect of training this framework with different parts of the training dataset and provide an analysis of its impact. For the explanation task (sub-task C-2) we exploit the idea of "occlusion sensitivity" (Zeiler and Fergus (2014)) and develop a mechanism to extract the relevant sentences from a document which serves as the explanation for its prediction using the classification method in sub-task C-1.

Our model ranked 2nd[1] out of 11 participating teams on sub-task C-1 obtaining 0.7228 macro-F1, and 7th[1] out of 11 on sub-task C-2 with ROUGE-2 = 0.0428 (8th, macro-F1 = 0.4). For extra details on the ranking of task C-1 refer to section 2.1.1 and section 5. Our code is available at GitHub[2].

2 Task Description

- **C-1:** Legal Judgment Prediction (LJP): The aim of this task is to predict the decision class for a case document. For the dataset provided (Section 2.1) this becomes a binary text classification problem.

- **C-2:** Court Judgement Prediction with Explanation (CJPE): CJPE's aim is to predict the decision for a document and give its explanation. The prediction here are the relevant sentences from the document which have the most contribution to the predicted decision.

Because explanations are difficult to annotate, explanations have to be made without explicit training

[1] User: irit_iris (https://codalab.lisn.upsaclay.fr/competitions/9558#results)
[2] https://github.com/NishchalPrasad/SemEval-2023-Task-6-sub-task-C-

Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023), pages 686–692

on the annotated explanations. This makes C-2 an extractive explanation task for which no annotated explanation is provided for training with the reasoning that a predictive model should be able to explain its prediction.

2.1 Dataset

SemEval Task-6: LegalEval, subtask C provides a corpus similar to the dataset provided by Malik et al. (2021) (Indian Legal Document Corpus (ILDC)) with the same validation set but with lesser and different training data. The dataset contains documents from the SCI, where the original decisions have been removed. Each document is provided with a label (0 = Rejected or 1 = Accepted) which is used as the class label. The dataset has two sub-sets, Multi and Single. Multi refers to those legal cases where there are multiple petitions with different decisions and Single where there is one same decision for all the petitions. For the Multi sub-set, the final decision is taken as the decision class label for the document. Analysis of the documents shows that the documents are lowercase and are noisy with many grammatical mistakes, misspellings, and word breaks which creep in during the data-cleaning and phrase removal phase. The dataset is slightly imbalanced, the details of which can be found in Table 1. We take this into account while training and experimenting with our approach. The test set for sub-task C-1 contains 1500 unlabeled documents. The Expert subset is for the sub-task C-2 where there are no training data and consists only of a test set with 50 documents (Table 1) for which the predictions and their explanations need to be made from the models developed on Multi, Single or any external data. Since the validation and the hidden test set is a combination of Multi and Single type documents we combined the Multi and Single set for training our models.

2.1.1 External Datasets

We also experimented with ILDC (Table 1), to test our models and approach, which we have also used to develop our models in our previous work (Prasad et al. (2022)). We used this dataset (test set) mainly for testing our approach. We used only one of its trained models to analyze its results on the hidden test set, which achieves the maximum score of 0.7228 macro-F1. With the same model architecture and only the training dataset of LegalEval Task-C, we achieve a score of 0.6848, which still ranks 2nd. We could not update this in the leaderboards due to the competition platform's system of displaying the highest metric value.

Table 1: Dataset statistics describing the split and label ratio (Accepted : Rejected)

Dataset	Split	**Multi**	**Single**	**Expert**
LegalEval Task-C	Train	1935 : 3147	3083 : 1899	-
	Validation	497 : 497		-
	Test	1500 (hidden labels)		50 (hidden explanations)
ILDC External	Train	13385 : 18920	1935 : 3147	-
	Validation	497 : 497		-
	Test	762 : 755		56 (with explanations)
Label (in the dataset)	0 = Rejected, 1 = Accepted			

2.2 Related Work

There have been several works in the past for predicting the judgment of legal cases using only the case facts (Chalkidis et al. (2019),Chalkidis et al. (2021), Zhong et al. (2020a)) which resonates with the primary aim of LegalEval Task-C-1. Pretrained transformer models (Devlin et al. (2019), Vaswani et al. (2017)) have achieved widespread success in natural language processing, and their variants in the legal domain (Chalkidis et al. (2020)'s LEGAL-BERT, Zheng et al. (2021)'s BERT trained on Case-HOLD, Paul et al. (2022)'s InLegalBERT and In-CaseLawBERT) have shown the importance of domain-specific pre-training. In our past work (Prasad et al. (2022)) we showed that intra-domain (different lexicon, syntax, or grammar setting) fine-tuning of a domain-specific pre-trained model, adapts well to its downstream task, which is beneficial when there is a lack of an intra-domain pre-trained language model. An explanation for a judgment is as essential as its prediction, without which the interpretation and reliability of the prediction come into question. Ye et al. (2018) propose a task of court view generation with an interpretable label-conditioned Seq2Seq attention model, with experiments on the Chinese legal case documents where they interpret a charge by selecting the relevant rationales in the document. There are few approaches in the extractive explanation for a model's legal judgment prediction. Zhong et al. (2020b) used Deep Reinforcement learning and developed a "question-answering" based model named QA-judge to provide interpretable decisions for their judgment prediction task. Malik et al. (2021) presented an extractive explanation approach for their CJPE task which is similar to the sub-task C-2.

2.3 Evaluation metric and ranking

The classification (i.e. prediction) on the sub-task C-1 is evaluated using the macro-F1 (m-F1) score. For sub-task C-2 the evaluation metric used is macro-F1 for the classification (i.e. prediction) and the explanation is evaluated using the ROUGE-2 (Lin (2004)) score. In both sub-tasks (C-1, C-2) the participants are ranked according to their maximum metric scores.

3 System Overview

3.1 Sub-task C-1: General Architecture

The documents in the task-C dataset are long with the average document length as $\approx$ 20000 tokens. Processing such a long sequence through a transformer encoder (such as BERT) poses a problem due to a limit on its input length (512 for BERT-based encoders). One turnaround to this limitation is processing the document in chunks through the encoder and then extracting their embeddings for further processing. This was done in Hierarchical Transformers (Pappagari et al. (2019)). We modified their architecture and used a custom fine-tuning approach from Malik et al. (2021) to fine-tune the backbone encoder used in our architecture. The general overview of the architecture can be seen in Figure 1, and their details are described below.

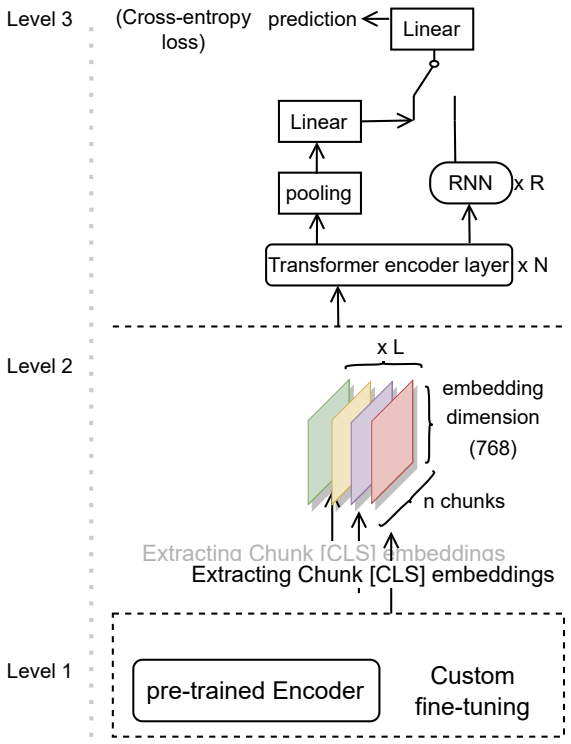

Figure 1: Three-level classification architecture

- Level 1 (Custom fine-tuning): We divide the document into chunks of length "c" (510 for BERT-based model) with overlaps and pass them through the pre-trained encoder tokenizer. The number of chunks for a document will vary with its length. The output tokens are padded and wrapped by [CLS] and [SEP] tokens on both ends to make the input size 512.

  These tokenized representations of each chunk with the document's label form an input to the BERT-based encoder for fine-tuning.

- Level 2 (Extracting chunk embedding): From the last four layers ($l = 4$) of the fine-tuned encoder, for each document we pass the individual tokenized representations of its chunks formed in level 1 and extract their [CLS] representation. A rough representation of the entire document can be obtained by combining all the [CLS] tokens together. So we accumulate these [CLS] representations and attach them with the original document label. This forms the new training data for the next phase of processing.

  The extracted chunk embeddings can be either concatenated (by appending along the feature dimension) or added (along the feature dimension) for post-processing.

- Level 3 (Post-processing chunk embedding): So that one chunk can attend to another (i.e. to learn the intra-chunk attention) we use N $\times$ transformer encoder layer (Vaswani et al. (2017)). We experiment with N = $\{1, 2, 3\}$. The output from the transformer encoder layer is max-pooled and processed through a Linear layer of 128 nodes and the ReLu activation function. We can also process the output from the transformer encoder through R $\times$ RNN layers. We considered 2 $\times$ BiLSTM for experimenting with this setting. For the final binary classification layer we use a simple linear layer of 1 node with a sigmoid activation function. Dropouts can be applied between each transformer layer and linear layers.

  For our best-performing model for the competition, we applied dropout between the transformer encoders (0.25%) and linear layers (0.15%).

3.2 Sub-task C-2: Explanation Extraction Algorithm

We used the idea of occlusion sensitivity (Zeiler and Fergus (2014)) with its two-level approach done by Malik et al. (2021) to develop our method of extracting explanation. The details of the approach can be found in the Algorithm 1. For each document, we divide it into chunks (as in level 1 of 3.1) and extract its chunk's [CLS] embeddings (steps 1-3). For each chunk, we occlude the chunk embedding with zeros and use the occluded input to get the probability output from level 3 of the classification architecture (3.1) (steps 5-8). We take the probability of the full chunk embeddings as the absolute and compare the occluded chunk probability with it. The greater the decrease in probability the more important is that chunk. So, we accumulate all the occluded chunk probabilities and sort them into ascending order and select the top $x\%$ chunks (steps 9,10). For a chunk in the selected chunk; in the order of the sort; we calculate its probability output $p_E(c)$ from the fine-tuned encoder E_f of level 1 of classification architecture. We split chunks into sentences (step 12,13). In a moving window of s sentences (sentence set), for each window, we occlude (zero masks) the whole sentence set and calculate the model's output probability $p_E(c_s)$ from E_f (steps 14-16). This shows us the impact of that sentence set in the output prediction of the model. If $p_E(c_s) > p_E(c)$ we imply that the sentence set is less relevant and penalize it by setting the sentence set score (window_{score}) as $p_E(c) - p_E(c_s)$. For each selected chunk we sort the sentence set according to the sentence set scores in decreasing order of the penalty and choose the top $k\%$ set (steps 17-26). Doing this for all sorted_{chunks} gives the extracted sentences which act as the explanation for the prediction of the document by the proposed Sub-task C-1's architecture.

Algorithm 1 Explanation Extraction Algorithm

Require: From level 3 (section 3.1), select post-processing model T, and the fine-tuned encoder E_f. $x = \%$ of chunks to prioritise. $k = \%$ of sentences to prioritise. $s =$ sentence window size.

1: **for** all documents **do**
2: Divide the document into chunks.
3: Extract chunk [CLS] embeddings from E_f (level 2 of section 3.1).
4: Get probability output from T for the document.
5: **for** each chunk **do**
6: Mask with 0.
7: $\text{probability}_{occluded\ chunk} \leftarrow$ probability from T after masking.
8: **end for**
9: Concatenate all $\text{probability}_{occluded\ chunk}$.
10: $\text{sorted}_{chunks} \leftarrow$ Sort and select the top x chunks
11: **for** chunk c in sorted_{chunks} **do**
12: $p_E(c) \leftarrow E_f(c)$, probability output from E_f
13: Split c into sentences.
14: **for** sentence window set (c_s) in c **do**
15: Mask c_s with 0.
16: $p_E(c_s) \leftarrow$ probability $E_f(c_s)$.
17: **if** $p_E(c_s) > p_E(c)$ **then**
18: $\text{window}_{score} \leftarrow p_E(c) - p_E(c_s)$
19: **else**
20: $\text{window}_{score} \leftarrow p_E(c_s) - p_E(c)$
21: **end if**
22: $c_{score} \leftarrow$ concatenate all window_{score}.
23: **end for**
24: Sort c_{score} in descending order.
25: Keep the top k sentences.
26: **end for**
27: **end for**

4 Experimental setup

We develop our models using the pytorch[3] and tensorflow[4] library. We use InLegalBERT (Paul et al. (2022)) as the encoder model as it was pretrained on the court cases of SCI (the same pool of documents used in the LegalEval task), with better performance than LEGAL-BERT (Chalkidis et al. (2020)) on Indian court setting. We oversampled[5] the training data (to equal class balance) for fine-tuning. Chunk overlap was kept to 90 tokens. AdamW (Loshchilov and Hutter (2019)) optimizer (learning rate $2e^{-6}$) was used for fine-tuning for 4 epochs and we chose the best performing one to extract the [CLS] embeddings from the last 4 layers of InLegalBERT. In level 3 of the classification architecture, we used Adam optimizer (learning rate $3.5e^{-6}$) and loss as "binary cross-entropy". For the transformer encoder layers, we used the number of attention heads = 8, and the internal feed-forward layer dimension as 2048. For the BiLSTM layer, we used 100 nodes. We chose $x\% = 0.3$, $k\% = 0.4$ and s=$\{1, 2\}$ for the explanation extraction algorithm and used the ntlk[6] library for sentence splitting. We also experimented with using the validation set as training data and the train set as validation data after the model and dataset analysis (section 5.1).

[3] https://pytorch.org/
[4] https://www.tensorflow.org/
[5] https://imbalanced-learn.org
[6] https://www.nltk.org/

5 Results

We uploaded the test predictions from some of the best models (Table 2) during the development process and show some of the experimental results for the models developed (with their ablation) during the competition in Table 3. Table 2 shows results with other metric scores (ROUGE-1, ROUGE-L (Lin (2004)), BLEU (Papineni et al. (2002)), METEOR (Lavie and Agarwal (2007)), Jaccard similarity, overlap-min, overlap-max (Malik et al. (2021))) apart from the competition's chosen metrics. We also used the ILDC's test set for testing our developed model to approximate their performance in a real setting. The fine-tuned InLegalBERT achieves an m-F1 score of 74.92 in the LegalEval validation set and 75.25 with over-sampled training data. Oversampling increased the number of samples with a balanced class label which helped the encoder to learn better. We choose the oversampled variant of InLegalBERT to extract the [CLS] embeddings. With N = 2 the performance is slightly higher than N = 3. This is because the increase in the model's parameters slightly overfits the training. Adding the RNN (2 x BiLSTM) reduced the scores. The embeddings from the last four layers of InLegalBERT, when added, had a better m-F1 score (≈ 0.3 points) than concatenating. We could not upload the hidden test results from these models because of the competition deadline but inferring from their performance during the development phase, they may attain higher scores than that of (b) in Table 2.

In Table 2 model (b*) ranks 2nd (C-1 leaderboard) which was trained on a similar larger dataset. With the same architecture, we still rank 2nd (0.6848 m-F1) with model (b) where the 3rd rank has a 0.6782 m-F1 score. Model (a) from Table 3 gives a 0.6575 m-F1 score which is improved by further analysis (Section 5.1) of the dataset and training on the validation set. The data skew (Figure 2) and the ability of task C-1's validation set to generalize over the C-1's training set can be seen with the metric scores in Table 3 which achieves $\geq 95\%$ m-F1 score.

The Explanation Extraction Algorithm is model dependent since its backbone is dependent on the model's output sensitivity to the parts of the input. This can be seen from the table 2, where the better the model's classification capability the better is the similarity metrics in table 2. While some similarity metrics (BLEU, METEOR) may be not behave as such with very slightly better similarity metric performance (when compared upto five decimal places) for a model with less classification performance. But for upto 3 decimal places they are almost similar.

5.1 Analysis

We analyzed the performance of our models during the training and validation phase and saw a huge gap (15 points) in the metric scores where on training with the C-1's train set split, the models in Level 3 achieved accuracy and m-F1 of $\geq 95\%$ while scores on the validation set split of the model were around 79%. So we analyzed the dataset by applying a named entity recognition (NER) and extracting the dates on which the legal cases were appealed. For an SCI legal case document, the date on which it is filed or appealed is in the beginning few sentences of the document. We used the NER (EntityRecognizer) from spaCy[7] to recognize and extract the dates from the first few sentences of the case document. We plot its statistics in Figure 2. As can be seen, the validation set is almost evenly distributed while the train set is skewed toward cases from 1958-1980. The skewness increases in the hidden test set where most of the cases are after 1990. We hypothesize that with the lack of case documents from these recent dates, the models trained on the train set will not be able to predict properly due to a lack of learning from the new law articles or case proceedings.

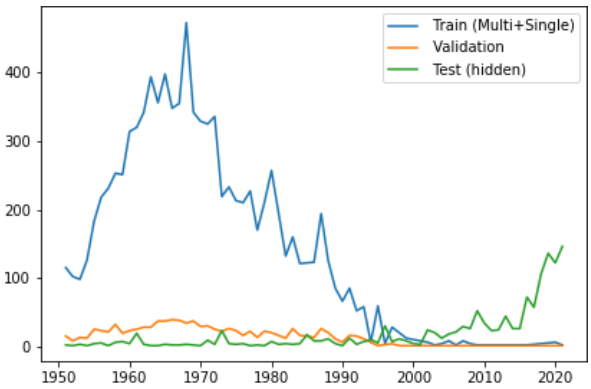

Figure 2: Distribution of cases (by year) in dataset split

6 Conclusion

In this paper, we presented our system for Subtask-C of SemEval 2023 Task-6. We propose a three-layered decision classification architecture for processing large legal case documents that build upon

[7] https://spacy.io/api/entityrecognizer

Table 2: Results on the hidden test data (rounded upto 5 decimal places)

	C-1	C-2								
Models	m-F1	m-F1	ROUGE-2	ROUGE-1	ROUGE-L	Jaccard	BLEU	METEOR	overlap-min	overlap-max
(b*) = (b) trained on ILDC_multi	**0.7228**	**0.4**	**0.04283**	0.19417	0.16802	0.10878	0.07840	0.19204	0.32662	0.15217
(a)	0.6575	0.3942	0.04129	0.19127	0.16498	0.10659	0.07633	0.19186	0.32752	0.14821
(b)	**0.6848**	**0.4**	**0.04195**	0.19617	0.16902	0.10911	0.07469	0.18868	0.33907	0.15057
(c)	**0.6793**	**0.3997**	**0.04189**	0.19674	0.17004	0.10939	0.07440	0.18943	0.33730	0.15121

Table 3: Experimental results of the C-1 classification architecture (in % (rounded upto two decimal places))

		Validation		Test	
		Acc.	m-F1	Acc.	m-F1
Level 1					
fine-tuned InLegalBERT (on Task-C-1 Train set)		Task-C-1 Val. set		ILDC Test set	
no sampling	e=1	74.14	74.2	71.72	71.72
	e=2	74.55	74.92	72.84	73.01
over sampling	e=1	75.15	75.25	72.38	72.40
	e=2	74.24	74.25	73.57	73.62
Level 3					
last l layers (InLegalBERT) Nx encoder	RNN				
(trained on Task-C-1 Train set)		Task-C-1 Val. set		ILDC Test set	
(a) 1x	Yes	79.07	79.01	78.25	78.22
2x	No	80.08	80.01	78.79	78.74
2x	Yes	79.68	79.61	77.79	77.76
(trained on Task-C-1 Val. set)		Task-C-1 Train. set		ILDC Test set	
concat. l = 4					
(b) 3x	No	95.64	95.64	81.54	81.49
2x	No	95.98	95.98	81.62	81.61
(c) 2x	Yes	95.00	95.00	81.54	81.50
add l = 4					
3x	No	96.36	96.36	81.21	81.20
2x	No	96.31	96.31	81.82	81.81
2x	Yes	95.97	95.97	81.08	81.08

pre-trained encoders, embedding extraction, and post-processing with transformer encoder and RNN layers. Which ranked 2nd out of 11 participating teams in sub-task C-1. We explored the effects of training this architecture with ablation and study its impact. We also developed an algorithm for extracting an explanation for a decision prediction and use it for explanation extraction in sub-task C-2, which ranks 7th out of 11 participating teams (ROUGE-2 metric). We give a detailed analysis of the training and test dataset used in the competition. In the future, we plan to take this work further to develop a more general transformer architecture for decision prediction and their explanation (both abstract and extractive) from general large legal case documents and specifically within the French legal system (project LAWBOT[8]).

Acknowledgements

This work is supported by the LAWBOT project (ANR-20-CE38-0013) and HPC/AI resources from GENCI-IDRIS (Grant 2022-AD011013937)

References

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.

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.

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.

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, Prague, Czech Republic. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

[8] https://anr.fr/Projet-ANR-20-CE38-0013

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 (Volume 1: Long Papers)*, pages 4046–4062, Online. Association for Computational Linguistics.

Ashutosh Modi, Prathamesh Kalamkar, Saurabh Karn, Aman Tiwari, Abhinav Joshi, Sai Kiran Tanikella, Shouvik Guha, Sachin Malhan, and Vivek Raghavan. 2023. SemEval-2023 Task 6: LegalEval: Understanding Legal Texts. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, Toronto, Canada. Association for Computational Linguistics (ACL).

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, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Raghavendra Pappagari, Piotr Zelasko, Jesús Villalba, Yishay Carmiel, and Najim Dehak. 2019. Hierarchical transformers for long document classification. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 838–844.

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

Nishchal Prasad, Mohand Boughanem, and Taoufiq Dkaki. 2022. Effect of hierarchical domain-specific language models and attention in the classification of decisions for legal cases. In *Proceedings of the 2nd Joint Conference of the Information Retrieval Communities in Europe (CIRCLE 2022), Samatan, Gers, France, July 4-7, 2022*, volume 3178 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

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 *Computer Vision – ECCV 2014*, pages 818–833, Cham. Springer International Publishing.

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.

Haoxi Zhong, Yuzhong Wang, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. 2020a. Iteratively questioning and answering for interpretable legal judgment prediction. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 1250–1257. AAAI Press.

Haoxi Zhong, Yuzhong Wang, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. 2020b. Iteratively questioning and answering for interpretable legal judgment prediction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(01):1250–1257.