Viettel-AI at SemEval-2023 Task 6: Legal Document Understanding with Longformer for Court Judgment Prediction with Explanation

Thanh Dat Hoang, Chi Minh Bui, and Khac-Hoai Nam Bui
Viettel Cyberspace Center, Viettel Group, Viet Nam
{datht17,minhbc,nambkh}@viettel.com.vn

Abstract

Court Judgement Prediction with Explanation (CJPE) is a task in the field of legal analysis and evaluation, which involves predicting the outcome of a court case based on the available legal text and providing a detailed explanation of the prediction. This is an important task in the legal system as it can aid in decision-making and improve the efficiency of the court process. In this paper, we present a new approach to understanding legal texts, which are normally long documents, based on data-oriented methods. Specifically, we first try to exploit the characteristic of data to understand the legal texts. The output is then used to train the model using the Longformer architecture. Regarding the experiment, the proposed method is evaluated on the sub-task CJPE of the SemEval-2023 Task 6. Accordingly, our method achieves top 1 and top 2 on the classification task and explanation task, respectively. Furthermore, we present several open research issues for further investigations in order to improve the performance in this research field.

1 Introduction

The sub-task of CJPE invites participants to analyze a legal judgment document to predict the outcome of the case and justify their prediction with relevant sentences from the document that contributes to the decision(Malik et al., 2021b). The objective of predicting court judgments is to enhance the efficiency and accuracy of the legal system by providing decision-makers with valuable predictions and insights. Transformer-based models, such as BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019), currently dominate most natural language processing (NLP) tasks, including text classification. However, the quadratic complexity of their attention mechanisms imposes limits on the maximum input length (512 sub-word tokens in BERT, RoBERTa), which may not be able to handle all the meaning in the legal domain where longer documents are common(Malik et al., 2021b).

To address this issue, sparse-attention models, such as LongFormer (Beltagy et al., 2020) and BigBird (Zaheer et al., 2020), increase the maximum input length to 4096 sub-word tokens, which may improve the performance in understanding legal documents. This paper focuses on investigating the data and experimenting with various large models, as we believe this sub-task involves processing long documents. Using the Longformer and TF-IDF features(Zhang et al., 2011), we achieve superior results and apply them to the explanation task. In the explanation task, we design a linear explanation algorithm and show the mechanism effectively inference the most important sentences leading to the decision of judgment. We also provide extensive experiments to analyse the capability of our proposed model on capturing semantic information in judgments. The main contributions of this study are summarized below:

- We analyse judgment corpus and show its unique characteristics that leads to model designation decision.
- We propose a novel approach for judgment decision prediction and explanation by enhancing Longformer with global information TF-IDF.
- We conduct a comprehensive experiment on both prediction and explanation tasks to justify the improvements of our proposed method over other baseline approaches.

In the SemEval 2023 Task 6(Modi et al., 2023), we have achieved the top-1 ranking for the sub-task legal judgment prediction, and top-2 ranking for the sub-task court judgment prediction and explanation[1]. Our implementation is available via Github[2].

[1]https://codalab.lisn.upsaclay.fr/competitions/9558
[2]https://github.com/thanhdath/semeval-2023-legaleval

Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023), pages 862–868

2 Related Work

Pretrained language models, which are trained on large-scale unsupervised corpora, have shown their effectiveness in various downstream tasks with limited training samples such as text classification(Minaee et al., 2021) and text generation(Li et al., 2021).

For text classification, traditional methods often apply a "short encoder" such as BERT(Malik et al., 2021b), RoBERTa(Malik et al., 2021b) and XLNet(Yang et al., 2019). Although these methods have archived promising performance on various domains including news' topic classification and movie reviews analysis(Minaee et al., 2021), they can only adapt to maximum 512 tokens, while real-world documents such as court judgments could be extremely large(Xiao et al., 2018). Therefore, more advanced algorithms including hierarchical models HAN(Yang et al., 2016), BERT + CNN and XLNet + BiGRU(Malik et al., 2021b) were invented to capture the semantic of arbitrary long documents. However, these methods are still not optimal due to two reasons. Firstly, longer documents also cause more noisy information, the methods have no mechanism to automatically de-noise trivial information of judgments such as case number and introduction which do not contribute to the judgment decision(Malik et al., 2021a). Secondly, the provided algorithms cannot be trained on an end-to-end fashion. The language models BERT and XLNet are not optimized to capture the characteristics of this specific legal domain.

While judgment decision prediction could be done automatically by machine learning, providing an explanation of the decision is crucial. This enables lawyers and courts to verify the correctness of machine answers. There has been a vast amount of research in model interpretability including LIME(Ribeiro et al., 2016), SHAP(Lundberg and Lee, 2017), and BreakDown(Staniak and Biecek, 2018). However, most of the explainers measure the important scores at word-level or highlight a short phrase only. Retrieving key sentences that lead to model decisions like CJPE tasks would require algorithm modification.

3 Court Judgment Data Analysis

Malik et al. (2021b) has mentioned that most of the important information is often located in the last part of the documents. As in the structure of judgment, it often begins with a preamble and is followed by state facts of the case, courts analysis(Kalamkar et al., 2022). Therefore, we illustrate the percentage of covered documents given a maximum tokens length. Here we choose Longformer tokenizer which segments the input text from ILDC dataset into subwords, and we examine if the number of subwords exceeds the given maximum tokens length. We also visualize the percentage of important sentences covered by the maximum token length. This is done by analysing if the last part of the documents contains the important sentences provided in ILDC expert dataset. All visualizations are shown in Figure 1.

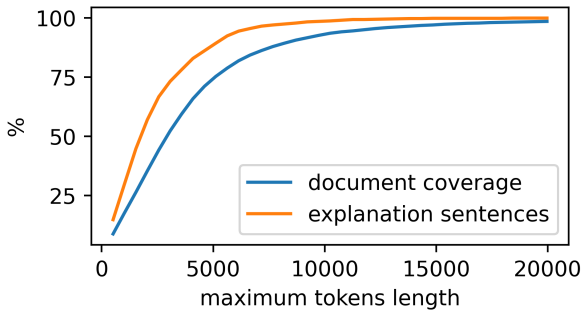

Figure 1: The percentage of documents and explanation sentences covered by given maximum tokens length.

With the maximum tokens length 512, the percentage of document coverage is only 8.8 and only 14.8% of important sentences are within the learned area. Therefore, using a simple encoder such as BERT(Devlin et al., 2018) and RoBERTa(Liu et al., 2019) cannot learn to predict the decision of the judgment. Using sparse-attention algorithm that supports larger models such as Longformer(Beltagy et al., 2020), BigBird(Zaheer et al., 2020) which support 4096 tokens can capture the semantic of about 82.9% documents. We also see that 99% of the important sentences are covered by about 11,000 tokens, this inspires us to design a hierarchical model using Longformer as a primary encoder, as it requires only 3 chunks to cover almost all documents. However, adding more information from the first part could add more noise and cause the learning process harder to optimize.

4 Methodology

4.1 System architecture

We first describe our architecture for the sub-task CJPE in Figure 2, which encompasses three main parts: an Encoder, a TF-IDF vectorizer, and a classifier. Given an input containing an unprocessed

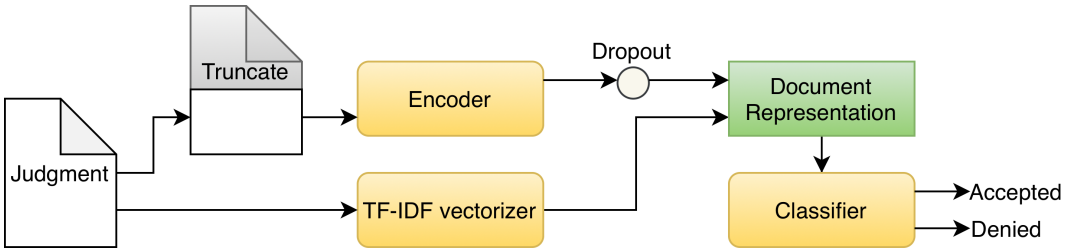

Figure 2: Our proposed system architecture. Accordingly, the architecture obtains four components, which are: i) Encoder layer for enabling contextual representation; ii) TF-IDF vectorizer for enriching global information; iii) Document Representation combining both features; and iv) Classifier for predicting the output.

legal case, we employ an Encoder model (such as RoBERTa Large or Longformer) to vectorize the document into an embedding space. This document embedding often well-captures the semantic information of a document thanks to the advancement of the Transformer architecture(Vaswani et al., 2017). For processing large documents, we have the option to select either the first or last chunk to represent the document, and the exceed tokens are truncated.

The Encoder could be short as it is often limited to the maximum tokens length (e.g. 512, 4096). Therefore, we enhance the document representation by concatenating the output of the Encoder with TF-IDF features. A TF-IDF feature is a document vector that captures the information from the whole document. This gives global information about the document and gives more evidence to predict the judgment decision. We assume that word-level counting would also help better predict the judgment decision. For example, if a judgment consists of a significant number of words "kill" and "dead", the decision would very likely to be accepted. Because the TF-IDF feature dimension could be exceedingly large, we try to build a vocabulary V consisting of meaningful words only. The vocabulary is built from the train data set. In which, we filter the rare words by using a frequency threshold F. All words that have the number of occurrences w less than F are truncated. The most common words are also filtered out since they are mainly stop-words and give zero information to the decision prediction task. The concatenated feature between Encoder and TF-IDF is thus forwarded to a classifier, in order to generate the probability of a document for each class.

4.2 Text Classification

For the classification task, our classifier includes two feed-forward layers aiming to reduce the number of output nodes to two desired classes (Accepted and Denied). Each feed-forward layer follows by a Tanh activation and a Dropout(Srivastava et al., 2014). This would avoid over-fitting problems and also increase the generalization of the model. The input to the classifier is the concatenated features of a Longformer encoder and the TF-IDF features, and the output is the final prediction for the classification task.

The model is trained in an end-to-end fashion with the traditional cross-entropy loss function. In which, we optimize the parameters of both the encoder and classifier at the same time with a small learning rate.

4.3 Explanation

Assume that the model is well-trained for the judgment decision prediction task. Given a decent classification model g which output the vector of prediction probabilities (p_0, p_1) for each class "Denied" and "Accepted", respectively. A document $D = \{s_0, ..., s_{N-1}\}$ is a sequence of N sentences. The explanation task aims to estimate the important score for each sentence s_i using a measurement function f. We define a linear estimator f, as the function for calculating the important score of sentences, which is computed as follows:

$$f(s_i) = g_m(D) - g_m(D \setminus \{s_i\}) \quad (1)$$

where $m = argmax(g(D)) \in \{0, 1\}$ is the predicted label for a document D. The equation 1 comes from a simple perspective that if a sentence s_i is masked and the model cannot correctly predict the label of D, it means the sentence s_i is important. We iteratively compute the important score for each sentence in the document and obtain a set $\{f(s_0), ..., f(s_{N-1})\}$. Later we rank this set to select the top-K most influence sentences to the decision of judgment. Also note that if the classification model leverage only the last part of a

Encoder	Truncation side	#Chunks	Chunk size	F1-macro		
				ILDC	SemEval dev	SemEval test
XLNet + BiGRU*	-	-	512	0.78^{+}	-	-
TF-IDF	-	-	-	0.57	0.56	-
LegalBERT	right	1	512	0.54	0.57	-
RoBERTa	right	1	512	0.54	0.69	-
Longformer	right	1	4096	0.80	0.77	-
LegalBERT	left	1	512	0.74	0.65	-
RoBERTa	left	1	512	0.78	0.67	-
Longformer	left	1	4096	0.86	0.70	-
Longformer H_{mean}	left	3	4096	0.81	0.68	-
Longformer H_{wsum}	left	3	4096	0.82	0.75	-
Longformer T	left	1	4096	**0.87**	0.79	-
Longformer F	left	1	4096	0.86	0.84	0.732
Longformer $F\&T$	left	1	4096	0.86	**0.86**	**0.748**

Table 1: Court decision prediction performance on ILDC and SemEval 2023 dataset. The reported baselines * are taken from (Malik et al., 2021b), where the report result + is selected by the best result of ILDC split (single/multi) dataset. The results on the SemEval test set are the results on the leaderboard.

document, we only rank the important score for sentences in the learned part.

5 Experiment

5.1 Datasets

We perform experiments on two datasets for CJPE tasks including ILDC(Malik et al., 2021b) and SemEval 2023 Task 6 dataset. Although the SemEval 2023 dataset is a subset of ILDC, the characteristics of the two datasets are different, and therefore, the prediction accuracy on the two datasets could be varied. Specifically, the SemEval 2023 test set is only used to evaluate submissions on the leaderboard.

5.2 Hyperparameters Setting

All experiments were produced on a single A100 40Gb GPU and the hyperparameters are set as follows: the number of training epochs 5 and the learning rate 2e-5. For the ILDC dataset, we set the frequency threshold for constructing vocabulary $F = 350$ and this results in the vocabulary size $|V| = 12,315$. For the SemEval dataset, F is set as 100, $|V| = 13,063$. During the training process, we evaluate the prediction accuracy on the validation set after each training epoch and save the best model for inference.

5.3 Court judgment prediction

In this section, we compare the effectiveness of different encoder architectures, including LegalBERT(Chalkidis et al., 2020), RoBERTa(Liu et al., 2019) and Longformer(Beltagy et al., 2020). Since judgment documents often exceed the maximum sequence sizes of pre-trained language models, we configure three settings to learn the semantics of long documents. These settings include 1) truncating the document from either the left side or right side, and 2) designing a hierarchical learning process which is represented in the Appendix section. The prediction performance is shown in Table 1. In which, we use the notation Longformer H to indicate the **H**ierarchical model using Longformer as a primary encoder, $mean$ and $wsum$ are aggregator functions. Longformer F represents the Longformer model that trained on ILDC and **F**inetuning on SemEval dataset, and Longformer T (**T**F-IDF) represents our proposed model enriched by TF-IDF document vector.

The experiment has shown that a powerful encoder architecture is crucial for learning semantic representations of judgments. Longformer achieve the best results compared to LegalBERT and RoBERTa. In particular, a normal Longformer model without an enriching process achieves 0.86 F1-macro on the ILDC test set. When applying hierarchical learning, we did not see the improvements of F1-macro on the ILDC dataset. Longformer H is even worse than a normal Longformer applying for the last part of the document. This is because a larger document includes more noisy information. However, the opposite tendency is observed on SemEval 2023 dataset, this could be due to dataset characteristics. By enriching the

Model name	EM	#S	ILDC expert			SemEval
			ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-2
XLNet + BiGRU*	-	40%	0.451+-0.051	0.297+-0.003	0.424+-0.016	-
Longformer	Linear	40%	**0.660+-0.057**	0.533+-0.072	**0.646+-0.064**	-
Longformer	SHAP	40%	0.641+-0.045	0.508+-0.055	0.623+-0.052	-
Longformer T	Linear	40%	0.655+-0.057	**0.534+-0.072**	0.641+-0.065	-
Longformer F	SHAP	40%	0.628+-0.043	0.491+-0.056	0.610+-0.051	-
Longformer F	SHAP	10	0.413+-0.013	0.269+-0.012	0.396+-0.011	0.046
Longformer $F\&T$	SHAP	40%	0.625+-0.042	0.484+-0.055	0.606+-0.051	-
Longformer $F\&T$	SHAP	10	0.408+-0.012	0.259+-0.013	0.390+-0.011	**0.047**

Table 2: Court prediction explanation performance on ILDC and SemEval 2023 dataset. The reported results * are taken from (Malik et al., 2021b). The results on the SemEval dataset are public on the leaderboard.

Longformer model with global document information TF-IDF, we achieve slightly better accuracy with 0.87 F1-macro on ILDC. In addition, the TF-IDF vector paves the way for our Longformer F to get the top-1 score on SemEval 2023 Task 6 leaderboard with the F1-macro 0.748.

5.4 Judgment explanation

We compare our proposed method with two explanation mechanisms (EM) including a linear masking strategy and SHAP(Lundberg and Lee, 2017). We report the average and standard deviation values for different metrics ROUGE-1, ROUGE-2, and ROUGE-L since each document is annotated by 5 experts. We also try with two settings for selecting the number of retrieved sentences (#S), either selecting the maximum top 10 the most important sentences or the maximum top 40%. It should be noted that the sentences that contribute negative values to the decision are filtered. Table 2 illustrates our judgment prediction accuracy compared to baseline approaches with different settings. The experiment has shown that choosing the number of retrieved sentences is crucial for explaining the judgment decision, as the ROUGE values from selecting the top 40% sentences are higher compared to the top 10. We also notice that the complex explanation method SHAP is worst than the simpler version Linear. This could be because SHAP was invented for inferencing the most important words that contribute to the prediction but are not compatible with sentence-level inference.

6 Leader-board of the Task

Table 3 show the results on the leaderboard for subtask C in SemEval 2023 Task 6[3]. Although we achieved top-1 with a score of 0.748 F1-macro in subtask C1, the performance on C2 is much lower compared to C1 with 0.479 F1-macro. We suspect that the text or domain of subtask C2 may differ from that of subtask C1, which causes the performance reduction.

Subtask C1		**Subtask C2**		
Team	F1	Team	F1	ROUGE-2
Ours	**0.7485**	Team 1	0.5417	0.0473
Team 2	0.7228	**Ours**	**0.4797**	**0.0470**
Team 3	0.6782	Team 3	0.4789	0.0465
Team 4	0.6771	Team 4	0.4781	0.0411
Team 5	0.6735	Team 5	0.4781	0.0410

Table 3: Leaderboard result of subtask C.

7 Conclusions

In this paper, we fine-tuned the Longformer model by incorporating TF-IDF features and employed various techniques to explain the CJPE task. Along with implementation, we experimented with different input settings and parameters to discover the best possible solution. As for future research, we aim to delve deeper into the SHAP theory and investigate ways to improve the explanation task beyond the simple linear method. Additionally, we are interested in exploring novel strategies to train and represent documents and identify essential features that influence court predictions and also the explanation task. Further investigation on discovering a more effective approach to leverage the hierarchy technique for large-scale language models presents an opportunity for enhancement in this endeavor and ought to be acknowledged.

[3]https://codalab.lisn.upsaclay.fr/competitions/9558#results

References

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

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. Legal-bert: The muppets straight out of law school. *arXiv preprint arXiv:2010.02559*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Prathamesh Kalamkar, Aman Tiwari, Astha Agarwal, Saurabh Karn, Smita Gupta, Vivek Raghavan, and Ashutosh Modi. 2022. Corpus for automatic structuring of legal documents. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4420–4429, Marseille, France. European Language Resources Association.

Junyi Li, Tianyi Tang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. Pretrained language models for text generation: A survey. *arXiv preprint arXiv:2105.10311*.

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. *CoRR*, abs/1907.11692.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.

Vijit Malik, Rishabh Sanjay, Shouvik Kumar Guha, Shubham Kumar Nigam, Angshuman Hazarika, Arnab Bhattacharya, and Ashutosh Modi. 2021a. Semantic segmentation of legal documents via rhetorical roles. *CoRR*, abs/2112.01836.

Vijit Malik, Rishabh Sanjay, Shubham Kumar Nigam, Kripabandhu Ghosh, Shouvik Kumar Guha, Arnab Bhattacharya, and Ashutosh Modi. 2021b. ILDC for CJPE: Indian legal documents corpus for court judgment prediction and explanation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4046–4062, Online. Association for Computational Linguistics.

Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. 2021. Deep learning–based text classification: a comprehensive review. *ACM computing surveys (CSUR)*, 54(3):1–40.

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).

Hai Van Pham, Dat Hoang Thanh, and Philip Moore. 2021. Hierarchical pooling in graph neural networks to enhance classification performance in large datasets. *Sensors*, 21(18).

Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should I trust you?": Explaining the predictions of any classifier. *CoRR*, abs/1602.04938.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15:1929–1958.

Mateusz Staniak and Przemysław Biecek. 2018. Explanations of Model Predictions with live and breakDown Packages. *The R Journal*, 10(2):395–409.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi 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 M. 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*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

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, San Diego, California. Association for Computational Linguistics.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big bird: Transformers for longer sequences. *CoRR*, abs/2007.14062.

Wen Zhang, Taketoshi Yoshida, and Xijin Tang. 2011. A comparative study of tf*idf, lsi and multi-words for text classification. *Expert Systems with Applications*, 38(3):2758–2765.

8 Appendix

8.1 Hierarchical Longformer for Court Judgment Prediction

In this section, we provide model details and implementation of Hierarchical Longformer, which is mentioned in the experiment section. The hierarchical architecture aims to learn the semantic of arbitrary long documents. Given a document D, we first split D into chunks with overlapping 100 tokens, and select maximum C chunks from the ending of the document. This enhances the learning area for Longformer compared to the original one which supports only 4096 tokens. By using 3 chunks, we can cover mostly all documents in the ILDC corpus, as 99% of important sentences is within 11.000 last tokens (see Section 3).

Let $h_i = g(c_i)$ be the chunk representation of chunk c_i, where g represents an encoder. Using hierarchical model, the document representation is computed as follows:

$$Z_D = aggregator(\{h_0, ..., h_{C-1}\}) \quad (2)$$

where $aggregator$ aims to aggregate chunk embeddings to a vector. $aggregator$ could be a mean, sum, or attention function. In this work, since the number of chunks is small when using the encoder Longformer. We use $mean$ and weighted sum ($wsum$) aggregators instead of complex mechanisms such as attention. The $mean$ function simply takes the element-wise mean of the vectors in $\{h_0, ..., h_{C-1}\}$.

Inspired from the work of Pham et al. (2021) which computed the graph embedding by applying the weighted sum function on node representations. We consider chunk embedding equals to node-level and the document vector is similar to graph embedding. Thus, the weighted matrix $S \in R^{C\times 1}$ for every chunks is computed as follow:

$$S = softmax(MLP(\{h_0, ..., h_{C-1}\})) \quad (3)$$

where MLP includes two linear layers with a non-linear activation function ReLU. The MLP function transforms chunk representations into scalar values which represent the important scores of each chunk. Thus, the overall document embedding Z_D is the weighted sum of chunk embeddings:

$$Z_D = \sum_{i=0}^{C-1} S_i h_i \quad (4)$$

The document embedding is thus forwarded to a classifier as introduced in Section 4.1 to predict the decision.

Hierarchical models could easily lead to the out-of-memory issue. This is because the memory needed to store model parameters and gradient values is about C times higher compared to a non-hierarchical one. However, we have implemented the hierarchical model which enables mini-batch gradient descent and end-to-end training thanks to the support of Huggingface(Wolf et al., 2020) and PyTorch Scatter[4] libraries.

8.2 Implementation details of Shapley values for model explanation

This section provides details on the implementation of SHAP(Lundberg and Lee, 2017) for court judgment explanation. SHAP (SHapley Additive exPlanations) is a technique based on game theory that is used to clarify the results produced by a machine learning model. It links the appropriate allocation of credit with explanations that are focused on specific instances by utilizing the well-established Shapley values as well as LIME (Local Interpretable Model-agnostic Explanations)(Ribeiro et al., 2016).

Since the original SHAP aims to measure the importance at word-level, we implemented our method with some modifications with the help of the SHAP library [5] (Lundberg and Lee, 2017) and Huggingface (Wolf et al., 2020). In particular, we modified the masker function from the SHAP library code to compound the entire sentence as a single feature. So, the output will calculate the importance score for each sentence in the chunks. After estimating the important scores for all sentences, we rank them and select the top-K most influential sentences. Those sentences with negative scores are filtered out. Details of our implementation can be found via Github[6].

[4] https://github.com/rusty1s/pytorch_scatter
[5] https://github.com/slundberg/shap
[6] https://github.com/thanhdath/semeval-2023-legaleval