Explanation Extraction from Hierarchical Classification Frameworks for Long Legal Documents

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

Abstract

Hierarchical classification frameworks have been widely used to process long sequences, especially in the legal domain for predictions from long legal documents. But being black-box models they are unable to explain their predictions making them less reliable for practical applications, more so in the legal domain. In this work, we develop an extractive explanation algorithm for hierarchical frameworks for long sequences based on the sensitivity of the trained model to its input perturbations. We perturb using occlusion and develop Ob-HEx; an Occlusion-based Hierarchical Explanation-extractor. We adapt Ob-HEx to Hierarchical Transformer models trained on long Indian legal texts. And use Ob-HEx to analyze them and extract their explanations for the ILDC-Expert dataset, achieving a minimum gain of 1 point over the previous benchmark on most of our performance evaluation metrics.

1 Introduction

Deep-learning-based hierarchical classification models are one of the important techniques for classifying inputs with long sequences and in terms of performance and computational requirements, these hierarchical models have shown to be at par (or better in some cases) with single standalone models which are limited to a certain input length (Chalkidis et al., 2022, 2019; Zhang et al., 2019). These hierarchical models have been largely used recently in the legal NLP domain, especially because of the long lengths of legal case documents. Amongst them, the variants of Hierarchical Transformers have seen quite a lot of usage (Pappagari et al., 2019; Zhang et al., 2019; Malik et al., 2021; Chalkidis et al., 2019, 2021, 2022; Prasad et al., 2022, 2023b,a, 2024; Modi et al., 2023). One of their major drawbacks is they are black boxes with no explanation for their predictions, and explanations are desired especially for reliability in high-stakes fields such as law and medicine. In this work, we develop and test Ob-HEx, an attribution-based post-hoc explanation (Molnar, 2022) extraction algorithm for these hierarchical classification models, which does not require training and relies only on the trained model and its input. In scenarios where there is a lack of annotation to train an explanation algorithm, an extractive explanation method is a good fit to create interpretations of the predicted judgments, which is synonym to the idea of our explanation algorithm. Also, explaining predictions of hierarchical models from long legal documents is a major problem in developing a reliable legal judgment prediction system. In our work, we focus on interpreting the hierarchical predictive models trained on long legal documents, where we rank and extract relevant sentences from the input document that impacted the prediction from the model. These sentences can serve as an explanation, to guide an expert on what led to/triggered a certain prediction. We test Ob-HEx for analysis and explanation from hierarchical models of Malik et al. (2021) and Prasad et al. (2022) on ILDC$_{expert}$ (Malik et al., 2021) obtaining new benchmarks.

2 Related Work

Past work on the explainability of deep neural networks (DNN) (Ras et al., 2022) used the attribution-based perturbation methods for explanations of images and short-text DNN classification models (Zhou et al., 2015; Li et al., 2016; Fong and Vedaldi, 2017; Zhou et al., 2016) that rely on the input and the DNN model's sensitivity to it, but these methods in our experiments and also of Malik et al. (2021) become complex to adapt to hierarchical DNN models for long documents. In the explanation of hierarchical models, little work has been done of which one is by Landecker et al. (2013) where they developed contribution propagation to explain individual image classification. More recently, in the legal domain, some strategies such as

Findings of the Association for Computational Linguistics: NAACL 2024, pages 1192–1201

occlusion sensitivity, keyword-based matching, extractive summarization, and span lengths were used for explanation extraction from hierarchical transformer models for long documents (Malik et al., 2021; Modi et al., 2023). Since we aim to extract explanations without training and solely relying on the trained model we develop the idea of attribution-based perturbation/occlusion sensitivity (Petsiuk et al., 2018) for hierarchical models for long legal documents, where the sentences/paragraphs are scored hierarchically (using a scoring function) against their absence in the input, and finally chosen according to the desirability of the scores (higher or lower).

3 Methodology

3.1 Occlusion-based Hierarchical Explanation-extractor (Ob-HEx):

Consider a hierarchical classification model M with r levels of hierarchy where each level was trained separately on its input I^r=$\{i^r_j | 0 \leq j \leq n\}$ of length n. Going from the top level to the bottom, consider for level $l \leq r$, $M^l(I^l)$=O^l_I to be the prediction without any occlusion, and $M^l(\{I^l|i^l_j\})$=$O^l_{I(j)}$ to be the prediction after the occlusion of i^l_j in I^l. The occlusion can be done by masking individual parts.

Occlusion-sensitivity Impact function: We define an "occlusion-sensitivity impact function" as $\hat{L}$, and choose $\hat{L}$ as the loss function L^l which was used to train the level l of the hierarchy i.e. the layer-wise loss functions L^l are chosen as $\hat{L}^l$.

For an input I, if an absolute prediction, P^t_I from the level t in the hierarchy was used to train all its lower levels, we take it as the "absolute-predicted class label" with which we rank the inputs in all the lower levels. "Absolute-predicted class label" is the prediction by that respective level of hierarchy and "absolute" means that predictions are changed to absolute labels. For example, in a binary labelling system the absolute-predicted class label P^t_I = 1 for a prediction output O^t_I of 0.7 probability.

For a level $l \leq t$ of hierarchy, if P^t_I is the absolute-predicted class label then,

$$\hat{L}^l_{I(j)} = L^l(O^l_{I(j)}, P^t_I), l \leq t \quad (1)$$

i.e loss of $O^l_{I(j)}$ from P^t_I. This impact function measures the importance of the input's occluded part for a prediction from the change in its prediction loss from P^t_I. Higher loss means more impact.

Normalized Weighted Occlusion-sensitivity Score: To rank these losses in terms of impact, we measure the deviance of $\hat{L}^l_{I(j)}$ from $\hat{L}^l_I$ by computing the "normalized weighted occlusion-sensitivity score" $\hat{S}^l$.

$$S^l_{I(j)}(s^l_I, \hat{L}^l_{I(j)}, \hat{L}^l_I) = s^l_I \times \left(\hat{L}^l_{I(j)} - \hat{L}^l_I\right) \quad (2)$$

$$\hat{S}^l_{I(j)}(s^l_I, \hat{L}^l_{I(j)}, \hat{L}^l_I) = \frac{S^l_{I(j)} - min(S^l_I)}{max(S^l_I) - min(S^l_I)} + \delta \quad (3)$$

Here $s^l_I = \hat{S}^{l+1}_I$ is the score weight from I^l's fragment used in the previous level of the hierarchy & $\hat{S}^{r+1}_I = 1$. We shift the axis by adding a constant δ in eq. 3 to keep the score > 0.

Ranking the input fragments for a hierarchical predictive model by weighing the impacts of the higher layers of the hierarchy on the lower layers using the "normalized weighted occlusion-sensitivity score", helps to align the impacts from all the layers of the hierarchy.

We calculate $\hat{S}$ starting from the top to the base level in the hierarchy. This scores fragments of the input, that can be ranked, from which the top $k\%$ input fragments can be chosen and ordered to form an explanation.

3.2 Base hierarchical model:

Here we adapt Ob-HEx to explain the decision prediction made by the trained hierarchical transformer models XLNet+BiGRU from Malik et al. (2021) and LEGAL-BERT+BiGRU from Prasad et al. (2022). These models process a long document broadly in two levels of hierarchy. In the first level the document is divided into chunks of 512 tokens, and using its gold class label the backbone transformer encoder (T) is fine-tuned on individual chunks. The chunk's global embedding is extracted from this fine-tuned transformer, which is combined to form another set of training data for the second level of hierarchy (BiGRU) which learns global document representation for final classification (M).

Ob-HEx adaptation to base hierarchical model: We implement Ob-HEx to process a document from the base model in its two levels (r=2) of hierarchy. (a) Find the impactful chunks from level l=2. (b) Find impactful sentences from these chunks from level l=1.

For occlusions, we use use zero-masking (0 value). Since binary cross-entropy loss (BCE_{loss}) and the same gold labels were used to train both

the levels of hierarchy in the base model, we fix L^l=BCE_{loss} for both levels l=$\{1, 2\}$, such that $\hat{L}^l_{I(j)}$=$BCE_{loss}(O^l_{I(j)}, P^l_I)$, where P^l_I=$P^{l=2}_I$ is the absolute final predicted label from the last level of hierarchy (i.e. we fix $t = 2$ in eq. 1). See appendix A and the GitHub repository [1] for a detailed implementation.

3.3 Experimental Setup

We use ROUGE-1, ROUGE-2, ROUGE-L (Lin, 2004), Jaccard similarity, and BERTScore (Zhang et al., 2020) to compare the model's explanation with the expert's. BERTScore was calculated using "microsoft/deberta-xlarge-mnli" (HuggingFace[2]). We also evaluate explanations from CJPE and Ob-HEx on the ranking-performance/length ratio, where we try to see how similar are the ranked sentences to the gold explanations (similarity is based on the chosen metric scores above) as the explanation lengths are restricted. A higher ratio indicates better performance.

3.4 Baseline explanation algorithm

To compare with Ob-HEx we use the algorithm developed by Malik et al. (2021) for their long sequence hierarchical models, and refer to as CJPE. CJPE ranks sentences from a document using a "chunk explainability score" based on the probability output of the model and takes the top $\approx$ 40% sentences as explanations. We use a chunk length of 512 tokens for all, and k=$\{0.15, 0.1\}$ for Ob-HEx with δ=0.01 (eq. 3). Since CJPE's explanations are quite long we also compare its top 256 and 512 words with Ob-HEx's (§4). We did not alter CJPE's k-value as it's different from Ob-HEx's. Ob-HEx ranks all the sentences in the document, while CJPE chooses only positive chunks and ranks their sentences. Reducing CJPE's k-value would make the explanations too short for some documents, and ultimately a lower performance. So to have a fair comparison, we choose the top 512/256 words and evaluate Ob-HEx (§4) with lower k=$\{0.15, 0.1\}$ for shorter sentences than CJPE (Fig. 2).

3.5 Dataset

We use ILDC$_{Expert}$ dataset from Malik et al. (2021), which includes unstructured English case transcripts from the Supreme Court of India (SCI) with the final decisions removed. A decision of "rejected" or "accepted" made by the SCI judge(s) serves as the class label. ILDC$_{Expert}$ (Table 1) is a test set consisting of gold explanations by legal experts which are texts from the document that are most relevant to the judgment. These gold explanations are ranked from 1-10, 1 being the most relevant to the judgment, and 10 being the least. The dataset statistics can be seen in Table 1.

# documents			**# explanations**				**# experts**			
56			280				5			
average # words			**maximum # words**				**labels**			
3716			23792				1 = Accepted 0 = Rejected			
Ranks	1	2	3	4	5	6	7	8	9	10
Average # words	306	406	456	273	88	28	19	6	3	1

Table 1: ILDC$_{Expert}$ statistics

On moving down to the last ranks, some experts' explanations have no sentences, hence the average number of words also becomes less. Also, since combining sentences in ranks 1-10 gives an average length of 1586 words (Table 1) which is quite large for an explanation, we mainly show the comparison of gold explanations in ranks 1-1 & 1-3 with the explanations from Ob-HEx and CJPE.

To have a comparable length with ranks 1-1 & 1-3 of the gold explanations and for fair evaluations, we also constraint the explanation lengths from Ob-HEx and CJPE respectively (§3.4, §4).

4 Results and Discussions

Analysis of explanation lengths: Figure 2 shows the distribution of percentage variation of the explanation length from CJPE to Ob-HEx, over the whole dataset, and shows that it is not influenced by a few documents. As seen in Figure 2, explanations from CJPE are quite long. The explanation from CJPE is 8.62% longer than Ob-HEx with k=0.15, and 21.4% and 8.9% longer than Ob-HEx@k=0.1 for their top 512 and 265 words respectively.

So we compare Ob-HEx's and CJPE's explanations on three fronts, (a) Long explanations, (b) Short explanations and (c) Brief explanations as shown in Figure 2. Table 2 shows the experimental results of these comparisons for the chosen evaluation metrics (§3.3). See §B for more detailed results.

(a) Long explanations: We compare CJPE vs Ob-HEx@k=0.15 with the gold (expert's) explanations in ranks 1-3. For XLNet+BiGRU, Ob-HEx performs better than CJPE in almost all metrics

[1] https://github.com/NishchalPrasad/Ob-HEx

[2] https://huggingface.co/docs/evaluate

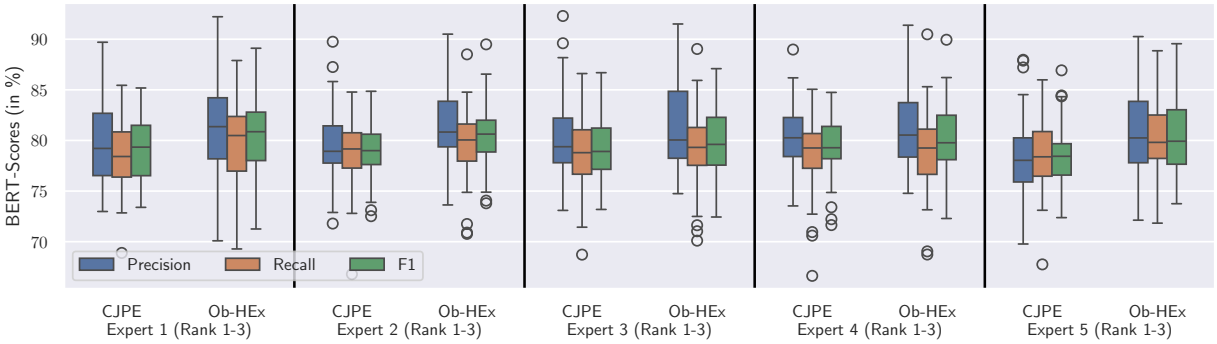


Figure 1: Box plot of BERTScore, for top 512 words, from CJPE & Ob-HEx@k=0.1 vs experts. (XLNet+BiGRU)

* = mean score	Expert				
	1	2	3	4	5
XLNet+BiGRU	(Baseline) CJPE vs Rank (1-3)				
ROUGE-1 *	45.25	44.58	**45.66**	46.2	41.68
ROUGE-2 *	29.02	26.82	**31.43**	30.1	23.98
ROUGE-L *	41.84	39.5	42.52	42.83	36.48
BERTScore(F1) *	79.06	78.98	79.4	79.43	78.45
Jaccard *	31.83	30.77	**33.03**	32.06	28.09
XLNet+BiGRU	Ob-HEx@k=0.15 vs Rank (1-3)				
ROUGE-1 *	**48.44**	**47.46**	45.58	**48.59**	**45.11**
ROUGE-2 *	**32.88**	**29.58**	31.38	**33.17**	**27.47**
ROUGE-L *	**46.26**	**44.15**	**44.08**	**46.56**	**41.76**
BERTScore(F1) *	**80.8**	**80.97**	**81.17**	**80.52**	**80.15**
Jaccard *	**35.05**	**33.12**	32.62	**33.83**	**30.83**
LEGAL-BERT+BiGRU	Ob-HEx@k=0.15 vs Rank (1-3)				
ROUGE-1 *	45.82	44.68	41.52	43.63	42.52
ROUGE-2 *	30.18	27.12	27.13	27.62	23.66
ROUGE-L *	43.97	41.35	39.98	41.62	38.88
BERTScore(F1) *	80.22	80.13	80.7	79.62	79.92
Jaccard *	32.34	30.67	28.87	29.15	28.53
XLNet+BiGRU	(Baseline) CJPE (top 512 words) vs Rank (1-3)				
ROUGE-1 *	40.36	40.3	**40.02**	**40.67**	38.44
ROUGE-2 *	23.97	22.26	**26.21**	24.76	20.14
ROUGE-L *	36.79	35.04	**37.08**	37.31	32.61
BERTScore(F1) *	79.06	78.98	79.4	79.43	78.45
Jaccard *	27.5	27	**27.74**	**27.19**	25.34
XLNet+BiGRU	Ob-HEx@k=0.1 (top 512 words) vs Rank (1-3)				
ROUGE-1 *	**41.51**	**42.65**	36.34	40.06	**42.3**
ROUGE-2 *	**26.39**	**25.38**	23.15	**25.68**	**24.91**
ROUGE-L *	**39.53**	**39.21**	35.06	**38.22**	**38.8**
BERTScore(F1) *	**80.62**	**80.52**	**80.3**	**80.06**	**80.35**
Jaccard *	**28.9**	**29.33**	24.42	26.63	**28.89**
LEGAL-BERT+BiGRU	Ob-HEx@k=0.1 (top 512 words) vs Rank (1-3)				
ROUGE-1 *	39.18	38.83	33.68	35.99	38.45
ROUGE-2 *	24.24	21.83	20.45	21.05	20.11
ROUGE-L *	37.31	35.61	32.42	34.23	34.58
BERTScore(F1) *	79.9	79.77	80.08	78.78	79.63
Jaccard *	26.91	26.04	22.37	23.18	25.55
XLNet+BiGRU	(Baseline) CJPE (top 256 words) vs Rank 1				
ROUGE-1 *	29.19	31.47	33.09	28.42	26.98
ROUGE-2 *	11.35	13.5	17.05	12.23	9.76
ROUGE-L *	23.32	25.27	27.84	22.82	21.72
BERTScore(F1) *	75.88	76.94	78.75	75.95	75.33
Jaccard *	18.33	19.79	21.59	17.73	16.41
XLNet+BiGRU	Ob-HEx@k=0.1 (top 256 words) vs Rank 1				
ROUGE-1 *	**31.85**	**35.4**	**36.24**	**29.77**	**31.39**
ROUGE-2 *	**14.65**	**17.41**	**20.52**	**13.96**	**14.56**
ROUGE-L *	**27.81**	**30.81**	**33.17**	**26.08**	**27.45**
BERTScore(F1) *	**77.52**	**78.56**	**79.65**	**77.19**	**77.62**
Jaccard *	**20.85**	**23.44**	**24.24**	**19.24**	**20.1**
LEGAL-BERT+BiGRU	Ob-HEx@k=0.1 (top 256 words) vs Rank 1				
ROUGE-1 *	30.19	32.63	33.94	26.83	29.87
ROUGE-2 *	12.41	13.5	18.3	10.53	11.75
ROUGE-L *	26.7	27.88	31.11	23.2	25.67
BERTScore(F1) *	77.2	78.15	79.48	76.27	77.10
Jaccard *	19.63	20.98	22.22	16.88	18.9

Table 2: Extracted explanations vs ILDC$_{Expert}$'s (values are in percentage (%)).

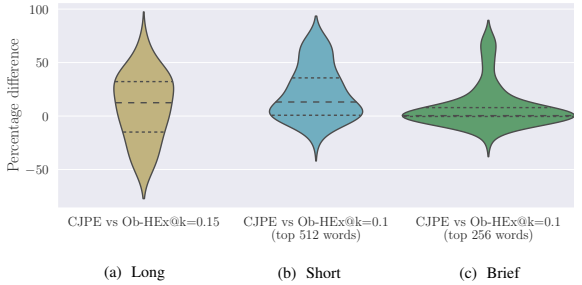


Figure 2: Violin plot of explanation lengths from CJPE vs Ob-HEx.

over all the experts, with an average metric points gain of 2.36, 2.63, 3.93, 1.66, and 1.93 in ROUGE-1, ROUGE-2, ROUGE-L, BERTScore(F1) and Jaccard similarity respectively. While for LEGAL-BERT+BiGRU the performance is slightly better than the baseline in some metrics. Since Ob-HEx infers the focus points of a trained hierarchical model using its input, its explanations from LEGAL-BERT+BiGRU only reflect the main focus parts in its input. And since LEGAL-BERT+BiGRU was trained using all the parts of the document rather than only the last parts as done in XLNet+BiGRU, their focus points are different (Figure 3). This gives some sentences that may not be present in the expert's explanations but are useful for a robust prediction as LEGAL-BERT+BiGRU outperforms XLNet+BiGRU (Prasad et al., 2022).

(b) Short explanations: We compare the top 512 words from CJPE and the top 512 words from Ob-HEx@k=0.1 with the gold (expert's) explanations in the ranks 1-3. For XLNet+BiGRU, Ob-HEx has an average metric points gain of 0.614, 1.634, 2.398, 1.31, and 0.68 in ROUGE-1, ROUGE-2, ROUGE-L, BERTScore(F1) and Jaccard similarity respectively over the baseline (CJPE). A box plot; showing the first quartile, third quartile and

the median; of BERTScore performance (Precision, Recall, F1-score) of Ob-HEx and CJPE vs experts for all the documents can be seen in Figure 1. We see that even for shorter explanations (≈ 400-500 words), Ob-HEx better ranks the predictive-sensitive sentences from the hierarchical model and better captures the semantic similarities with the gold explanations than CJPE.

(c) Brief explanations: To see the ranking-performance/length ratio of Ob-HEx we further constraint the explanation length and compare its similarity performance over the baseline. To do so we experiment with the top 256 words from CJPE and the top 256 words from Ob-HEx@k=0.1 and compare them with the gold explanations only in rank 1. This is done considering the average length of rank 1 explanations is 306 words (Table 1). We see that Ob-HEx still performs better than CJPE with a gain of 3.1, 3.442, 4.87, 1.538, and 2.804 average metric points in ROUGE-1, ROUGE-2, ROUGE-L, BERTScore(F1) and Jaccard similarity respectively.

This shows that the ranking-performance/length ratio of Ob-HEx is better than CJPE. This can be attributed to how Ob-HEx ranks the sentences, where it hierarchically uses the s_I (§3.1) from the previous layers which helps it to relatively measure the importance of each part of the input document.

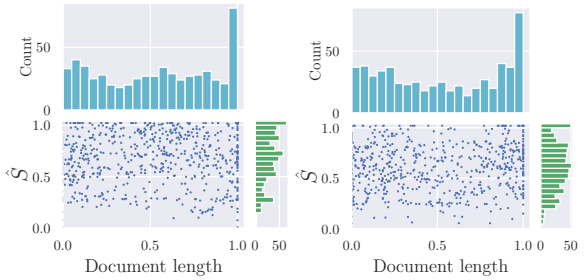
Even though CJPE has longer explanation lengths (Figure (2)) for long, short and brief explanations, its performance is lower than that of shorter explanations from Ob-HEx.

Overall, for the same base hierarchical model, Ob-HEx gives shorter and better-ranked sentences as explanations than CJPE.

Since the Ob-HEx does not train and depends on the hierarchical model we get the same results for every run (except when the hierarchical model is updated with new weights). So we cannot perform a significance test for the runs. But in such situations, the significance could be seen from different test sets, where every expert's explanation is a distinct test set (§6, §3.5), and for each expert, Ob-HEx performs better than the baseline on most of the metrics.

4.1 Analysis on base hierarchical model

The $\hat{S}$ scores from Ob-HEx can be used to visualize the focus points of the hierarchical model. So to analyze the focus points of the base models and see how the focus point varies between them, we use



(a) XLNet+BiGRU (b) LEGAL-BERT+BiGRU

Figure 3: Final $\hat{S}$ score (Eq. 3) distribution with Ob-HEx@k=0.1 on ILDC$_{Expert}$.

Ob-HEX, and plot (Figure 3) the final $\hat{S}$ score (Eq. 3) distribution obtained by Ob-HEX@k=10% from the base level (l=1) of the hierarchy of base models (§3.2) over ILDC$_{Expert}$. Both the base models focus more on the final part of the documents, even though LEGAL-BERT+BiGRU was trained on the full document length. XLNet+BiGRU also puts more emphasis on the first 25% and 40-70% of the document's length compared to LEGAL-BERT+BIGRU's emphasis on the first 25% and 85-95% of the document's length. While the score distributions vary heavily between the models, from 0.5-1.01 for XLNet+BiGRU compared to 0.3-0.7 and 1-1.01 for LEGAL-BERT+BiGRU.

5 Conclusion

We explore the problem of explaining a hierarchical model's prediction from long sequences in the legal domain and develop Ob-HEx based on the perturbations-based occlusion sensitivity of the trained model. For it, we develop a 'normalized weighted occlusion sensitivity score' to hierarchically score parts of a long input that is ranked and used as explanations. We adapt Ob-HEx to Hierarchical Transformers of Malik et al. (2021) and Prasad et al. (2022) and experiment with ILDC$_{Expert}$ to achieve new benchmarks over the previous methods. We also used Ob-HEX to analyze and interpret the focus points for our base model. Ob-HEx can be generalized and uses a layer-wise loss function (Occlusion-sensitivity Impact function) to score the occlusions, and uses the "normalized weighted occlusion-sensitivity score" to score the input fragments taking into the impacts from the previous layers. In future, we aim to use Ob-HEx as a selective-re-training strategy for the trained hierarchical models and analyze its effects on predictions. We also aim to implement Ob-HEx in other domains using hierarchical frameworks.

6 Limitations

Since explanations from the trained hierarchical model using Ob-HEx are based solely on the model and its input, they may contain sentences from parts of an input document that may be different from an expert's explanations. This is also due to the ability of the model to learn latent features which are not visible to an expert. And since explanations from an expert can be different from explanations from another expert (Malik et al. (2021)) no single explanation can be used as ground truth to measure the explanations from an extractive explanation algorithm like Ob-HEx. And as explanations from a single expert cannot be taken as absolute we do not rely only on the improvements of an individual expert. Hence, in our experiments, we use explanations from all the experts to have a varied set for comparison with our explanations and the ones from the baseline explanation algorithm (CJPE).

The explanations extracted from a hierarchical model using Ob-HEx reflect how the model looks at its inputs and the parts where it focuses most to make the predictions. Ob-HEx tries to approximate these focus points using the "impact function" and "normalized weighted occlusion sensitivity score" and ranks them to serve as an explanation. And since Ob-HEx doesn't train the model or make changes to its internal weights, the explanations we get using Ob-HEx are heavily model-dependent while Ob-HEx tries to best approximate the focus points of the model and extracts them.

We did not conduct any human evaluation to show the importance of those sentences extracted from Ob-HEx which are missed by CJPE. Because, to show with a strong claim that these excluded sentences are meaningful to the final prediction/-judgment we require a human legal expert, which we leave for future research.

Other existing explainability-based techniques (Zhou et al., 2015; Li et al., 2016; Fong and Vedaldi, 2017; Zhou et al., 2016) were "not applicable" or "complex to adapt" in their entirety, in our case of long legal documents & hierarchical frameworks (§2). This was our motivation behind Ob-HEx. Since only CJPE's algorithm existed for our problem setup we used it for comparison.

7 Ethical concerns

Our work aligns with the ethical consideration of the datasets (ILDC (Malik et al., 2021) and the hierarchical models used here for the experimentation and evaluation of our approach. We conform to the license under which the models and dataset were released (Malik et al. (2021)'s GPL-3.0 license) or shared with us (Prasad et al. (2022)'s GPL-3.0 license). We add certain points to this. The framework developed here is in no way to create an "explanatory" judge/lawyer or replace one in real life. Rather we develop Ob-HEx to analyze how deep-learning-based hierarchical models can be interpreted on legal documents to extract and provide legal professionals with patterns and insights that may not be implicitly visible. The methods developed here are in no way foolproof to predict and generate an explanatory response, and should not be used for the same in real-life settings (courts) or used to guide people unfamiliar with legal proceedings. The results from our framework should not be used by a non-professional to make high-stakes decisions in one's life concerning legal cases.

Acknowledgements

This work is supported by the LAWBOT project (ANR-20-CE38-0013) and HPC/AI resources from GENCI-IDRIS (Grant 2023-AD011013937R1). We also acknowledge the guidance from Dr. Rajesh Piryani (IRIT, Toulouse) during the review process.

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, Xiang Dai, Manos Fergadiotis, Prodromos Malakasiotis, and Desmond Elliott. 2022. An exploration of hierarchical attention transformers for efficient long document classification.

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.

Ruth C. Fong and Andrea Vedaldi. 2017. Interpretable explanations of black boxes by meaningful perturbation. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3449–3457.

Will Landecker, Michael D. Thomure, Luís M. A. Bettencourt, Melanie Mitchell, Garrett T. Kenyon, and Steven P. Brumby. 2013. Interpreting individual classifications of hierarchical networks. In *2013 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, pages 32–38.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. Understanding neural networks through representation erasure. *CoRR*, abs/1612.08220.

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.

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 Kumar 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)*, pages 2362–2374, Toronto, Canada. Association for Computational Linguistics.

Christoph Molnar. 2022. *Interpretable Machine Learning*, 2 edition.

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.

Vitali Petsiuk, Abir Das, and Kate Saenko. 2018. RISE: randomized input sampling for explanation of black-box models. In *British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018*, page 151. BMVA Press.

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.

Nishchal Prasad, Mohand Boughanem, and Taoufiq Dkaki. 2023a. Exploring semi-supervised hierarchical stacked encoder for legal judgement prediction. *ArXiv*, abs/2311.08103.

Nishchal Prasad, Mohand Boughanem, and Taoufiq Dkaki. 2023b. IRIT_IRIS_C at SemEval-2023 task 6: A multi-level encoder-based architecture for judgement prediction of legal cases and their explanation. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 686–692, Toronto, Canada. Association for Computational Linguistics.

Nishchal Prasad, Mohand Boughanem, and Taoufiq Dkaki. 2024. Exploring large language models and hierarchical frameworks for classification of large unstructured legal documents. In *Advances in Information Retrieval*, pages 221–237, Cham. Springer Nature Switzerland.

Gabrielle Ras, Ning Xie, Marcel van Gerven, and Derek Doran. 2022. Explainable deep learning: A field guide for the uninitiated. *J. Artif. Int. Res.*, 73.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Xingxing Zhang, Furu Wei, and Ming Zhou. 2019. HIBERT: Document level pre-training of hierarchical bidirectional transformers for document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5059–5069, Florence, Italy. Association for Computational Linguistics.

B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. 2016. Learning deep features for discriminative localization. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2921–2929, Los Alamitos, CA, USA. IEEE Computer Society.

Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. 2015. Object detectors emerge in deep scene cnns. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

A Description of Ob-HEx's implementation to base hierarchical model (Hierarchical Transformers)

Using Algorithm 1, we describe Ob-HEx adapted to trained hierarchical transformer models in §3.2, and detail the steps involved.

We start from the top-level M (l=2) of the hierarchical model to find the highly sensitive chunks (steps 2-14), for a document. Here, I=E. We calculate the probability output from M. Since M is at the top level of the hierarchy we take its absolute prediction as the absolute-predicted

Algorithm 1 Ob-HEx: Hierarchical Transformers

Require: From 3.2, Select T, M from 1^{st} & 2^{nd} level of hierarchy respectively. $k = \%$ of sentences to choose.

1: **for** all documents **do**
2: Divide the document into chunks $\{c_i | 1 \leq i \leq n\}$.
3: $E \leftarrow$ Extract all chunk embeddings from T.
4: $O_E \leftarrow M(E)$, probability output.
5: $P \leftarrow$ absolute-predicted class label from O_E
6: $\hat{L}_E \leftarrow 0$, impact with itself $L(P, P)$
7: **for** chunk c_i in E **do**
8: Mask c_i embedding.
9: $O_{E(i)} \leftarrow M(\{E|c_i\})$, output after masking c_i.
10: $\hat{L}_{E(i)} \leftarrow L(O_{E(i)}, P)$
11: $\hat{S}_{E(i)} \leftarrow \hat{S}_{E(i)}(1, \hat{L}_{E(i)}, \hat{L}_E)$ (Eq. 3)
12: **end for**
13: $N_E \leftarrow$ concatenate all $(c_i, \hat{S}_{E(i)})$.
14: $\hat{N}_E \leftarrow$ Sort N_E in descending order of $\hat{S}_{E(i)}$.
15: **for** i, (c, s) in $\hat{N}_E$ **do**
16: $O_c \leftarrow T(c)$, probability output from T
17: $\hat{L}_c \leftarrow L(O_c, P)$
18: Split c into sentences, $\{s_j | 1 \leq j \leq \text{m}\}$.
19: **for** s_j in c **do**
20: Mask s_j.
21: $O_{c(j)} \leftarrow T(\{c|s_j\})$, output after masking s_j.
22: $\hat{L}_{c(j)} \leftarrow L(O_{c(j)}, P)$
23: $\hat{S}_{c(j)} \leftarrow \hat{S}_{c(j)}(s, \hat{L}_{c(j)}, \hat{L}_c)$ (Eq. 3)
24: $A_{score} \leftarrow$ concatenate all $(i, s, \hat{S}_{c(j)})$.
25: **end for**
26: **end for**
27: Sort A_{score} in descending order of $\hat{S}_{c(j)}$.
28: $A_{score}[k] \leftarrow$ keep the top $k\%$ sentences.
29: $A_{score}[k] \leftarrow$ rearrange in the order of (i,s).
30: **end for**

class label P=$P_I^{t=2}$=$P_I^{l=2}$ and take the self-impact score as 0 (Step 2-6). We mask/occlude the chunks and calculate their impact score using the impact function $\hat{L}^l_{I(j)}$=$BCE_{loss}(O^l_{I(j)}, P)$ (§3.2) and then their "normalized weighted occluded sensitivity scores" (Eq. 2) concerning the whole document i.e. self-impact score (steps 8-11). Since this is the top level we use 1 as the score weight (s^l_I §2). We sort the accumulated scores in order of their sensitivity score (i.e. higher value is given more importance).

To rank the sentences (steps 15-28) we iteratively start from the highest-scored chunk and take its probability output from the fine-tuned transformer T (from level l=1) to calculate its impact function score w.r.t P (step 17). We then split this chunk (c) into sentences and iteratively mask/occlude a sentence s_j inside the chunk to calculate its 'normalized weighted occluded sensitivity score' ($\hat{S}_{c(j)}$) (steps 19-24). To weigh the overall importance of each sentence of this chunk as compared to the sentences belonging to other chunks, we weigh the impact shift of s_j with the sensitivity score s of c from the previous level (l=2) of hierarchy. We store the sentences along with their chunk number and sensitivity score in A_{score}. We sort A_{score}, ranking in the order of $\hat{S}_{c(j)}$. Since this is the base level (l=1) of the hierarchy we stop and take the top $k\%$ sentences. To arrange the sentences with their sequential occurrence in the document we arrange $A_{score}[k]$ according to the chunk number i and the sentence in the chunk. These sentences serve as the explanation for a document's prediction. The time complexity is model dependent, and is $O(n^2)$ here, due to the quadratic complexity of the fine-tuned transformer (T) used, where asymptotically n is the average length of all the documents for a batch.

B Distribution of evaluation results

B.1 Long explanations

Figure 4 shows the box plot of the results on long explanations for the respective evaluation metrics (§3.3).

B.2 Short explanations

Figure 5 shows the box plot of the results on short explanations for the ROUGE-1, ROUGE-2 and Jaccard similarity metrics (§3.3). The box plot on short explanations for BERT-Score is shown in Figure 1.

B.3 Brief explanations

Figure 6 shows the box plot of the results on brief explanations for the respective evaluation metrics (§3.3).

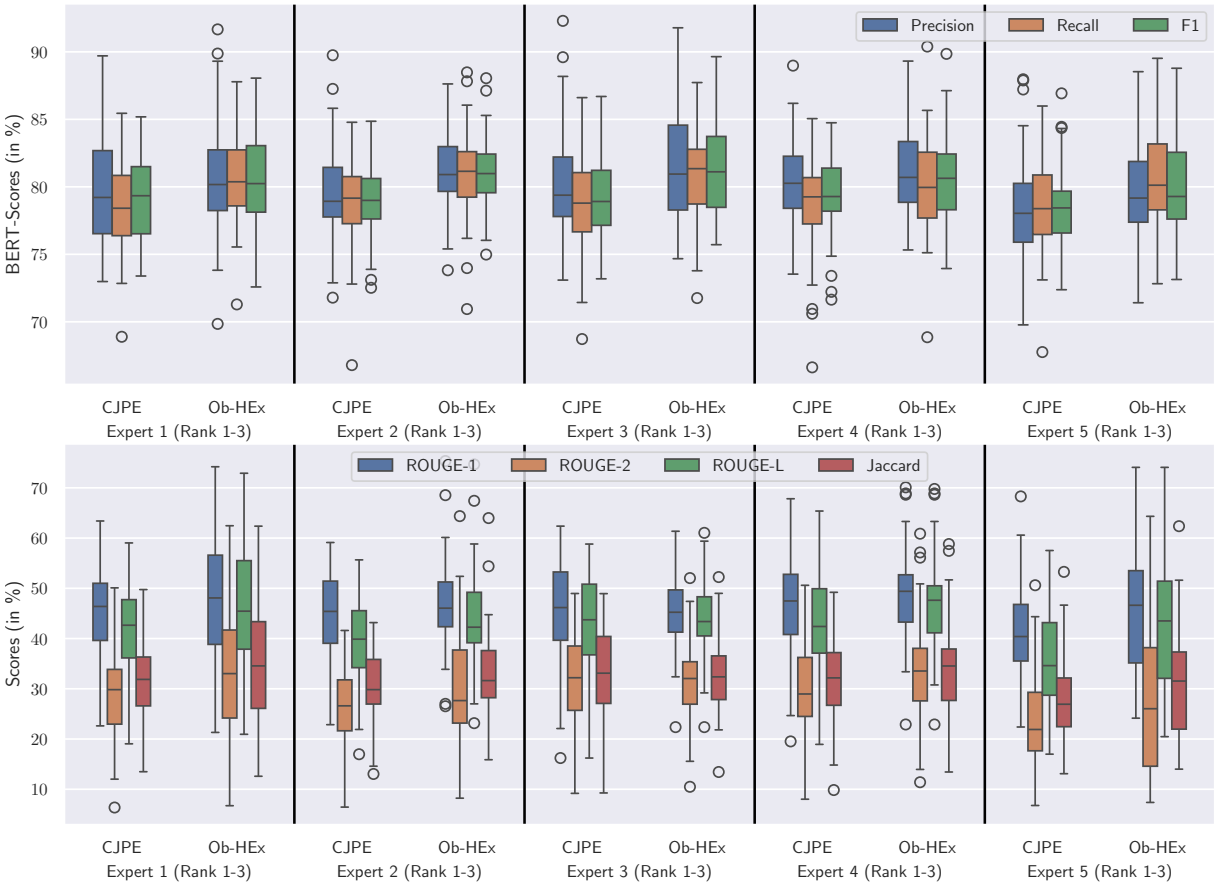


Figure 4: Box plot of evaluation metric scores from CJPE & Ob-HEx@k=0.15 vs experts. (XLNet+BiGRU). This plot shows the first quartile, third quartile and the median.

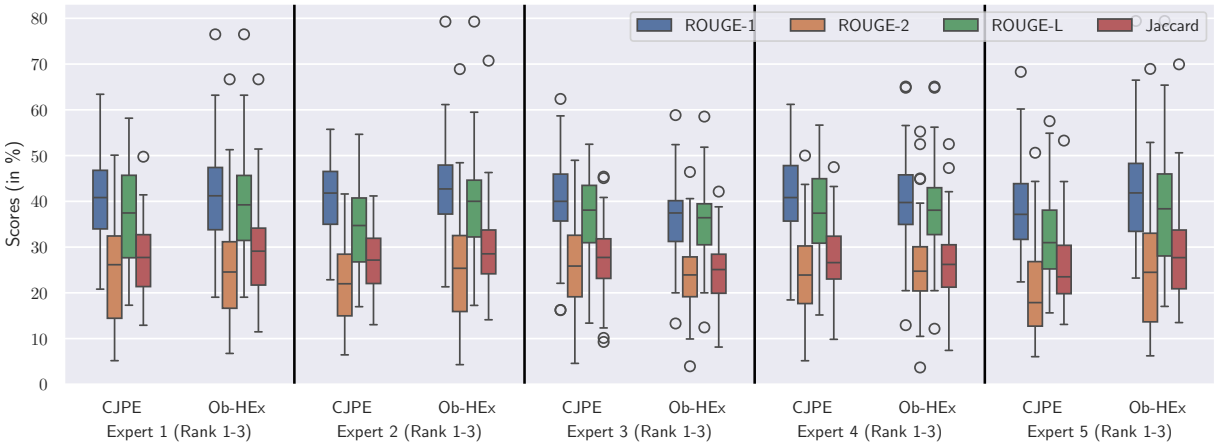


Figure 5: Box plot of ROUGE scores and Jaccard similarity, for top 512 words, from CJPE & Ob-HEx@k=0.1 vs experts. (XLNet+BiGRU). This plot shows the first quartile, third quartile and the median.

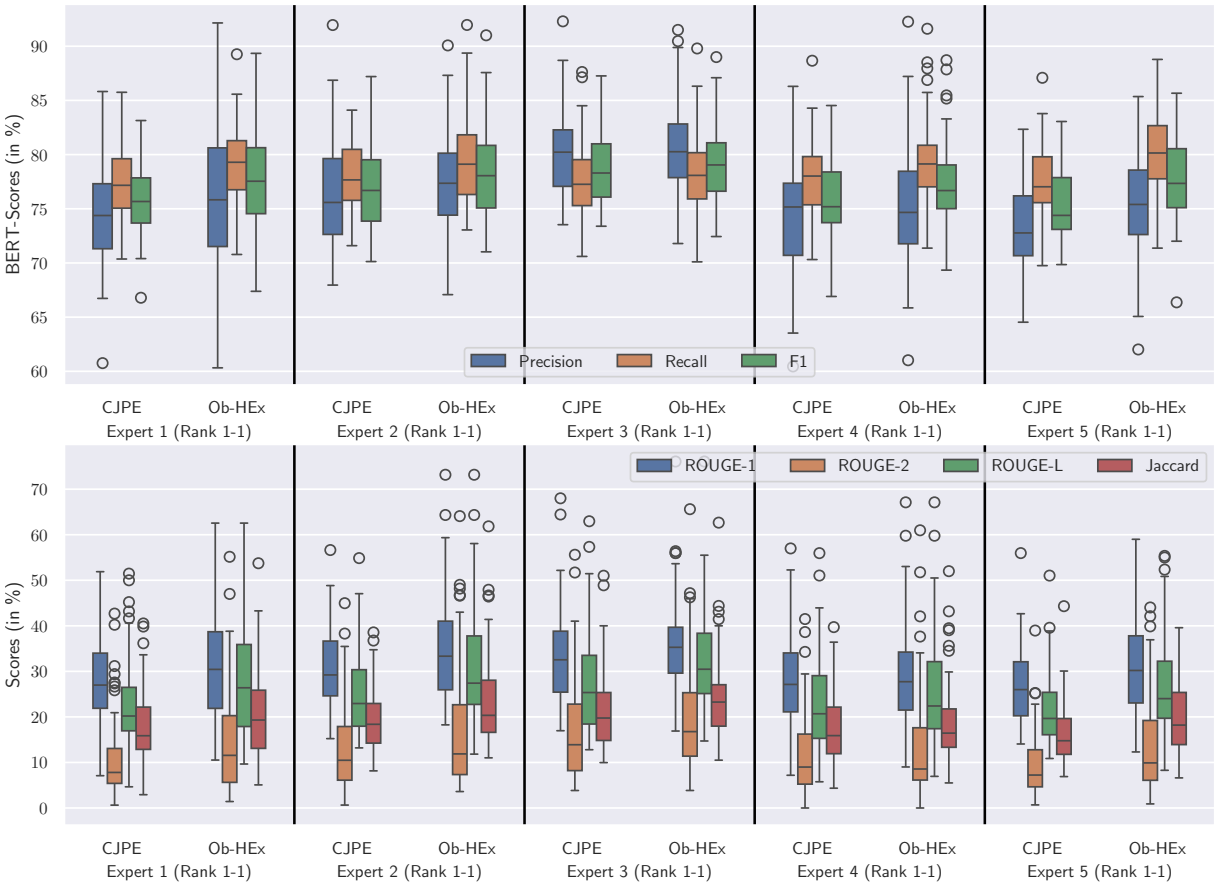


Figure 6: Box plot of evaluation metric scores, for top 256 words, from CJPE & Ob-HEx@k=0.1 vs experts. (XLNet+BiGRU). This plot shows the first quartile, third quartile and the median.