Explaining Finetuned Transformers on Hate Speech Predictions using Layerwise Relevance Propagation

Ritwik Mishra[1 [0000−0001−7058−0037]], Ajeet Yadav[1 [0009−0009−1944−6284]], Rajiv Ratn Shah[1 [0000−0003−1028−9373]], and Ponnurangam Kumaraguru[2 [0000−0001−5082−2078]]

[1] Indraprastha Institute of Information Technology, Delhi
[2] International Institute of Information Technology, Hyderabad
ritwikm@iiitd.ac.in ajeet19010@iiitd.ac.in
rajivratn@iiitd.ac.in pk.guru@iiit.ac.in

Abstract. Explainability of model predictions has become imperative for architectures that involve fine-tuning of a pretrained transformer *encoder* for a downstream task such as hate speech detection. In this work, we compare the explainability capabilities of three post-hoc methods on the HateXplain benchmark with different *encoders*. Our research is the first work to evaluate the effectiveness of Layerwise Relevance Propagation (LRP) as a post-hoc method for fine-tuned transformer architectures used in hate speech detection. The analysis revealed that LRP tends to perform less effectively than the other two methods across various explainability metrics. A random rationale generator was found to be providing a better interpretation than the LRP method. Upon further investigation, it was discovered that the LRP method assigns higher relevance scores to the initial tokens of the input text because fine-tuned *encoders* tend to concentrate the text information in the embeddings corresponding to early tokens of the text. Therefore, our findings demonstrate that LRP relevance values at the input of fine-tuning layers are not a good representative of the rationales behind the predicted score.

Keywords: LRP · LIME · SHAP · Hate Speech · Explainability

1 Introduction

Neural networks have gained extensive use in diverse applications such as natural language processing, speech recognition, and image recognition. Despite their widespread applicability, a significant critique of Deep Neural Networks is their

Disclaimer: This study includes quotes of text considered profane or offensive to illustrate the model's workings but does not reflect the authors' views. The authors condemn online harassment and offensive language.

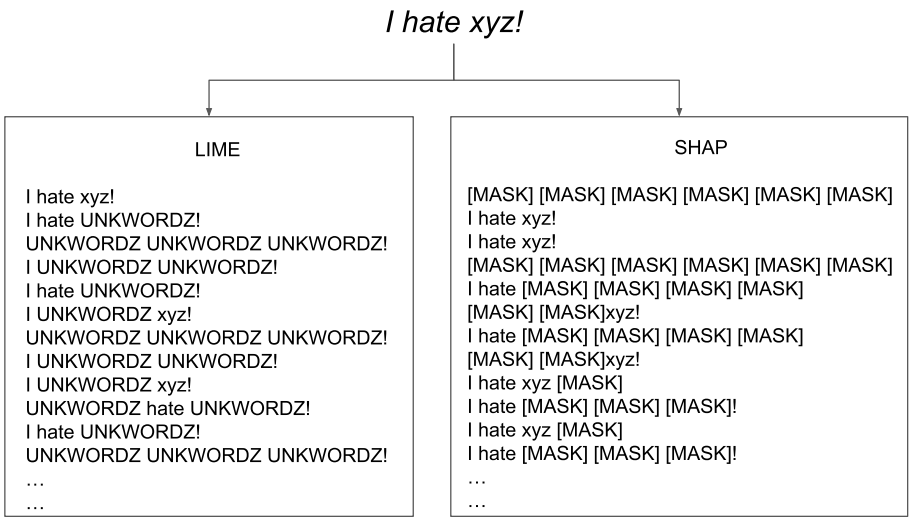

Fig. 1. Manipulation of text in LIME and SHAP techniques. Observe that LIME employs word-level masking within the provided string using the `UNKWORDZ` token, while SHAP employs subword-level masking using the `[MASK]` token. In the SHAP method, every altered text is fed to a fine-tuned model, and the relevance of subwords is calculated by the resulting change in model predictions. However, for the LIME method, a linear model is trained based on the modified text and its corresponding model output. This linear model is subsequently used to assign relevance values to each word.

opaque nature, which renders it challenging to comprehend how they arrive at their predictions [11]. Furthermore, there have been reports suggesting that these models may exhibit biases towards individuals of particular races [17], genders [26, 13], or ethnicity [1].

We used HateXplain benchmark [22], to evaluate the effectiveness of various post-hoc explainability methods. The benchmark includes not only gold class labels but also gold rationales. These gold labels are binary annotations provided by human annotators, classifying a given text as either containing hate speech (1) or not containing hate speech (0). The objective of hate speech detection revolves around predicting a score closer to 1 if the text includes hate speech and closer to 0 otherwise. Gold rationales are exclusively available for texts that annotators classify as containing hate speech. Originally introduced by [32], a gold rationale consists of a binary label vector with a length matching the number of words in the respective text. Therefore, our research inquiry for this study is as follows: *How much explainability LRP provides on the HateXplain benchmark compared to other post-hoc methods?*

In this study, we present a comparison of three Explainable AI (XAI) methods: Local Interpretable Model-agnostic Explanations (LIME) proposed by [25], SHapley Additive exPlanations (SHAP) by [19], and Layer-wise Relevance Prop-

agation (LRP) presented by [6]. These methods were selected for their ability to generate post-hoc explanations (rationales) for models that are trained without access to explicit gold *rationales*.

The LIME method constructs a linear model by altering the provided text and generating additional samples in its vicinity. This learned model is subsequently employed to predict relevance values for each input feature. Conversely, the SHAP method utilizes partial dependency plots to compute Shapley values (as a proxy for relevance values) for each input feature. Drawing from game theory, a Shapley value for an input feature is established based on the alteration in model output when the input feature is present or absent in the manipulated text. Both approaches modify the given text to calculate relevance values for the input features. LIME treats words as input features, whereas SHAP considers subwords when the input is text and the model is a transformer-based fine-tuned model. Figure 1 illustrates the distinct ways text manipulation is carried out in LIME and SHAP.

To the best of our knowledge, no previous studies have utilized LRP or SHAP methods to assess the explainability of transformer-based hate speech prediction models on the HateXplain benchmark. Our study is the first to compare the three aforementioned XAI methods on HateXplain benchmark, and we implemented LRP method as a constituent of this work.

The present study concentrates on prediction models for detecting hate speech that employs pretrained transformer-based encoders (or simply *encoders* henceforth) to generate text embeddings. Such models have become popular because they have exhibited state-of-the-art performances in hate speech detection [2, 20]. Furthermore, many works have emphasized the importance of interpretability in transformer-based architectures [8, 26, 28]. To the best of our knowledge, LRP method has not been applied to hate speech prediction models that utilizes pretrained *encoders*.

The LRP method operates by backpropagating relevance values. Specifically, the output layer of the hate speech model is used to determine the relevance values, which are then propagated backward through the network. The relevance value for a node j in layer L is determined by considering three factors: (a) the relevance values of all nodes in the succeeding layer $(L + 1)$, (b) the learned parameters between L and $L + 1$, and (c) the activation values in layer L. We refer readers to [24] for a detailed explanation of LRP.

Numerous previous studies have emphasized the societal implications of Explainable AI (XAI). The significant applications of XAI have been adequately outlined in the exhaustive survey conducted by [11]. Furthermore, a comparative investigation of two XAI techniques provides future researchers with an empirical rationale for selecting a particular method based on their task. Our research endeavors to investigate which method generates superior rationales and to present the insights derived from these rationales.

We release[3] our implementation of the LRP method for transformer-based text classification models. It provides word-level rationales for a predicted class

[3] https://github.com/ritwikmishra/hateXplain-metrics-calculation

in a multi-class classification setting. to the best of our knowledge, it is the first implementation of LRP method for fine-tuned transformer architectures.

2 Related Works

The literature has widely employed the LIME method to explain hate speech predictions [3, 23]. However, using the LRP method to explain hate speech predictions has not been common. Karim et al. [16] have employed the LRP method to explain hate speech predictions for the *Bengali* language, but their model was based on the Long Short-Term Memory (LSTM) architecture, a variant of Recurrent Neural Networks (RNN). Previously, Arras et al. [4] used the LRP method to explain sentiment predictions by a RNN model. Similarly, the LRP method's explanatory potential has been demonstrated in intent detection with Bidirectional LSTMs [15]. The LRP method has been applied to transformer-based Neural Machine Translation (NMT) models to analyze the contributions of source and target tokens in the translation process [30]. However, to the best of our knowledge, there is no prior work that implements LRP to explain class predictions from a fine-tuned transformer-based model.

Our decision to employ the LRP method in this study is rooted in its ability to produce post-hoc explanations for generated predictions. Furthermore, it has been observed that LRP yields meaningful explanations in various tasks, including question classification and semantic role labeling [9]. The relevance values derived from LRP have also found utility in refining pretrained word embeddings [29]. Moreover, the LRP method has been applied in Layerwise Relevance Visualization (LRV) for sentence classifiers based on Graph Convolution Networks (GCN) [27].

Numerous studies have analyzed the performance of two or more XAI methods. In comparison to the LIME method, SHapley Additive exPlanations (SHAP) [19] has shown to provide better explanations for disease classification [12]. However, for various models from the finance domain, LIME rationales were found to be more stable than SHAP [21]. Balkir et al. [7] proposed scores based on *necessity* and *sufficiency* to explain the predictions of a hate speech detection model. After comparing the performance of LIME and SHAP with their proposed scores, LIME failed to generate relevant rationales for false-positive predictions. In a sentiment analysis task, Jørgensen et al. [14] compared the rationales generated by SHAP and LIME. It was observed that SHAP is more successful in selecting relevant spans from the text, whereas LIME rationales align better with the ranking of words in human rationales.

3 Experimental Setup

In this study, we used the *transformers* library [31] to load various *encoders*. We used the Ferret [5] library to predict rationales by LIME and SHAP methods because it is built on top of the official implementation of these methods and the tool generates subword-level rationales. We calculated word-level rationales

```
>>> from transformers import AutoTokenizer
>>> tokenizer = AutoTokenizer.from_pretrained("bert-base-cased")
>>> print(tokenizer.tokenize('I hate xyz!'))
['I', 'hate', 'x', '##y', '##z', '!']
>>> print(tokenizer('I hate xyz!',add_special_tokens=False).word_ids())
[0, 1, 2, 2, 2, 3]
```

Fig. 2. Python code illustrating the output of the `.word_ids()` function from Huggingface (transformers) tokenizers. The provided example text is *I hate xyz!*. It can be observed that the word *xyz* is tokenized into three subwords: *x*, *##y*, and *##z*. Hence, the output of the `.word_ids()` function includes three instances of the index 2.

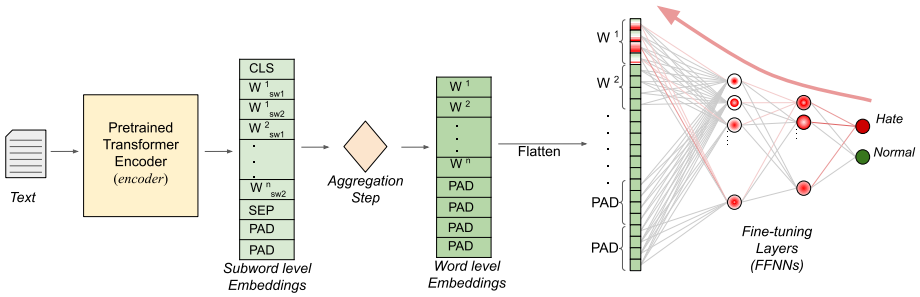

Fig. 3. Overall architecture of Hatespeech prediction model and relevance calculation using LRP method. The confidence score for the *Hate* label is highlighted in red. The predicted values are adopted as relevance scores at the output layer and subsequently propagated backward up to the initial fine-tuning layers (denoted by the red arrow), following the formulations introduced in the work by [24].

from subword-level rationales using the `.word_ids()` function from Huggingface tokenizers. The output of the `.word_ids()` function is depicted in Figure 2. We implemented different variants of LRP for the fine-tuning layers of the hate speech classifier due to the unavailability of any such tool. To test the validity of our LRP implementation, we ensured that the sum of relevance values remains consistent across each layer during the relevance backpropagation.

As shown in Figure 3, we generated word-level embeddings by computing an unweighted average of the corresponding subword embeddings. This approach was adopted due to the presence of groundtruth rationales at the word level rather than the subword level. As a result, embeddings of special tokens (e.g., CLS and SEP) that indicate sentence boundaries were omitted, as there are no groundtruth rationales available for such tokens.

The flattened word-level embeddings are then passed through the fine-tuning layers, which consist of multiple Feed-forward Neural Networks (FFNNs). We employed three linear layers with a dropout rate of 0.1 and relu activation applied in between. The relevance values of the flattened word embeddings are summed up to obtain the relevance value of the corresponding word. Due to the difficulty in backpropagating relevances through the transformer architecture containing

multi-head attention, we calculate relevance values only till the input layer of the fine-tuning module in this work. We implemented the following LRP variants by using formulations of Montavon et al. [24].

- **LRP-0**: It redistributes relevance in proportion to how much each input contributed to the activation of the neurons. Relevance of node j in layer L is calculated as:
$$R_j = \sum_k \frac{a_j w_{j,k}}{\sum_{j=0}^{j} a_j w_{j,k}} R_k \tag{1}$$
where k is the number of nodes in layer $L+1$.
- **LRP-ϵ**: This enhancement of the basic LRP-0 rule adds a small positive term ϵ in the denominator:
$$R_j = \sum_k \frac{a_j w_{jk}}{\epsilon + \sum_0^j a_j w_{jk}} R_k \tag{2}$$
When the contributions to the activation of neuron k are weak or inconsistent, the role of ϵ is to absorb some importance.
- **LRP-γ**: This enhancement of the basic LRP-0 rule adds a small positive term γ in the denominator:
$$R_j = \sum_k \frac{a_j.(w_{jk} + \gamma w_{jk}^+)}{\sum_0^j a_j.(w_{jk} + \gamma w_{jk}^+)} \cdot R_k \tag{3}$$
The parameter γ controls by how much positive contributions are favored. As γ increases, negative contributions start to disappear.

In their work, Montavon et al. [24] suggested utilizing gradients to compute relevance values. We discovered that gradient-based relevance values would not yield the desired results if bias is enabled in the fine-tuning layers. We refer readers to Appendix A for further details. Our LRP implementation supports the absence or presence of bias in the fine-tuning layers.

In all three explainability methods, a token is classified as relevant (1) or not-relevant (0) based on a threshold on its calculated relevance value. Similar to [22], we set the threshold value to 0.5 for all the methods considered in this study.

For the experimental investigation, we utilized two different *encoders*, namely *bert-base-cased* (BERT) by [10] and *roberta-base* (RoBERTA) by [18]. These encoders were chosen due to their prevalence in the literature for hate speech detection in English [2, 20]. To explore the impact of the encoder on rationale prediction, we trained our hate speech prediction model while keeping the underlying encoder frozen or allowing it to be fine-tuned. All models were trained for 10 epochs with an *encoder* learning rate of 5e-7 and fine-tuning layers learning rate of 1e-6. None of the models exhibited signs of either overfitting or underfitting the data.

The explainability metrics used in this study are classified into two categories: (a) Plausibility, and (b) Faithfulness. The former measures the extend

Encoder	Method	Performance		Explainability				
				Plausibility			Faithfulness	
		Accuracy↑	**Macro-F1↑**	**IOU F1↑**	**Token F1↑**	**AUPRC↑**	**Compr.↑**	**Suff.↓**
$BERT_{f-t}$	LRP	74 (74±0)	72 (71.7±0.6)	0.10 (0.11±0.0)	0.17 (0.178±0.0)	0.45 (0.469±0.01)	0.11 (0.12±0.0)	0.22 (0.217±0.01)
	LIME			**0.26 (0.25±0.01)**	**0.30 (0.30±0.0)**	**0.64 (0.64±0.0)**	**0.30 (0.29±0.01)**	0.09 (0.11±0.01)
	SHAP			**0.26 (0.26±0.01)**	**0.30 (0.30±0.01)**	**0.64 (0.64±0.0)**	0.29 (0.29±0.01)	0.09 (0.11±0.01)
$BERT_{fr}$	LRP	67 (67.6±1.1)	63 (63.7±1.2)	0.13 (0.13±0.0)	0.23 (0.23±0.0)	0.49 (0.50±0.01)	0.07 (0.08±0.01)	0.09 (0.10±0.01)
	LIME			0.14 (0.15±0.01)	0.24 (0.25±0.01)	0.54 (0.54±0.0)	0.09 (0.10±0.01)	0.08 (0.09±0.01)
	SHAP			0.17 (0.18±0.0)	0.26 (0.27±0.01)	0.56 (0.57±0.01)	0.10 (0.10±0.0)	0.08 (0.09±0.01)
$RoBERTA_{f-t}$	LRP	**75 (75.7±0.6)**	**73 (73.7±0.6)**	0.11 (0.12±0.01)	0.17 (0.18±0.02)	0.46 (0.47±0.01)	0.11 (0.11±0.0)	0.27 (0.26±0.01)
	LIME			0.24 (0.24±0.0)	0.27 (0.27±0.0)	0.61 (0.61±0.0)	0.07 (0.07±0.0)	0.07 (0.07±0.0)
	SHAP			0.24 (0.23±0.01)	0.26 (0.26±0.0)	0.61 (0.61±0.0)	0.07 (0.07±0.0)	0.06 (0.06±0.0)
$RoBERTA_{fr}$	LRP	67 (67.3±0.6)	63 (63.7±1.2)	0.13 (0.13±0.0)	0.23 (0.23±0.0)	0.49 (0.48±0.01)	0.05 (0.05±0.0)	0.06 (0.06±0.0)
	LIME			0.15 (0.15±0.0)	0.24 (0.24±0.0)	0.55 (0.55±0.0)	0.0 (0.01±0.01)	**0.01 (0.01±0)**
	SHAP			0.16 (0.16±0.0)	0.25 (0.26±0.01)	0.55 (0.56±0.01)	0.0 (0.0±0.0)	**0.01 (0.01±0.0)**
Random		50	50	0.10	0.23	0.47	0.27	0.26

Table 1. A comparison of different post-hoc rationale generation methods (LIME, SHAP, and LRP) on hate speech prediction architectures with different *encoders*. The architectures where the underlying encoder was fine-tuned during the training phase are represented by the subscript $f-t$, whereas architectures where the encoder was frozen during training, are represented by the subscript fr. We report numbers on the official test set of the HateXplain benchmark. The numbers inside round brackets represent the mean and std over 3-fold cross-validation. An upward arrow signifies that higher values are preferable, while a downward arrow indicates that lower values are preferable. It is observed that fine-tuned RoBERTA achieves the best Performance metrics whereas fine-tuned BERT achieves the best Explainability metrics.

to which the predicted rationales are similar the gold rationales annotated by humans while the later does not rely on gold rationales and expresses the sensitivity of the underlying model with respect to the predicted rationales. Under the plausibility category, different metrics like Intersection over Union (IOU) F1, Token F1, and Area Under Precision Recall Curve (AUPRC) are used. In contrast, faithfulness encompasses two metrics namely comprehensiveness and sufficiency. It's important to note that, unlike other metrics, lower sufficiency values are preferable. Readers are referred to the Hatexplain paper [22] for a detailed explanation of the definitions of the explainability metric used in this study.

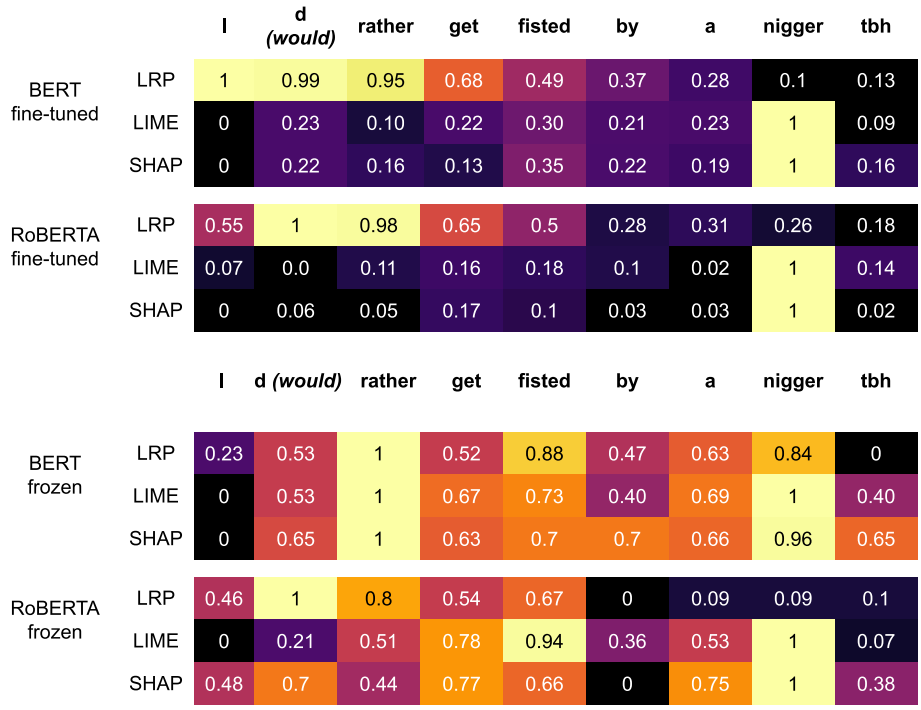

Fig. 4. Visualization of the predicted rationales by different methods on an example from the test set of the HateXplain benchmark. It can be seen that LRP rationales on fine-tuned *encoders* tend to give high relevance values to the early tokens, whereas LIME and SHAP focus more on the profane word.

4 Results

The performance of various encoders under different training paradigms is presented in Table 1. Our experimentation showed a lack of substantial performance variation among the different LRP variants. Therefore, we present results based on the LRP-0 variant in this study. The results of different LRP variants are presented in Appendix B.

Our analysis shows that the rationales predicted by LIME exhibit similar performance to SHAP across almost all explainability metrics. Moreover, the explainability power of LRP rationales on fine-tuned *encoders* was less than that of a random rationale generator. Additionally, we observed that while the plausibility scores of LIME and SHAP decrease for architectures with frozen encoders, they increase for LRP.

After a qualitative analysis of the rationales predicted by LRP, LIME, and SHAP for fine-tuned *encoders*, we observed that while LIME and SHAP predicted high relevance values for profane words, the LRP method predicted high relevance values on the early tokens of the sentence. Figure 4 illustrates the rel-

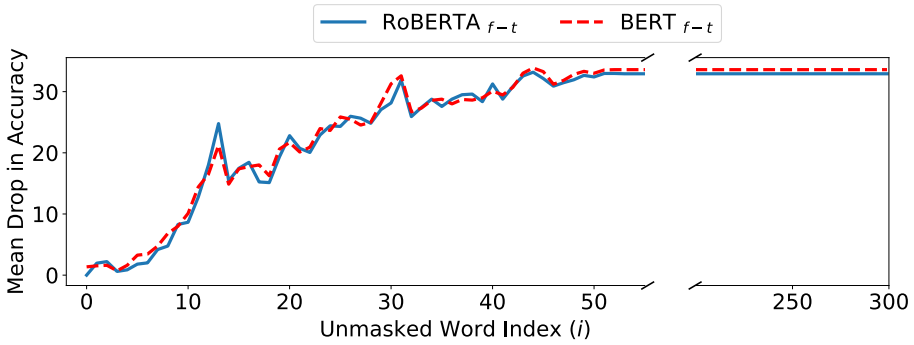

Fig. 5. Mean drop in accuracy over 3-fold cross-validation when word embeddings of i^{th} word are fed to fine-tuning layers while rest of the embeddings are masked to zero. It can be seen that drop in accuracy is negligible (y-axis $\rightarrow$ 0) when the embeddings of the first few tokens (x-axis $\rightarrow$ 0) are fed to the fine-tuning layers.

evance values of different methods on a sentence predicted as hate speech by all the architectures based on fine-tuned *encoders* and the human annotators of HateXplain benchmark.

Since relevance values by LRP represent the contribution of that node while making the prediction, high relevance on early tokens of the sentence indicates that the embeddings of early tokens primarily contribute towards the model prediction. To find out the importance of the embeddings corresponding to a token (i), we set the embeddings corresponding to the rest of the tokens $(\{0, 299\} - \{i\})$ as zero. Figure 5 illustrates that drop in accuracy of architectures with fine-tuned *encoders*. We infer that when encoders are fine-tuned during the training phase, they learn to concentrate all the sentence information in the embeddings of the first few tokens only.

Among the sentences which were predicted as containing hate speech, the LRP method predicts first token as relevant in 75% of sentences[4] in BERT$_{f-t}$ architecture, and 86% of sentences in RoBERTA$_{f-t}$ architecture.

5 Conclusion

Transformers-based architectures have shown state-of-art performances in various tasks ranging from vision to language. However, explainability in such deep neural architecture becomes of utmost importance since feature extraction happens in an end-to-end manner. In this work, we attempt to measure the explainability power of the LRP method using different *encoders* for hate speech detection. We noticed that LIME performs similarly to SHAP across the majority of the explainability metrics in the HateXplain benchmark. However, rationales predicted by LRP led us to conclude that fine-tuning a pretrained transformer

[4] Averaging over the three test sets in 3-fold cross-validation.

based *encoder* results in such a model that concentrates the entire text information in the embeddings of its first few tokens. Therefore, the LRP relevance values until the fine-tuning layers may not provide an accurate representation of the underlying semantic rationale behind the predicted score. Nonetheless, it also suggests that fine-tuned *encoders* exhibit a property of feature space reduction that can be used to justify text visualizations based on the embeddings corresponding to early tokens of the text.

6 Limitations and Future Work

Since the implemented LRP is limited to the fine-tuning layers of transformed-based models, the present study cannot explain the concentration of sentence information in the embeddings of its early tokens. Furthermore, we undertake a performance comparison of various explanation techniques in the context of hate speech detection, utilizing a single benchmark dataset, with the understanding that our findings are constrained to this particular dataset. To make analogous assertions in different domains and with other datasets, additional research is essential. Our goal is to expand upon the formulations presented in the work by Voita et al. [30] and conduct backpropagation of relevance values through the *encoder* block, thereby enhancing our understanding of the fine-tuning procedures applied to these *encoders*.

Acknowledgements

Ritwik Mishra extends his appreciation to the University Grant Commission (UGC) of India, as he receives partial support through the UGC Senior Research Fellowship (SRF) program. Rajiv Ratn Shah acknowledges the partial assistance received from the Infosys Center for AI (CAI) and the Center of Design and New Media (CDNM) at IIIT Delhi.

References

1. Ahn, J., Oh, A.: Mitigating language-dependent ethnic bias in bert. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. pp. 533–549 (2021)
2. Aluru, S.S., Mathew, B., Saha, P., Mukherjee, A.: A deep dive into multilingual hate speech classification. In: Machine Learning and Knowledge Discovery in Databases. Applied Data Science and Demo Track: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part V. p. 423–439. Springer-Verlag, Berlin, Heidelberg (2020). https://doi.org/10.1007/978−3 − 030 − 67670 − $4_2$6, *https* : //*doi.org*/10.1007/978 − 3 − 030 − 67670 − $4_2$6
3. Aluru, S.S., Mathew, B., Saha, P., Mukherjee, A.: A deep dive into multilingual hate speech classification. In: Machine Learning and Knowledge Discovery in Databases. Applied Data Science and Demo Track: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part V. pp. 423–439. Springer (2021)

4. Arras, L., Montavon, G., Müller, K.R., Samek, W.: Explaining recurrent neural network predictions in sentiment analysis. In: Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis. pp. 159–168. Association for Computational Linguistics, Copenhagen, Denmark (Sep 2017). https://doi.org/10.18653/v1/W17-5221, https://aclanthology.org/W17-5221
5. Attanasio, G., Pastor, E., Di Bonaventura, C., Nozza, D.: ferret: a framework for benchmarking explainers on transformers. In: Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations. Association for Computational Linguistics (May 2023)
6. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PloS one **10**(7), e0130140 (2015)
7. Balkir, E., Nejadgholi, I., Fraser, K., Kiritchenko, S.: Necessity and sufficiency for explaining text classifiers: A case study in hate speech detection. In: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 2672–2686. Association for Computational Linguistics, Seattle, United States (Jul 2022). https://doi.org/10.18653/v1/2022.naacl-main.192, https://aclanthology.org/2022.naacl-main.192
8. Bourgeade, T.: From Text to Trust: A Priori Interpretability Versus Post Hoc Explainability in Natural Language Processing. Ph.D. thesis, Université Paul Sabatier-Toulouse III (2022)
9. Croce, D., Rossini, D., Basili, R.: Auditing deep learning processes through kernel-based explanatory models. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 4037–4046 (2019)
10. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. CoRR **abs/1810.04805** (2018), http://arxiv.org/abs/1810.04805
11. Ding, W., Abdel-Basset, M., Hawash, H., Ali, A.M.: Explainability of artificial intelligence methods, applications and challenges: A comprehensive survey. Information Sciences (2022)
12. Dolk, A., Davidsen, H., Dalianis, H., Vakili, T.: Evaluation of lime and shap in explaining automatic icd-10 classifications of swedish gastrointestinal discharge summaries. In: Scandinavian Conference on Health Informatics. pp. 166–173 (2022)
13. Garimella, A., Amarnath, A., Kumar, K., Yalla, A.P., Anandhavelu, N., Chhaya, N., Srinivasan, B.V.: He is very intelligent, she is very beautiful? on mitigating social biases in language modelling and generation. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. pp. 4534–4545 (2021)
14. Jørgensen, R., Caccavale, F., Igel, C., Søgaard, A.: Are multilingual sentiment models equally right for the right reasons? In: Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP. pp. 131–141 (2022)
15. Joshi, R., Chatterjee, A., Ekbal, A.: Towards explainable dialogue system: Explaining intent classification using saliency techniques. In: Proceedings of the 18th International Conference on Natural Language Processing (ICON). pp. 120–127 (2021)
16. Karim, M.R., Dey, S.K., Islam, T., Sarker, S., Menon, M.H., Hossain, K., Hossain, M.A., Decker, S.: Deephateexplainer: Explainable hate speech detection in under-

resourced bengali language. In: 2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA). pp. 1–10. IEEE (2021)
17. Kwako, A., Wan, Y., Zhao, J., Chang, K.W., Cai, L., Hansen, M.: Using item response theory to measure gender and racial bias of a bert-based automated english speech assessment system. In: Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022). pp. 1–7 (2022)
18. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized BERT pretraining approach. CoRR **abs/1907.11692** (2019), http://arxiv.org/abs/1907.11692
19. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. Advances in neural information processing systems **30** (2017)
20. Maimaitituoheti, A.: Ablimet@ lt-edi-acl2022: A roberta based approach for homophobia/transphobia detection in social media. In: Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion. pp. 155–160 (2022)
21. Man, X., Chan, E.P.: The best way to select features? comparing mda, lime, and shap. The Journal of Financial Data Science **3**(1), 127–139 (2021)
22. Mathew, B., Saha, P., Yimam, S.M., Biemann, C., Goyal, P., Mukherjee, A.: Hatexplain: A benchmark dataset for explainable hate speech detection. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 14867–14875 (2021)
23. Mehta, H., Passi, K.: Social media hate speech detection using explainable artificial intelligence (xai). Algorithms **15**(8), 291 (2022)
24. Montavon, G., Binder, A., Lapuschkin, S., Samek, W., Müller, K.R.: Layer-wise relevance propagation: an overview. Explainable AI: interpreting, explaining and visualizing deep learning pp. 193–209 (2019)
25. Ribeiro, M.T., Singh, S., Guestrin, C.: " why should i trust you?" explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. pp. 1135–1144 (2016)
26. Sarat, P., Kaundinya, P., Mujumdar, R., Dambekodi, S.: Can machines detect if you're a jerk (2020)
27. Schwarzenberg, R., Hübner, M., Harbecke, D., Alt, C., Hennig, L.: Layerwise relevance visualization in convolutional text graph classifiers. In: Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13). pp. 58–62 (2019)
28. Szczepański, M., Pawlicki, M., Kozik, R., Choraś, M.: New explainability method for bert-based model in fake news detection. Scientific reports **11**(1), 23705 (2021)
29. Utsumi, A.: Refining pretrained word embeddings using layer-wise relevance propagation. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 4840–4846 (2018)
30. Voita, E., Sennrich, R., Titov, I.: Analyzing the source and target contributions to predictions in neural machine translation. 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). pp. 1126–1140. Association for Computational Linguistics, Online (Aug 2021). https://doi.org/10.18653/v1/2021.acl-long.91, https://aclanthology.org/2021.acl-long.91
31. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T.L., Gugger, S., Drame, M., Lhoest, Q., Rush, A.M.: Transformers: State-of-the-art natural language processing. In: Proceedings

of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. pp. 38–45. Association for Computational Linguistics, Online (Oct 2020), https://www.aclweb.org/anthology/2020.emnlp-demos.6
32. Zaidan, O., Eisner, J., Piatko, C.: Using "annotator rationales" to improve machine learning for text categorization. In: Human language technologies 2007: The conference of the North American chapter of the association for computational linguistics; proceedings of the main conference. pp. 260–267 (2007)

A Appendix

Montavon et al. [24] mentions the generic rule to calculate the relevance value of node j in layer L as:

$$R_j = \sum_k \frac{a_j.\rho(w_{jk})}{\epsilon + \sum_0^j a_j.\rho(w_{jk})} R_k \tag{4}$$

The relevance are backpropagated using the following four steps:

- Forward pass: $z_k = \epsilon + \sum_0^j a_j.\rho(w_{jk})$
- Division: $s_k = R_k/z_k$
- Backward pass: $c_j = \sum_k \rho(w_{jk}).s_k$
- Relevance: $R_j = a_j c_j$

The paper asserts that c_j can be represented by the gradients attached to a, denoted as c_j =`a.grad`. We intend to explain the reasoning behind it. To simplify the explanation, we assume that $\rho(w_{jk}) = w$.

1. We know that $c_j = w \times s$ where $dim(w) = (j, k)$ and $dim(s) = (k, 1)$.
2. We know that `z = w.forward(a)` which is equivalent to $w^T \times a$ where $dim(a) = (j, 1)$. Therefore $dim(z) = (k, 1)$.
3. We know that $s = R_k/z$ where $dim(R_k) = (k, 1)$. Therefore $dim(s) = (k, 1)$.
4. If we use gradients then `z*s.data` can be written as $t = (w^T \times a) \cdot s$ where $dim(t) = (k, 1)$.
5. When t is summed and the gradients are back-propagated using `.backward()`, the gradients will be attached to w and a both.
6. `a.grad` will be the differentiation of t with respect to a. Therefore, `a.grad`= $\partial t/\partial a = w \times s = c_j$.

The deliberate use of `s.data` by the authors is intended to prevent the flow of gradients into a through a new path, given the dependence of s on a. However, when biases are present in the neural network layers, equation 4 will contain biases in both the numerator and the denominator. As differentiation ignores the biases, the gradient attached to a will remain as $w \times s$. Nevertheless, in order to satisfy the LRP constraint that requires an identical sum of relevance values in each layer, the expression of c_j in the backward pass needs to be modified as follows:

$$c_j = \sum_k \left(\rho(w_{jk}) + \frac{b_k}{|j|a_j} \right) \cdot s_k$$

Hence, `a.grad` will not be equivalent to c_j when the fully-connected neural network layers have bias enabled.

Encoder	Method	Performance		Explainability				
				Plausibility			Faithfulness	
		Accuracy↑	Macro-F1↑	IOU F1↑	Token F1↑	AUPRC↑	Compr.↑	Suff.↓
BERT_{f-t}	LRP-0	74 (74±0)	72 (71.7±0.6)	0.10 (0.11±0.0)	0.17 (0.178±0.0)	0.45 (0.469±0.01)	0.11 (0.12±0.0)	0.22 (0.217±0.01)
	LRP-ϵ			0.10 (0.11±0.0)	0.16 (0.171±0.0)	0.45 (0.464±0.01)	0.11 (0.11±0.0)	0.22 (0.224±0.0)
	LRP-γ			0.10 (0.10±0.0)	0.16 (0.17±0.01)	0.45 (0.466±0.01)	0.11 (0.12±0.0)	0.22 (0.222±0.01)
BERT_{fr}	LRP-0	67 (67.6±1.1)	63 (63.7±1.2)	0.13 (0.13±0.0)	0.23 (0.23±0.0)	0.49 (0.50±0.01)	0.07 (0.08±0.01)	0.09 (0.10±0.01)
	LRP-ϵ			0.14 (0.14±0.0)	0.23 (0.23±0.0)	0.48 (0.49±0.01)	0.08 (0.08±0.0)	0.09 (0.09±0.0)
	LRP-γ			0.14 (0.14±0.01)	0.23 (0.23±0.0)	0.49 (0.50±0.01)	0.08 (0.08±0.01)	0.09 (0.09±0.01)
RoBERTA_{f-t}	LRP-0	**75 (75.7±0.6)**	**73 (73.7±0.6)**	0.11 (0.12±0.01)	0.17 (0.18±0.02)	0.46 (0.47±0.01)	0.11 (0.11±0.0)	0.27 (0.26±0.01)
	LRP-ϵ			0.11 (0.12±0.01)	0.16 (0.18±0.02)	0.46 (0.47±0.01)	0.11 (0.11±0.0)	0.20 (0.20±0.0)
	LRP-γ			0.11 (0.12±0.01)	0.16 (0.18±0.02)	0.46 (0.47±0.01)	0.12 (0.11±0.01)	0.20 (0.20±0.0)
RoBERTA_{fr}	LRP-0	67 (67.3±0.6)	63 (63.7±1.2)	0.13 (0.13±0.0)	0.23 (0.23±0.0)	0.49 (0.48±0.01)	0.05 (0.05±0.0)	0.06 (0.06±0.0)
	LRP-ϵ			0.14 (0.14±0.0)	0.23 (0.23±0.0)	0.49 (0.48±0.01)	0.05 (0.05±0.0)	0.05 (0.06±0.01)
	LRP-γ			0.13 (0.14±0.01)	0.23 (0.23±0.0)	0.49 (0.48±0.01)	0.05 (0.05±0.0)	0.06 (0.06±0.0)

Table 2. An evaluation of various LRP variants on hate speech prediction architectures with different *encoders*. Results are presented based on the official HateXplain benchmark's test set, and the figures in parentheses indicate the mean and standard deviation derived from 3-fold cross-validation. Architectures with fine-tuned encoders are denoted by the subscript $f-t$, while those with frozen encoders are indicated by the subscript fr. Notably, there is minimal variation in results among different LRP variants.

B Appendix