Textual Insights : A Deep Dive into Cyberbullying Detection Through Interpretable AI

Avinash Chaluvadi - CMPSCI 6390

May 8, 2024

Contents

1 Abstract

Interpretable Machine Learning (IML) is a sub-field of machine learning dedicated to making machine learning models, particularly deep neural networks (DNN's) or transformer based, more understandable and explainable to humans. As the use of powerful yet complex DNN's has grown, IML has emerged as a crucial research area. Despite achieving state-of-the-art performance across numerous domains, the intricate inner workings of DNN's make them challenging to interpret and explain.

This project report explores various techniques to enhance the interpretability of machine learning models. It begins with an overview of key interpretability methods, including Integrated Gradients, LIME (Local Interpretable Model-agnostic Explanations), and Shapley Values. Each of these techniques serves a distinct purpose: analyzing feature or word attributions, generating localized explanations, and determining feature importance, respectively.

The report then presents a case study where these interpretability methods are applied to two distinct BERT language models—one trained on raw data and another on oversampled data. The analysis compares the inner workings and explanations provided by these models, illuminating how the choice of training data affects model behavior and interpretability. By shedding light on these differences, the case study highlights the impact of data preparation techniques on the model's ability to provide understandable and reliable explanations.

In conclusion, this report provides a comprehensive comparison of two BERT models trained with different data strategies and showcases how interpretability techniques like Integrated Gradients, LIME, and Shapley Values can be used to analyze and compare their inner workings. These insights and methodologies can serve as a starters guide for developing more transparent deep learning models.

Motivation: Interpretable Machine Learning (IML) has gained significant attention as a research direction in recent years, leading to numerous open questions in the field. Broadly, IML aims to achieve two objectives: (1) develop and train models that are inherently interpretable, or (2) uncover the underlying mechanisms of highly complex prediction models in a meaningful and informative manner.

2 Dataset

The **"Cyberbullying Dataset"** For the experimental study, I downloaded a dataset, namely the Cyberbullying dataset, from the Kaggle website here. This dataset contains 47,692 samples across six labels: religion, age, ethnicity, gender, not_cyberbullying, and other_cyberbullying. Each of these labels has approximately 7,800 samples. The dataset is pre-processed to handle any missing values and standardize figures before being split into training, validation, and testing sets. Typically, 60% of the data is used for training the model, 20% for validation, and 20% for testing. This split ensures a comprehensive evaluation of the model's effectiveness in identifying different forms of cyberbullying under varied conditions.

2.1 Overview of Data

Sentence	**Label**
All the girls in high school that bullied me	age_cyberbullying
Admit it, everyone has a little bit of racism in them	ethnicity_cyberbullying
Look at how the Democrats support their cultural sites	religion_cyberbullying
I need to just switch to an organization-based GitHub, but I don't want to pay $25/month	other_cyberbullying
Let's talk about what it means to be a victim	not_cyberbullying

Table 1: Sample sentences with their corresponding labels

3 Data Processing

The first step in data processing is data cleaning, which involves removing unnecessary words or characters such as emojis, hashtags, extra spaces, non-English characters, numbers, single-character words, repeated punctuation's, and web-based URLs. After this thorough cleaning, it was noted that tweets belonging to the "other_cyberbullying" class were removed due to the class being highly unbalanced and overly "generic" compared to other classes. After these steps, the maximum observed sentence length in the dataset was 81 tokens. This pre-processing stage helped ensure that only relevant, well-structured data was included for further analysis and model training, leading to more accurate classification results.

3.1 Data Splitting

In the preparation of the dataset, the complete data was partitioned into training, validation, and testing segments. Specifically, 60% of the data was designated for training, while the remaining 40% was equally divided between validation and testing.

3.2 Word Embeddings

The research paper introduces two techniques for converting words into vector representations:

1. **Word2Vec**: This technique employs neural networks to learn continuous word embeddings based on context. Word2Vec provides static word embeddings, meaning that each word has a single representation, regardless of the surrounding context.
2. **BERT Tokenizer**: This technique uses a pre-trained transformer model to tokenize text into sub-word tokens and generate contextualized word embeddings. The BERT tokenizer offers rich contextual representations that adapt to the specific usage of each word, making it a powerful tool for understanding and representing textual data.

3.3 Overview of Data Post-Vectorization

1. **Word2Vec:**

```
(tensor([101, 7087, 6340, 2048, 2304, 2611, ...]),
 tensor([1, 1, 1, 1, 1, 1, ...]))
```

2. **BERT Tokenizer:**

```
(tensor([101, 7087, 6340, 2048, 2304, 2611, ...]),
 tensor([1, 1, 1, 1, 1, 1, ...]),
 tensor(1))
```

4 Modelling and Training

In the original research paper, three different models were trained and evaluated. Each model assessed the ability to classify various forms of cyberbullying effectively, providing unique insights into model architectures and their application in the cyberbullying domain. The models include:

1. **Naive Bayes Baseline Classifier**: This baseline model served as a benchmark for comparison. It performed well on the dataset, achieving an overall accuracy of 87% across all classes. The Multinomial Naive Bayes algorithm from the scikit-learn package was used to implement this model.

2. **Custom LSTM with Attention RNN**: This architecture leveraged an LSTM network with attention mechanisms, implemented in PyTorch. Data was fed to the LSTM-based model through the PyTorch DataLoader with a batch size of 32, ensuring efficient data batching and processing. The training parameters included: **Learning Rate (LR):** 4e-4, **Dropout:** 0.5, **Epochs:** 10 The model achieved an overall accuracy of 93%, with some classes obtaining F1 scores exceeding 95%.

3. **BERT Transformer Model**: Utilizing the BERT transformer architecture, this model outperformed the previous methods with an overall accuracy of approximately 95% and F1 scores exceeding 96%. The model was a pre-trained BERT model from the Hugging Face library, fine-tuned for the cyberbullying classification task by adjusting the final layer's weights and biases. The training parameters included: **Optimizer:** AdamW, **Learning Rate (LR):** 5e-5, **Weight Decay:** 1e-8, **Batch Size:** 32 (using the PyTorch DataLoader), **Sequence Length:** 128 (sentences shorter than 128 tokens were zero-padded), **Epochs:** 2

NOTE: Data is shuffled, so results may vary each time. I selected the BERT model to generate human-understandable explanations and provide mechanistic interpretability of last layer

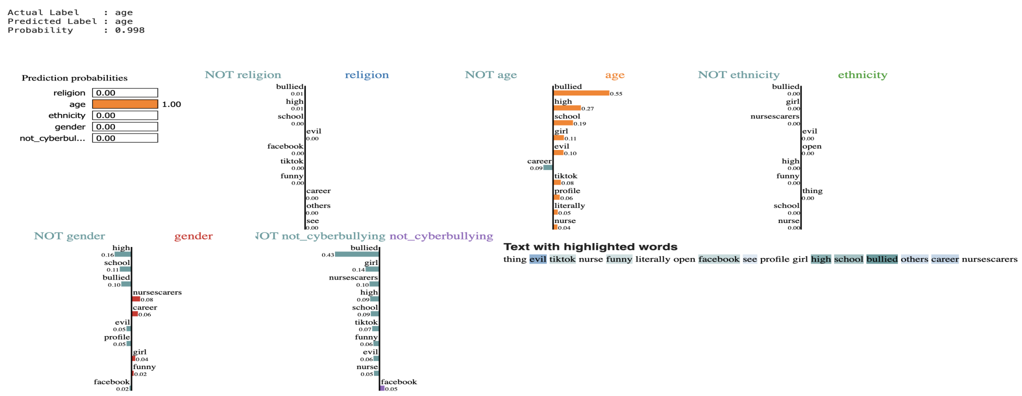
5 Model Explainability

To explain and understand individual predictions of BERT model, I utilized local model interpretation methods such as **LIME**, **Shapley values**, and the **Integrated Gradients** method. These techniques were applied to the fine-tuned BERT model to generate explanations for its predictions, offering insights into the influence of individual features. By leveraging these complementary methods, a more comprehensive understanding of how the BERT model arrives at its predictions was obtained.

5.1 LIME

LIME, or Local Interpretable Model-Agnostic Explanations, is a local model interpretation technique that employs local surrogate models to approximate the predictions of the underlying black-box model. Local surrogate models, such as linear regression or decision trees, are interpretable models used to explain the individual predictions of a black-box model[3]. LIME creates a surrogate model by generating a new dataset through perturbations around the data point of interest, allowing the surrogate to provide insight into the prediction of the original model.[4]

5.1.1 LIME Interpretability on Correct Explanation



```
Actual Label    : age
Predicted Label : age
Probability     : 0.998
```

Text with highlighted words

thing evil tiktok nurse funny literally open facebook see profile girl high school bullied others career nursescarers

Figure 1: LIME Explanation of a Correct Prediction

We can see in Figure 1 that the model accurately identified the tweet as belonging to the **age cyberbullying** category, with a probability of 0.998. Using 500 samples, LIME trained a surrogate model whose explanation aligns well with the predicted label. Key terms like ”bullied,” ”high,” ”school,” ”girl,” ”evil,” and ”tiktok” play a significant role in the classification process, as the surrogate model learned that these words contribute to the age category, excluding other categories.

Some terms, such as ”nursescarers” and ”career,” highlighted in red, were categorized under gender but with lower attribution scores, emphasizing their lesser importance. Overall, this analysis demonstrates how LIME provides insights into the model’s decision-making, revealing how words are attributed to particular labels for accurate classification

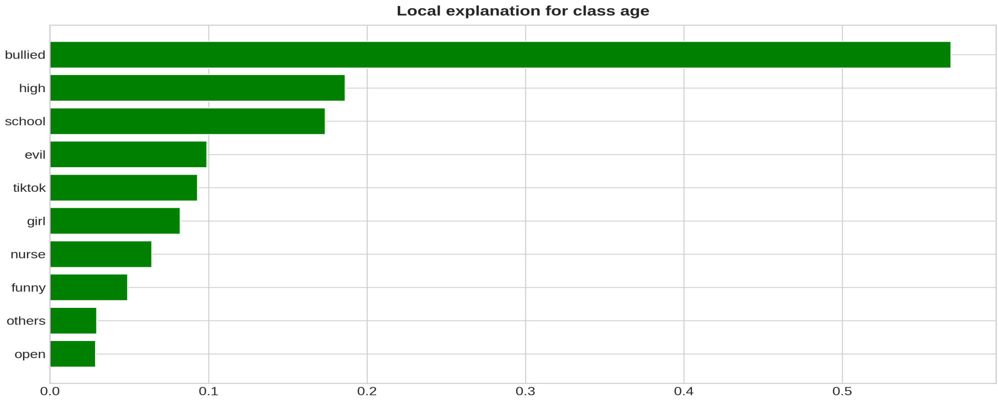


Figure 2: Attribution Scores for Age Label

Figure 2 provides a visual explanation using LIME’s `as_pyplot_figure` method for the **age** class. The bar chart displays the contribution (attribution scores) of specific words toward classifying the text as belonging to this category. Words like ”bullied,” ”high,” ”school,” ”tiktok,” and ”evil” have high positive scores, meaning their presence heavily influenced the model to classify the text in this category. There are no negative-score words displayed on the bar chart with a high score, indicating that the explanation aligns perfectly with the model’s prediction.

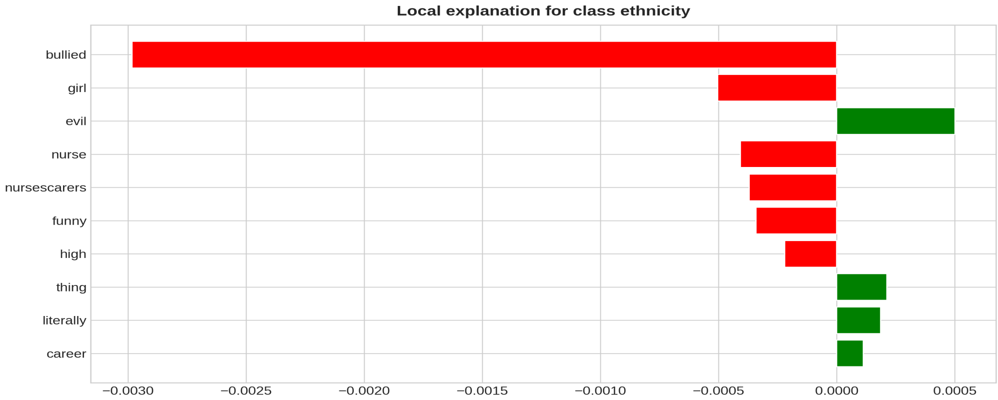
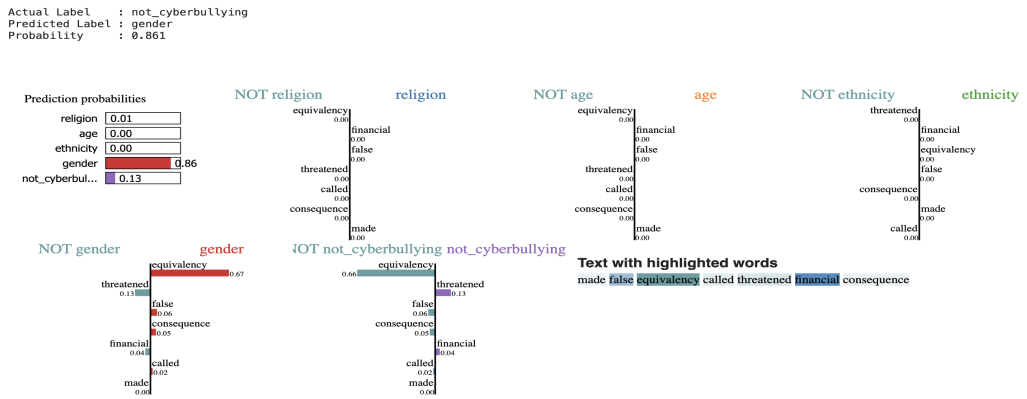


Figure 3: Attribution Scores for Ethnicity Label

Figure 3 illustrates that words like ”bullied,” ”girl,” and ”nurse” have strong negative scores for ethnicity, as the actual label is ”age.” The surrogate model identifies **evil** with positive scores for both age and ethnicity, indicating that the presence of ”evil” contributes positively to both categories. This alignment suggests that the explanation generated by LIME is consistent with the model’s predictions.

5.1.2 LIME Interpretability on Wrong Prediction



```
Actual Label    : not_cyberbullying
Predicted Label : gender
Probability     : 0.861
```

Figure 4: LIME Explanation of Misclassified Tweet

Figure 4 illustrates that the model inaccurately identified the tweet as belonging to the **gender cyberbullying** category, with a probability of 0.861. Using 500 samples, LIME trained a surrogate model whose explanation doesn't align with the actual label. Key terms like "equivalency" and "false" played significant roles in the classification process, as the surrogate model learned that these words contribute to the gender category, excluding other categories. However, the actual prediction is **not_cyberbullying**.

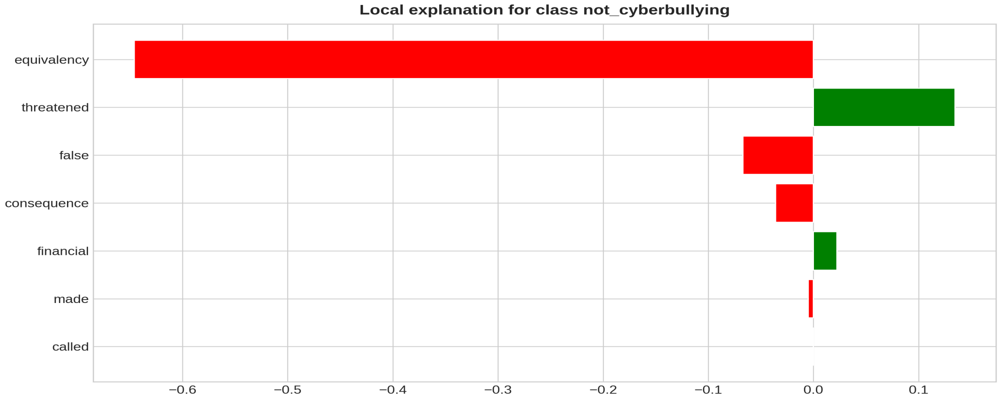


Figure 5: Attribution Scores for not_cyberbullying Label

Figure 5 illustrates that words like "equivalency" and "false" have strong negative scores for the **not_cyberbullying** category, where the actual label is "not_cyberbullying." Despite this, the surrogate model identified the provided sentence as belonging to the **gender** category. The surrogate model appears to interpret the "equivalency" and "false" features as indicative of the "gender" category.

5.2 SHAP

SHAP, or SHapley Additive exPlanations, is a is a local model interpretation technique. It uses Shapley values to quantify the contribution of each feature to the prediction of a model. SHAP provides both global and local interpretability by calculating the impact of individual features on a specific prediction and the overall behavior of the model.

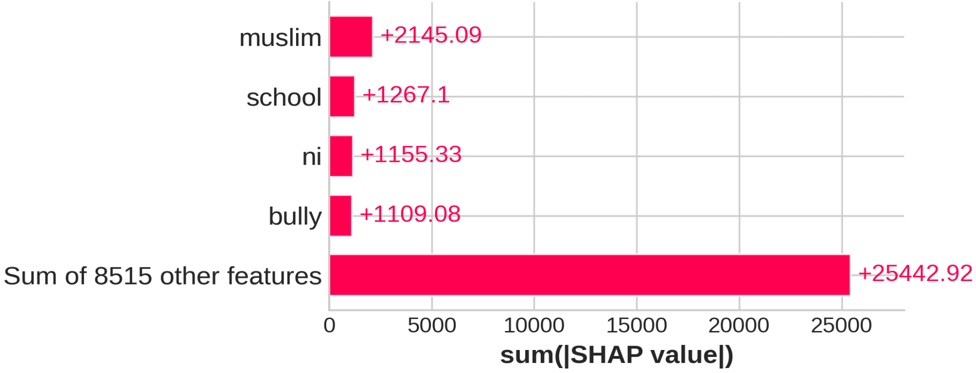


Figure 6: Top Features/SHAP Values for Religion

Figure 6 illustrates that for the label religion, based on a training sample size of 5000, the top SHAP value identified is **muslim** with a feature importance of +2145.09. Other features like "school," and "bully," also contribute significantly to the model's prediction with positive SHAP values. The visualization helps highlight key factors influencing the classification decisions, offering an interpretative view into the model's inner workings.

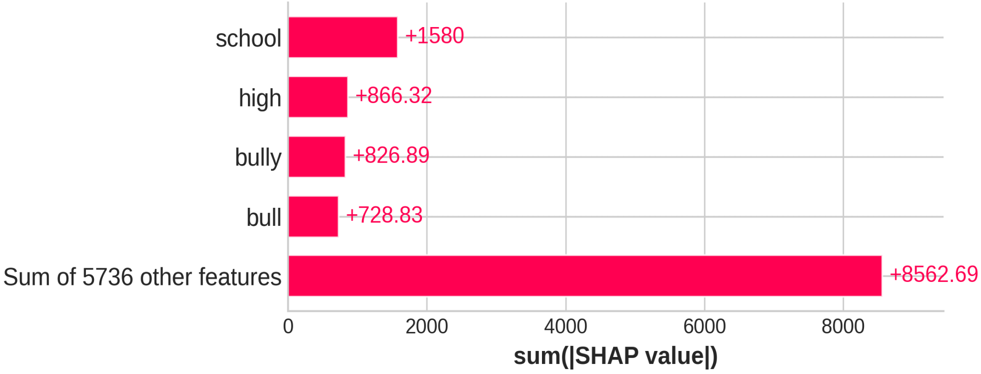


Figure 7: Top Features/SHAP Values for Age

Figure 7 illustrates that for the label age, based on a training sample size of 2000, the top SHAP value identified is **school**, with a feature importance of +1580. Other features like high and bully also contribute significantly to the model's prediction with positive SHAP values. School and bully are the second and fourth most important features in predicting the label religion.

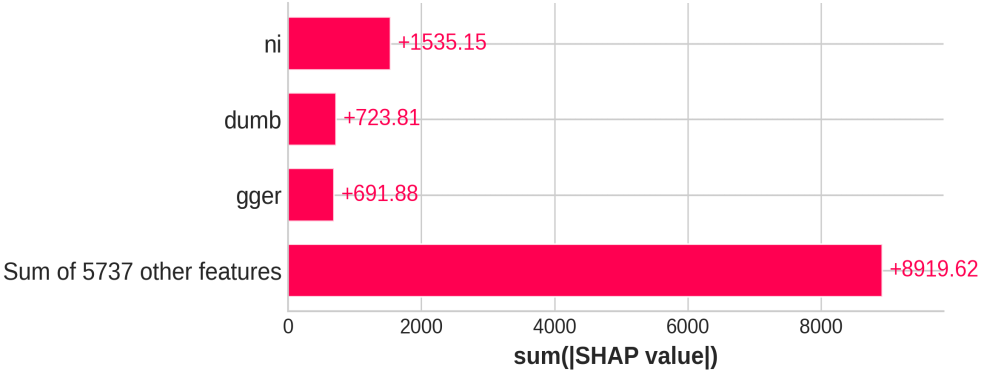


Figure 8: Top Features/SHAP Values for Ethnicity

Figure 8 illustrates that for the label ethnicity, based on a training sample size of 2000, the top SHAP value identified is **ni**, with a feature importance of +1535.15. Other features like "dumb" and "gger" also contribute significantly to the model's prediction with positive SHAP values. The word nigger is split into two separate tokens by BERT, resulting in two words with top feature importance's.

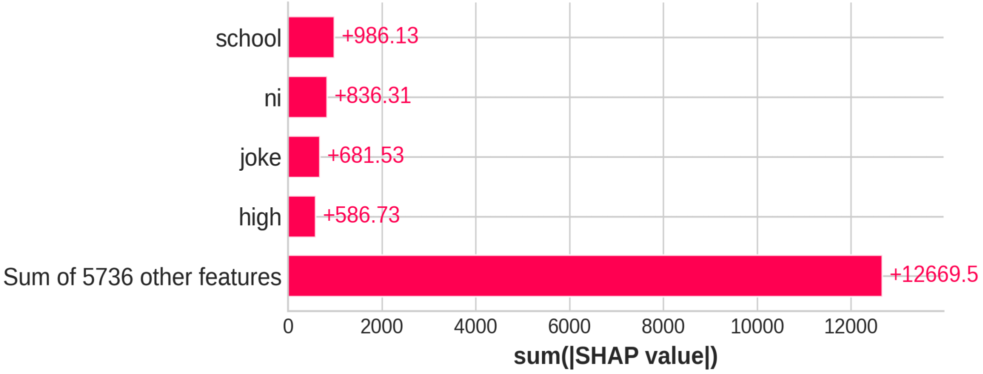


Figure 9: Top Features/SHAP Values for Gender

Figure 9 illustrates that for the label gender, based on a training sample size of 2000, the top SHAP value identified is **school**, with a feature importance of +986. Other features like ni and joke also contribute significantly to the model's prediction with positive SHAP values. The word ni is top important feature in ethnicity.

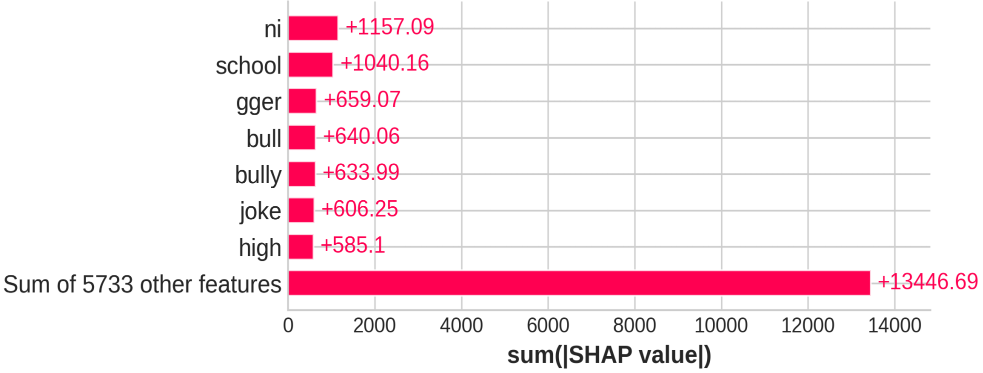


Figure 10: Top Features/SHAP Values for Not Cyberbullying

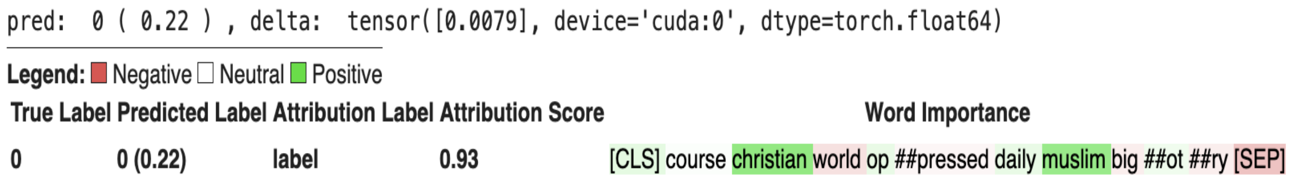
Figure 10 suggests that for the label not_cyberbullying, based on a training sample size of 2000, the top SHAP value identified is **ni**, which is the top feature among the gender and ethnicity categories.

Important Finding: Obtaining global importance values of words reveals the most significant words across various categories, but there is considerable overlap, as many of the top features belonging to the not_cyberbullying category are also found among the top features of other labels. Additionally, the SHAP technique performed poorly in identifying the feature importance scores for a single instance of explanation.

5.3 Integrated Gradients

An attribution method[2] evaluates the input data based on the model's predictions, attributing scores to each input signal or feature. Integrated Gradients is one such technique, which assigns scores by multiplying each feature by its gradient. The gradient serves as a signal to the neural network, guiding the adjustment of weights or coefficients during back-propagation [1].

5.4 Integrated Gradients Explanation for a Correct Prediction



```
pred:  0 ( 0.22 ) , delta:  tensor([0.0079], device='cuda:0', dtype=torch.float64)
```

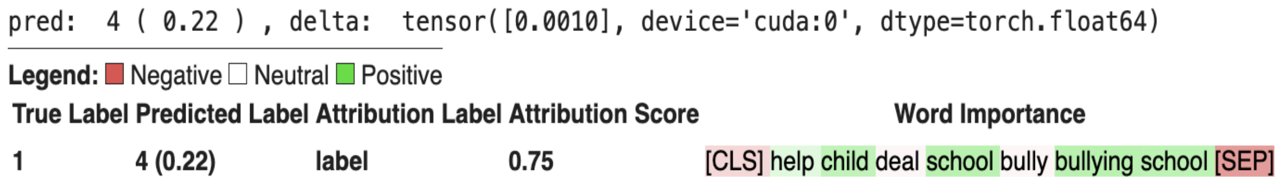
Legend: 🟥 Negative □ Neutral 🟩 Positive

True Label	Predicted Label	Attribution Label	Attribution Score	Word Importance
0	0 (0.22)	label	0.93	[CLS] course christian world op ##pressed daily muslim big ##ot ##ry [SEP]

Figure 11: Integrated Gradients Attribution Scores

Figure 11 demonstrates that the model's prediction aligns closely with the actual label. Terms such as ”christian” and ”muslim” positively contribute to the prediction, while ”oppressed” and ”daily” shows a slight positive influence. Conversely, ”world” contributes negatively to the final prediction. The Integrated Gradients method used 300 steps to calculate these attribution scores, revealing the varying degrees of influence each word has on the model's classification.

5.5 Integrated Gradients Explanation for a Wrong Prediction



```
pred:  4 ( 0.22 ) , delta:  tensor([0.0010], device='cuda:0', dtype=torch.float64)
```

Legend: 🟥 Negative □ Neutral 🟩 Positive

True Label	Predicted Label	Attribution Label	Attribution Score	Word Importance
1	4 (0.22)	label	0.75	[CLS] help child deal school bully bullying school [SEP]

Figure 12: Integrated Gradients Attribution Scores

Figure 12 shows that the model's prediction does not align with the actual label. Terms such as ”child,” ”school,” ”bullying,” and ”help” positively contribute to the prediction. However, despite containing words like ”child,” ”school,” and ”bullying,” which correspond to label ”age” the Integrated Gradients method classified the prediction as not_cyberbullying. Increasing the n_steps parameter may improve the prediction accuracy and attribution scores.

6 Mechanistic Interpretability

In the original code, the training data was over-sampled because the amount of data corresponding to the non_cyberbullying label was half that of other labels. Training was conducted using this augmented/over-sampled dataset. Analyzing the similarity between the weights of the final layer in two models, one with oversampling and one without (i.e., using raw data), provides insight into the impact of oversampling on the model's fine-tuning parameters. This analysis helps answer the question of whether oversampling significantly changes the last layer weights or if it only causes minimal changes. Only the final layer weights were considered because all attention head weights and other feed-forward layers were frozen, meaning their weights were unaffected during training.

6.1 Interpretation of BERT last layer weights with Oversampling

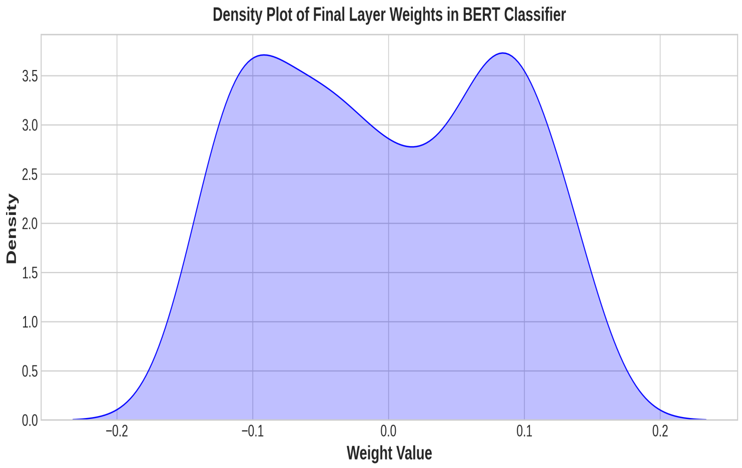


Figure 13: Density Plot of BERT Classifier's Final Layer Weights with Oversampling

Observations

1. The density plot has a smoother and more balanced shape, indicating that the weights are distributed more evenly.
2. The plot shows two primary peaks, suggesting the model is relying on two main groups of features or patterns for classification.

6.2 Interpretation of BERT last layer weights without Oversampling

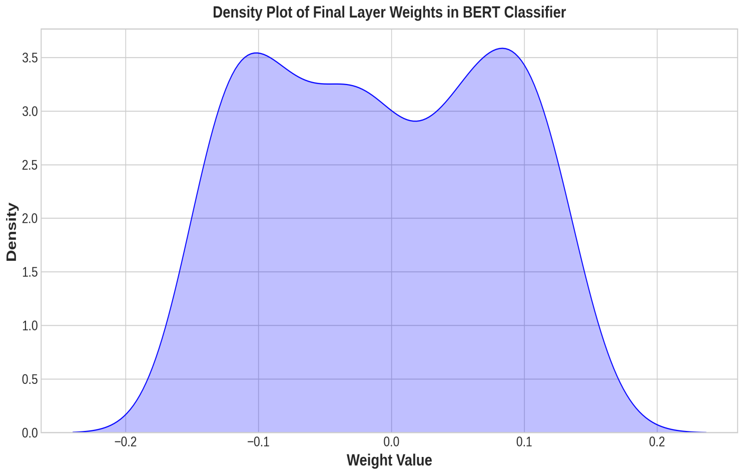


Figure 14: Density Plot of BERT Classifier's Final Layer Weights without Oversampling

Observations

1. This density plot shows a similar shape but with more pronounced peaks and valleys, indicating less smoothness in weight distribution..
2. The plot shows two primary peaks, suggesting the model is relying on two main groups of features or patterns for classification and the plots looks less balanced and more irregular.

6.3 Comparison of Final Layer Weights Using Cosine Similarity

To compare the learned representations in the final classification layer of the two BERT models, I first extracted the weights from this layer. The final classification layer is responsible for generating the output logits. The cosine similarity between the extracted weights was then calculated to analyze the alignment of the learned representations.

Observations: The computed cosine similarity score between the final layer weights of the two BERT models is -0.0489. This result suggests that the final layer weights of the two models capture different feature representations, reflecting distinct patterns learned during training. The slight negative correlation implies that the learned features are mildly opposed to each other.

Overall, this comparison shows that using different training strategies can impact the final representation of a neural network model's last layer, highlighting the importance of training data.

7 Challenges Faced

1. Initially, I encountered a challenge in loading the fine-tuned BERT model from the original paper. This issue was resolved by including a constructor that functions like a BertConfig in the original fine-tuned BERT model class. The constructor was missing in the original source code of the cyberbullying fine-tuned BertClassifier class, and adding it ensured proper configuration and loading of the model.

2. Initially, I encountered a challenge in integrating LIME with the BERT model due to the discrepancy in data formats. LIME requires word embeddings, while the BERT model's forward function accepts input_ids and token_type_ids as a tuple. To overcome this, I had to extract input_ids and token_type_ids beforehand and design a wrapper to pre-process the sentences of interest. This wrapper ensured the data was correctly formatted and passed to the model, allowing for seamless integration of LIME explanations with BERT predictions.

3. I encountered difficulties in obtaining attribution scores using the Integrated Gradients method with the pre-trained BERT model. The model only returns logits, whereas Integrated Gradients requires both attention and input embeddings generated from BERT. To resolve this issue, I developed a custom function to extract the necessary embeddings from the BERT model. This function generated the required embeddings, which were then passed to the Integrated Gradients method for accurate attribution scoring and meaningful explanations.

8 Conclusion

In this report, I compared various interpretable local explanation techniques such as LIME, SHAP, and Integrated Gradients. The findings demonstrate that most of these methods effectively generate human understandable explanations based on local regions of interest. The results demonstrated that these methods are capable of producing human-understandable explanations for specific instances, despite the challenges posed by the complexity of the data and the models used. Throughout this study, I have gained a deeper understanding of several explanation techniques, including how to apply LIME, SHAP, and Integrated Gradients to large transformer-based models like BERT. I also learned how to interpret BERT's attention heads, what meaningful explanations they can generate, and how to analyze the weights of the final layer. Further improvements include investigating why SHAP's global interpretations overlap across different labels, such as Age and Ethnicity, while still providing high feature attribution scores. Additionally, understanding BERT's layer weights at a granular level through a technique called probing can help identify which layers contribute the most to the model's predictions, offering greater insight into the interpretability of BERT models. In conclusion, this project not only advances my understanding of interpretable AI in the context of cyberbullying detection but also highlights critical areas for future research. By continuing to refine these interpretative techniques, reliability and utility in real-world applications can be enhanced, ultimately contributing to safer environments.

References

[1] Integrated gradients for nlp. https://databasecamp.de/en/ml/integrated-gradients-nlp, 2024. Accessed: 2024-05-08.

[2] Integrated gradients on bert using torch. https://github.com/sherlcok314159/Integrated-Gradients, 2024. Accessed: 2024-05-07.

[3] Hanjie Chen, Guangtao Zheng, and Yangfeng Ji. Generating hierarchical explanations on text classification via feature interaction detection. *arXiv preprint arXiv:2004.02015*, 2020.

[4] Dina Mardaoui and Damien Garreau. An analysis of lime for text data. In *International conference on artificial intelligence and statistics*, pages 3493–3501. PMLR, 2021.