Hate Speech Detection Using HateXplain Dataset

Qihua Huang, Yumin Li

1. Introduction

The Tiktok community norms forbid hate speech or behavior in an effort to foster a friendly and secure social atmosphere. As a result, hate speech needs to be recognized. In order to assist the Tiktok community with content assessment, the project aims to create a model that can more precisely identify and categorize hate speech based on the training and testing of the HateXplain dataset.
The HateXplain dataset includes 20,148 examples annotated for target groups (race, religion, gender, sexual orientation, and others), hate intensity (hate, offensive, or normal), and explanations, offering insightful information for studies on hate speech identification. We used 80% of the dataset for training, and 20% for validation and testing. Our model is based on bert-base-uncased. We used performance metrics like accuracy, bias, and explainability to assess and analyze the outcomes.

2. Traditional machine learning models

2.1 Data Preprocessing

At first, read the data from the `dataset.json` file, and read the segmentation information of the training, validation, and test sets from `post_ids_divisions.json` in accordance with the HateXplain (A Benchmark Dataset for Explainable Hate Speech Detection) dataset's instructions.

After that, the original data is loaded and converted into a Pandas DataFrame format for processing machine learning models.

Subsequently, the datasets for training, validation, and testing are generated based on pre-established divisions of the datasets, which operate as a basis for feature extraction and model training.

Based on statistical features, TF-IDF (Term Frequency-Inverse Document Frequency) vectorization is applied to convert text data into numerical features. TF-IDF reflects the importance of a word in a post relative to its occurrence across the entire corpus. Using LabelEncoder and loading classes.npy file to add labels to the data encoding.

2.2 Model Training

For multi-class classification applications, Logistic Regression and Linear Support Vector Machine (SVM) are the two selected classifiers since they are simple and effective.

Subsequently, the models were trained using the training set (X_train, y_train) in order to identify the relationships between the labels and features.

2.3 Model Evaluation

We calculate test accuracy, the macro F1-score, and the AUROC(Area Under the Receiver Operating Characteristic Curve) score in accordance with conventional procedures. Among them, the test accuracy means the Percentage of correct predictions on the testing set. And macro F1 Score considers the average F1 score across all classes. It provides a balanced measure of the model's precision and recall across different classes. ROC curves for each class are plotted to visualize the trade-off between true positive rate and false positive rate across different classification thresholds. Weighted-average AUROC is also reported to give insight into overall predictive performance.

2.4 Results and Discussion

Table 1 displays the linear SVM and logistic regression performance results.

Model	Performance		
	Accuracy	Macro F1 score	Weighted-average AUROC
Linear SVM	0.604	0.587	0.77
Logistic Regression	0.608	0.584	0.78

Table 1: Performance outcomes of traditional machine learning models

Figure 1 displays the ROC curves for each class of Linear SVM and Logistic Regression.

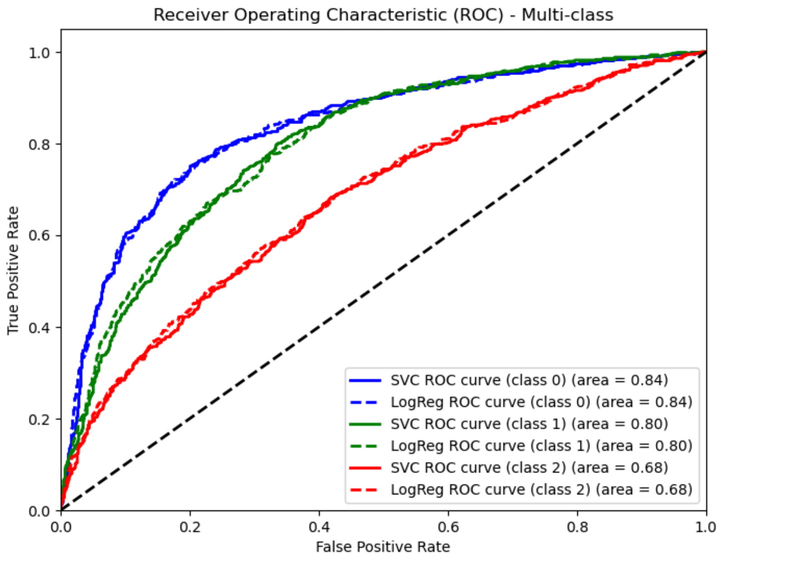


Figure 1: ROC curves for traditional machine learning models

The performance of such models is about the same as the least effective model reported in the initial study done by Mathew et al. (Mathew et al., 2021), suggesting that applying conventional machine learning methods to this problem yields no appreciable advantage. As a result, this project's next stage will involve researching deep learning models.

3. Deep Learning Models

3.1 Data Preprocessing

The first step in making sure the data is standardized, organized, and optimized for further modeling and analysis is data preparation. This section introduces the several steps that go into preparing textual data.

3.1.1 Tokenization
Posts are tokenized using the BERT tokenizer (bert-base-uncased).

3.1.2 Majority Voting for Labels and Target Groups
The dataset contains posts that are classified into three labels ("hatespeech" , "normal", "offensive"). Also, each post is identified a target group like race, religion, and gender. Consequently, by reviewing every post in the dataset, the annotations are totaled up using majority voting to identify the dominant labels and target groups.

3.1.3 Rationales Processing
Softmax scores are computed by processing the given rationales to determine the relative value of each token in the post.

3.1.4 Data Structuring
Each post is structured with the following components:

- post_id: Unique id for each post
- input_ids: Input tokens of each post
- target_group: Predominant target group derived from majority voting
- label: Predominant label derived from majority voting
- rationales: Softmax scores indicating token importance within the post

3.1.5 Data Saving
Then preprocessed data is divided according to file `post_id_divisions.json` and saved into distinct files (`test_data.pkl`, `train_data.pkl`, and `val_data.pkl`).

3.2 Model Training

3.2.1 Model Selection

As BERT(Bidirectional Encoder Representations from Transformers) has consistently demonstrated strong performance in text classification or other textual challenges, we began with the BERT base model and chose the bert-base-uncased model.

3.2.2 Model Architecture

The core of the model architecture is a custom implementation, named 'BertForMultiTaskClassificationAndTagging', builds upon the foundational BertModel from the transformers library.

This unique model is intended for multi-task learning and addresses two different tasks:

- Task1: Classification of Hate Speech
  Three classes of texts are distinguished in this task: "normal," "offensive," and "hatespeech." Based on the input text, it makes use of a classifier (classifier1) to predict the correct label.
- Task2: Classification of Target Groups
  Text must be categorized according to its affiliation with several target groups, such as "Miscellaneous," "Race," "Religion," "Gender," and "Sexual Orientation." It uses a second classifier (classifier2), just like Task 1, to determine which target group the text belongs to. We specify that each text in this assignment can only be a part of one community.

The model also includes a tagging mechanism that makes use of attention mechanisms to tag rationale. The attention ratings from the final layer of the BERT model (last_layer_attention) are collected and averaged.

3.2.3 Model Training

Both task 1 (classifying hate speech) and task 2 (classifying target groups) use nn.CrossEntropyLoss() as loss function. Apart from that, a new loss function named MaskedBCELoss is built for the tagging reason task. With the use of binary cross-entropy loss with masking, this loss function, MaskedBCELoss(), makes it easier to tag rationales by ensuring that only pertinent points in the input sequence are used to calculate the loss.

The model was trained for five epochs. The optimizer chosen for model parameter updates is Adam optimizer with weight decay (torch.optim.Adam()). This is because our model fits well for this optimizer, which offers effective updates while reducing overfitting with weight decay regularization.

3.3 Model Evaluation

The performance of our model in task 1, task 2, and tagging for each token would be assessed using these criteria.
In terms of performance, as in Chapter 2, we evaluate using three metrics: accuracy, the macro F1-score, and the AUROC score.

In terms of bias, as in the study done by Mathew et al., we evaluate using three metrics: GMB (Generalized Mean of Bias) AUC with Subgroup AUC, GMB AUC with BPSN (Background Positive, Subgroup Negative) AUC, and GMB AUC with BNSP (Background Negative, Subgroup Positive) AUC (Mathew et al., 2021).

These metrics assess the model's ability to handle toxic comments within specific communities. More specifically, Subgroup AUC evaluates differentiation between toxic and normal comments; BPSN AUC measures the model's tendency to misclassify normal community-related comments as toxic; BNSP AUC assesses misclassification of toxic community-related comments as normal; while GMB AUC combines these bias metrics into an overall measure.

We consider faithfulness and plausibility while evaluating the explainability of the model. IOU (Intersection-Over-Union) F1-Score and token F1-Score are used to assess plausibility; for soft token selection, the AUPRC (Area Under the Precision-Recall curve) is used. Token F1 assesses accuracy and recall at the token level, whereas IOU measures the overlap between the expected and ground truth reasoning. Comprehensiveness, which evaluates how model predictions vary when rationales are eliminated, and sufficiency, which assesses how extracted rationales affect predictions, are two metrics used to quantify faithfulness.

3.4 Results and Discussion

Table 2 displays the bert-base-uncased model performance results of task1 and task2. Table 3 indicates the overall bias score results of our model in task2. Our model’s total score of explainability is demonstrated in Table 4.

Task	Performance		
	Accuracy	Macro F1	AUROC
Task 1	0.687	0.669	0.833
Task 2	0.712	0.690	0.910

Table 2: Performance

Bias		
GMB AUC		
Subgroup AUC	BPSN AUC	BNSP AUC
0.804	0.811	0.792

Table 3: Bias

Explainability				
Plausibility			Faithfulness	
IOU F1	Token F1	AUPRC	Comp.	Suff.
0.068	0.231	0.368	-0.098	-0.026

Table 4: Explainability

According to those tables, in terms of task 1 performance, our model's accuracy, Macro F1 score, and AUROC are slightly lower than the best-performing model (BERT-HateXplain with 0.698, 0.687, and 0.851, respectively) in the study of Mathew et al. (Mathew et al., 2021).

On the other hand, our model does rather well in task 2. The results of accuracy, Macro F1 score, and AUROC are quite competitive. This demonstrates the potential of our approach for multi-classification challenges.

Furthermore, our model excels the best model results (BERT-HateXplain with 0.807, 0.745, and 0.763, respectively) in the work of Mathew et al., showing high in terms of bias measures (Mathew et al., 2021). All values are competitive compared to BERT-HateXplain model.

As for explainability, our model shows lower scores in explainability metrics compared to the top models, such as BiRNN-HateXplain [Attn] with an IOU F1 of 0.222 and Token F1 of 0.506. This implies room for improvement in how well our model's explanations align with ground truth.

Figure 2 displays Generalized Mean of Bias scores in each subgroup. Figure 3 compares BPSN AUC scores across different communities. Figure 4 compares BNSP AUC scores across different communities.

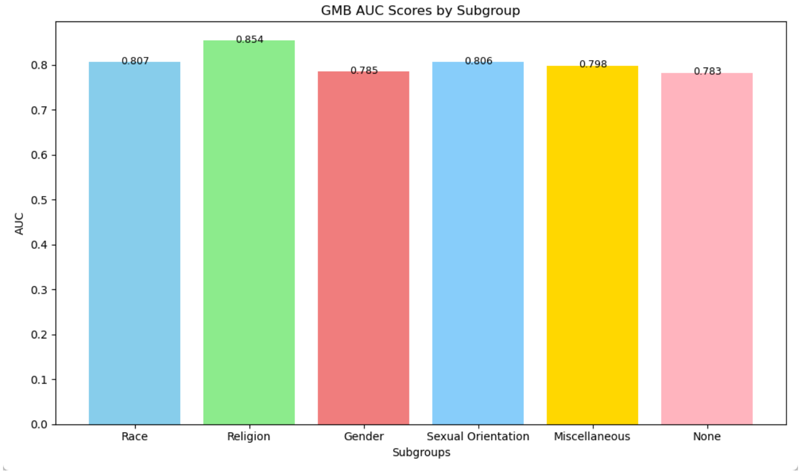


Figure 2: Scores GMB AUC with Subgroup AUC

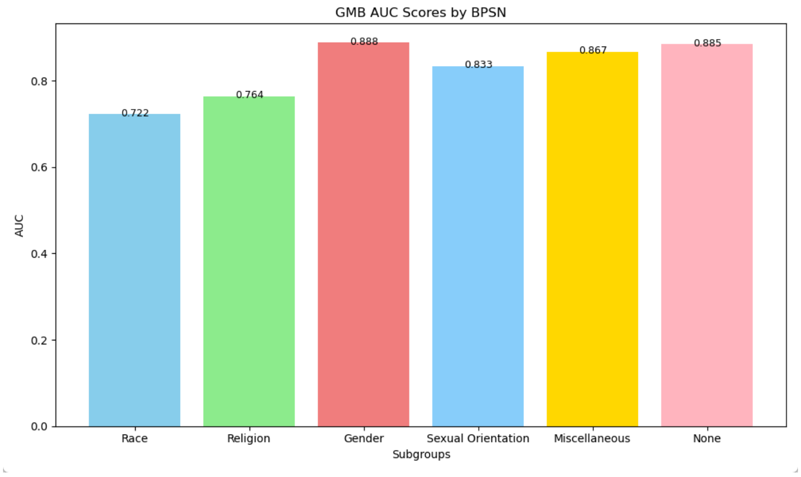


Figure 3: Scores GMB AUC with BPSN AUC

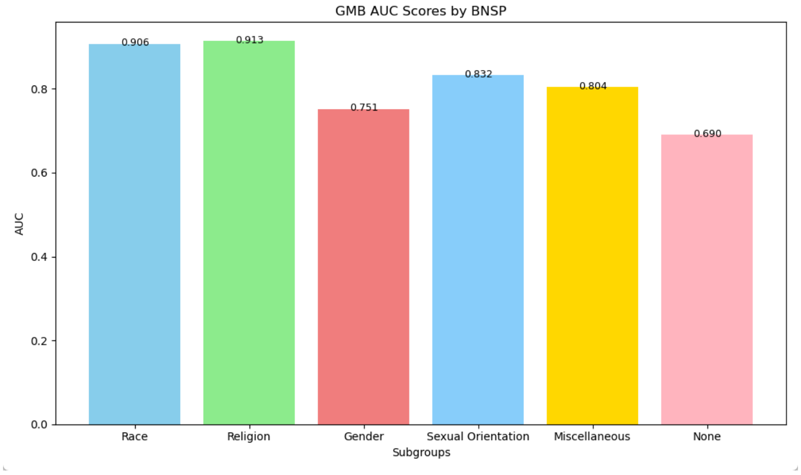
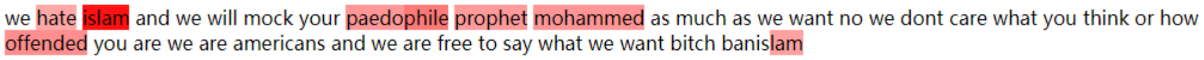


Figure 4: Scores GMB AUC with BNSP AUC

For instance, Task 1's output for the test text seen in Figure 5 is ["normal", "offensive", "hatespeech"] = [0.0039, 0.5704, 0.4257]. As shown in Figure 5, the colored ones represent the model's reasonales for considering this text to be toxic, and the color depth represents the model's assessment of the token's level of toxicity.



we hate islam and we will mock your paedophile prophet mohammed as much as we want no we dont care what you think or how offended you are we are americans and we are free to say what we want bitch banislam

Figure 5: One example from the dataset.The highlighted portion of the text represents the model’s rationale.

4. Conclusion

Our project used the HateXplain dataset to construct a model to detect and classify hate speech of text. For improved performance, we switched from the conventional machine learning models—Linear SVM and Logistic Regression—to a BERT-based model. The BERT model's demonstrated competitive results in multi-class classification and bias assessment, while its explainability scores were lower, suggesting space for improvement. Improving the model's explainability and effectiveness in identifying hate speech should be the main goal of future research. Results could be further enhanced by refining the model and growing the dataset.

References

Mathew, B., Saha, P., Yimam, S. M., Biemann, C., Goyal, P., & Mukherjee, A. (2021). HateXplain: A Benchmark Dataset for Explainable Hate Speech Detection. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 35, No. 17, pp. 14867-14875).