A Study of Hate Speech Detection and Classification

1 Introduction

As part of my one and a half year course at OMU University, I am devoting my research project to a subject very current today, which tackles the issues of online etiquette and user-friendly experience on social media, online video games and online communities. This study explores the application of advanced Natural Language Processing (NLP) techniques to detect and classify hate speech using the HateXplain dataset. The primary objective is to develop an effective hate speech detection model utilizing the BERT (Bidirectional Encoder Representations from Transformers) architecture. [1] The HateXplain dataset is a benchmark dataset for explainable hate speech detection, consisting of 20k samples categorized into ”hateful,” ”offensive,” and ”normal” labels. [2]

2 Related research

2.1 BERT: Bidirectional Encoder Representations from Transformers

BERT (Bidirectional Encoder Representations from Transformers) is a pre-trained transformer-based model designed for natural language understanding tasks. Developed by Google, BERT marked a significant advancement in NLP by employing bidirectional training of Transformer models, which allows it to understand the context of a word based on its surrounding words. [1]

2.1.1 BERT Architecture

BERT's architecture is based on the Transformer model introduced by Vaswani et al. (2017) [1]. The key components include:

- **Self-Attention Mechanism**: This allows BERT to weigh the importance of different words in a sentence when encoding a particular word.
- **Bidirectionality**: Unlike previous models that read text sequentially (left-to-right or right-to-left), BERT reads the entire sequence of words simultaneously, thereby capturing richer context.
- **Layers**: The model comes in two main sizes: BERT-Base (12 layers) and BERT-Large (24 layers), where each layer is a Transformer block.

2.1.2 Pre-training and Fine-tuning

BERT undergoes a two-phase training process:

1. **Pre-training**: BERT is pre-trained on a large corpus of text (e.g., Wikipedia) using two unsupervised tasks [1]:
   - **Masked Language Modeling (MLM)**: Randomly masks some of the tokens and predicts the masked words based on their context.
   - **Next Sentence Prediction (NSP)**: Predicts whether a given sentence B follows sentence A, helping the model understand relationships between sentences.
2. **Fine-tuning**: BERT is then fine-tuned on specific downstream tasks like question answering, sentiment analysis, and named entity recognition by adding a small number of task-specific parameters. [1]

2.2 HateXplain

2.2.1 HateXplain Dataset Overview

HateXplain is a benchmark dataset for explainable hate speech detection. It was introduced to address the need for explainability in hate speech detection models. [2] The dataset includes:

- **Text Samples**: Collected from Twitter[1] and Gab [2].
- **Labels**: Each text sample is labeled as ”Hateful,” ”Offensive,” or ”Normal.”
- **Rationales**: Annotators highlight specific parts of the text that justify their labeling, providing explanations for each label.
- **Target Groups**: The dataset also includes information about which target groups (e.g., race, religion) are being attacked.

2.2.2 BERT and HateXplain

BERT can be fine-tuned on the HateXplain dataset to leverage its powerful contextual understanding for hate speech detection. Studies have shown that BERT-based models, when trained on HateXplain, outperform traditional machine learning models in identifying and explaining hate speech. [1] [2]

[1] https://twitter.com/
[2] https://gab.com/

- **Performance Metrics**: Fine-tuning BERT on HateXplain has resulted in higher accuracy, precision, and recall compared to previous state-of-the-art models.
- **Explainability**: The use of BERT in conjunction with HateXplain's rationales has led to the development of models that not only detect hate speech but also provide human-understandable explanations for their decisions.

3 Proposed method

The proposed method to classify the dataset involves fine-tuning a pre-trained BERT model on the HateXplain dataset to create a hate speech detection system. The key components of the method include data preprocessing, model training, evaluation, and explainability.

4 Initial Dataset Analysis

4.1 Class Distribution

The dataset comprises 4,748 hateful posts, 4,384 offensive posts, and 6,251 normal posts. [2] This distribution presents a challenge in terms of class imbalance, which needs to be addressed to ensure that the model does not become biased towards the more frequent class.

4.2 Challenges

- Ambiguity in Language: Differentiating between offensive and non-offensive content can be subtle, relying heavily on context and nuances.
- Contextual Dependence: The meaning of posts can change based on external context, such as user history or cultural background, which is often not captured in the dataset.
- Inter-Annotator Variability: Different annotators may have varying interpretations of what constitutes hate speech or offensive content, leading to inconsistencies in labeling.
- Include references at the end.
- Evolving Language: Hate speech and offensive language evolve over time, introducing new terms and phrases regularly, which static datasets may not effectively capture.

5 Model Development

5.1 Training Configuration

Training arguments were specified to control various aspects of the training process: Training Arguments: num train epochs: 3 per device train batch size: 8 evaluation strategy: "steps" learning rate: Adjusted dynamically with warmup steps. The Trainer class from the Hugging Face transformers library was employed to handle the training and evaluation process, simplifying the management of training loops and evaluation metrics.

5.2 Steps

5.2.1 Data Preprocessing

- **Tokenization**: Convert text samples into tokens using the BERT tokenizer.
- **Label Mapping**: Combine "hateful" and "offensive" labels into a single category for simplification.

5.2.2 Model Architecture

- **BERT Model**: Utilize the `bert-base-uncased` model for its robust language understanding capabilities.

5.2.3 Training the Model

- **Fine-Tuning**: Fine-tune the BERT model on the HateXplain dataset, including rationales for better context understanding.

5.2.4 Evaluation

- **Metrics**: Evaluate the model using accuracy, precision, recall, and F1-score.
- **Confusion Matrix**: Analyze the model's performance across different categories.

5.2.5 Explainability

- **Rationales**: Use the rationales provided in the HateXplain dataset to highlight which parts of the text the model uses to make its predictions.

5.2.6 Detailed Explanation

Data Preprocessing The text data from the HateXplain dataset is tokenized using BERT's tokenizer. This converts the text into a format suitable for input into the BERT model.

Model Architecture The `bert-base-uncased` model is used for its pre-trained language understanding capabilities. This model will be fine-tuned on the HateXplain dataset to adapt it to the specific task of hate speech detection.

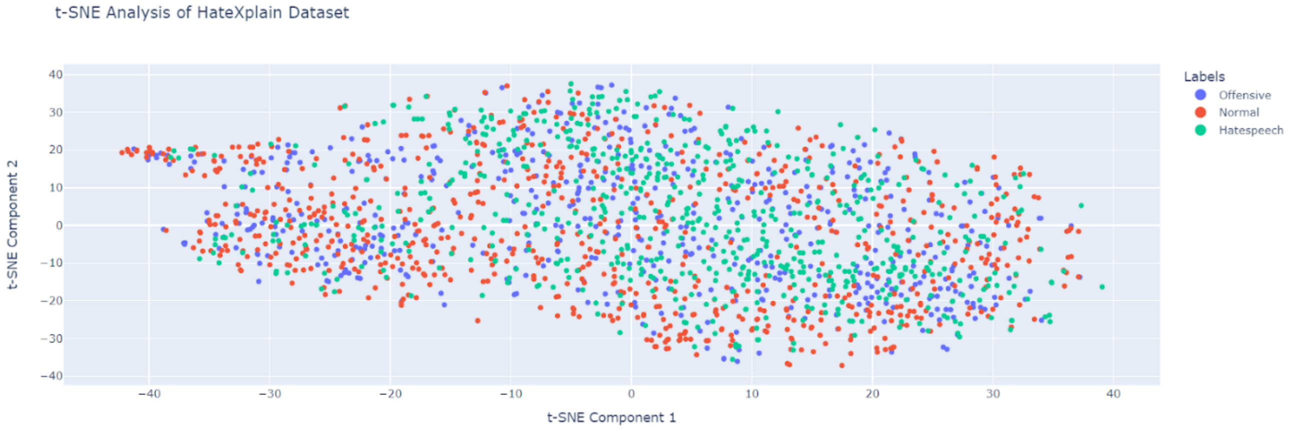

Figure 1: t-SNE of the BERT model fine tuned to the HateXplain dataset

Training the Model The BERT model is fine-tuned on the HateXplain dataset, including the use of rationales to help the model understand which parts of the text are important for the classification task.

Evaluation The model's performance is evaluated using standard metrics such as accuracy, precision, recall, and F1-score. These metrics help in understanding how well the model is performing in identifying hate speech. A confusion matrix is used to visualize the performance of the model across different categories, showing true positives, false positives, true negatives, and false negatives.

Explainability The rationales provided in the HateXplain dataset are used to highlight the parts of the text that the model focuses on for making its predictions. This enhances the explainability of the model, making it easier to understand why certain texts are classified as hate speech.

6 Model Training and Evaluation

This section summarizes the presentation and discuss the challenges that need to be addressed in the future.

6.1 Training Process

The BERT model was fine-tuned on the HateXplain dataset using the specified training arguments. The Trainer class managed the training loop, including forward and backward passes, gradient updates, and logging, ensuring an efficient and streamlined training process.

6.2 Evaluation

The model's performance was evaluated using metrics such as accuracy, precision, recall, and F1-score. The confusion matrix was employed to illustrate the model's performance across different classes, showing true positives, false positives, and false negatives for each class.

7 Results and Observations

7.1 t-SNE analysis

BERT embeddings were used to capture the semantic representations of the text data. These embeddings are high-dimensional and contain rich contextual information about the text. t-SNE was applied to reduce the high-dimensional BERT embeddings to a two-dimensional space, enabling visualization of the data points. The resulting in figure 1. a 2D plot displays the data points color-coded by their labels: Red: Normal; Green: Hateful; Blue: Offensive

Observations: The visualization shows some degree of clustering within each category, indicating that the BERT embeddings capture distinct features for each class. But there is still a huge overlap between all categories, suggesting that these classes share common features that make them harder to distinguish. More significant overlap is observed between the "Offensive" and "Normal" categories. This overlap could be due to the subtle differences in language use and context that make it challenging to distinguish between offensive and benign content.

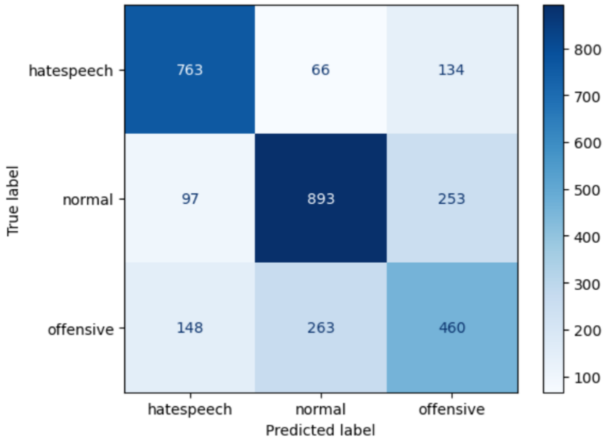

Figure 2: Confusion matrix of the BERT model fine tuned to the HateXplain dataset

7.2 Confusion Matrix Analysis

- Hate speech: True Positives (TP): 763, False Negatives (FN): 245, False Positives (FP): 200.
- Normal: True Positives (TP): 893, False Negatives (FN): 329, False Positives (FP): 350.
- Offensive: True Positives (TP): 460, False Negatives (FN): 387, False Positives (FP): 411.

7.3 Example Misclassifications

- True Label: Hatespeech, Predicted Label: Normal: ”i do not moozlim shitty hands on my food fuck tyson and starbucks”
- True Label: Hatespeech, Predicted Label: Offensive: “Warning women men boy girls and toddlers all are fair game for rapes and other forms violence from these obama friendly terrorist immigrants”

8 Challenges and Improvements

8.1 Misclassification Issues

The model exhibited significant misclassifications, especially between the ”normal” and ”offensive” categories. Subtle differences in language and context dependency contributed to these misclassifications.

8.2 Potential Improvements

- Data Augmentation: Increasing the diversity of the training data to better capture variations in language and context.
- Class Weight Adjustment: Balancing the impact of underrepresented classes to prevent model bias.
- Hyperparameter Tuning: Optimizing model parameters for better performance.
- Advanced Models: Exploring more sophisticated models like RoBERTa or BERT-large for potential performance improvements.
- Ensemble Methods: Combining multiple models to enhance accuracy and robustness.

9 Future Work

9.1 Cross-Domain Transfer Learning

Adapting the model to different datasets and domains to improve generalizability and robustness.

9.2 Explainability

Improving the model’s ability to provide transparent and understandable explanations for its predictions, enhancing trust and usability in real-world applications.

10 Conclusion

This study demonstrates the potential of using BERT for hate speech detection. While the model shows promising results, there is room for improvement, especially in handling subtle and context-dependent language. Future work will focus on enhancing model accuracy, interpretability, and adaptability to different languages and domains. The ultimate goal is to contribute to the development of more effective and transparent hate speech detection models, promoting safer online communities.

References

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Long and Short Papers*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423

[2] B. Mathew, P. Saha, S. M. Yimam, C. Biemann, P. Goyal, and A. Mukherjee, “Hatexplain: A benchmark dataset for explainable hate speech detection,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2020, pp. 3451–3463.