Hate Speech Implication Generation

{lakshay21060, krishna21058, mayank21065, arnav21019} @iiitd.ac.in

April 25, 2024

Abstract

In this paper, we present a detailed methodology for addressing the pervasive issue of hate speech on social media platforms. We recognize the dual nature of social media, serving as a vital communication tool while also harboring harmful behaviors such as hate speech. Our approach encompasses a range of techniques, including binary classification(rf, XGBoost, Bert and SVM) and generative frameworks, to accurately identify and combat hate speech implications in online content. By harnessing the power of machine learning algorithms, explainability techniques, and sophisticated NLP models like GPT-2 and T5, we strive to provide a comprehensive solution to the challenges posed by hate speech. Through meticulous preprocessing, classification, and generative modeling, our methodology aims to contribute to fostering a safer and more inclusive online environment for all users.The dataset provides structured annotations of over 150,000 social media posts, spanning various demographic groups and social contexts. Each post is labeled with target stereotypes, target categories, and other relevant attributes.

1 Introduction

Social media has gradually become an irreplaceable facet of our lives, easily facilitating conversations, connections, and commerce. But while it provides people with platforms to voice their opinions, it's also opened the doors to the proliferation of what's conventionally classified as 'anti-social behavior', including harassment, trolling, bullying, and hate speech. Social media platforms, digital social clubs, and online communities often serve as breeding grounds for the dissemination of discriminatory language, dog whistles, and dated attitudes toward oppressed demographic groups. Many now view the prevalence of hate speech on social media platforms as a major problem, given how quickly it spreads and encourages harm to both individuals and society writ large. Recognizing the gravity of this issue, researchers and policymakers have increasingly turned their attention toward developing tools and methods to identify and mitigate the spread of such harmful content.

In the aftermath of the 2016 U.S. Elections, research on the detection of hate speech has garnered significant attention within both humanities and sciences, driven by the need to address the impacts of hate speech on social and political discourse. Scholars from linguistics and computer science have undertaken studies to understand the mechanisms and implications of hate speech in social media, stemming from its potential to incite violence and perpetuate discrimination and aggression, especially in code-mixed languages, where it's especially harder to detect.

Advancements in natural language processing, through the creation and utilization of transformer-based models like BERT and GPT-2, have contributed to enhanced hate speech detection capabilities. Sahoo, Gupta, and Bhattacharya (2022) utilized the aforementioned models to identify hate speech across multiple bias classes, achieving state-of-the-art performance. On the other hand, Badjatiya et al. (2017) and Malik, Pang, and Hengel (2022) employed deep learning architectures and embeddings, with the latter showcasing a higher detection efficacy when using both approaches in concord.

Though these efforts have largely (and somewhat successfully) focused on identifying hate speech itself, there is a growing need to detect the target groups and, by extension, stereotypes perpetuated within such content, which we seek to address through this paper.

2 Related Work

Hate Speech Identification: 'Detecting Unintended Social Bias in Toxic Language Datasets' by Sahoo, Gupta, and Bhattacharyya utillised BERT and GPT-2 for bias detection and generation of targets and implications from toxic language datasets. It showcased the prediction of categorical variables and implications as a generation task, utilizing the GPT-2 model. Sahoo et al. further employed BLEU and ROUGE-L to assess the generation performance. Their approach was successfully able to mitigate model bias and lexical overfitting through counter-narrative data augmentation, leveraging counter-narrative datasets, including CONAN and Multi-target CONAN, to provide positive instances mentioning terms related to race, religion, gender, etc. These counter-narratives' introduction into their dataset reduced the model's tendency to stick to identity-based terms and associate them with biased labels and led to improvements in AUC-based bias metrics, demonstrating the effectiveness of their strategy.

Stereotype Identification: 'StereoHate: Towards identifying Stereotypical Bias and Target group in Hate Speech Detection' focused on hate speech detection in low-resource languages, emphasising on the explainability aspect through text generation. It introduced the HHES corpus, which annotates Hindi hate speech posts with stereotypical biases and target group categories, developing the CGenEx framework, which reframes hate speech detection as a text-to-text generation task. This enabled the model to simultaneously generate explanations for hate speech and classify it, offering insights into the internal biases present in the text. CGenEx addresses the dual challenges of hate speech detection and explanation by leveraging a commonsense-aware generative framework. The authors further proposed two variants of CGenEx:- CGenEx-con and CGenEx-fuse which incorporated pre-trained sequence-to-sequence models, such as mBART and mT5, to generate explanations for hate speech while classifying it into target categories, demonstrating the effectiveness of CGenEx in generating explanations for hate speech in Hindi.

'CO-STAR: Conceptualisation of Stereotypes for Analysis and Reasoning' by Kwon and Gopalan proposed the CO-STAR framework that structures stereotypes in a [TARGET GROUP] [RELATION] [IMPLIED STATEMENT] format, where the [RELATION] component is chosen from a set of 8 linking verbs inspired by ConceptNet-5 relations while incorporating "stereotype conceptualizations" to provide contextual knowledge. However, we haven't incorporated it in our project since our project solely considers the [TARGET GROUP] class for stereotype categorization. Nonetheless, the paper's approach improved the quality and relevance of generated stereotypes through ablation studies compared to previous models' performance in its absence on the dataset introduced in 'Social Bias Frames: Reasoning about Social and Power Implications of Language' by Sap et al.

Implication Deduction: 'Latent Hatred: A Benchmark for Understanding Implicit Hate Speech' by ElSherief, Ziems et al. introduced a taxonomy of implicit hate speech and developed a benchmark corpus with fine-tuned labels for each message and its implied meaning. The paper essentially explores natural language descriptions' generation for implied stereotypes from a dataset of implicit hate speech tweets, using GPT and GPT-2, evaluating the generated implications using BLEU and ROUGE-L, which just about match our evaluation approach with the exception of BERTScore. Notably, the authors identified linguistic challenges in detecting and explaining implicit hate speech, like coded symbols, discourse relations, entity framing, and irony.

3 Methodology

In our methodology, we have explored various approaches to determine the target stereotype for a corresponding post. Initially, we ran baseline models such as BART and GPT2 for generation, and the evaluation metrics for these models are presented in **Section 5 Table 2**. Subsequently, we shifted our focus to binary classification, wherein we aimed to classify whether a post targets a stereotype or not. Along with the posts, we utilized a rich set of features **(refer to Section 4.1 Dataset for more details)** and employed machine learning algorithms including SVM, BERT, RF, and XGBoost. After these machine learning classifiers, we employed explainability techniques like LIME to gain insights into which features were the most important. Additionally, we identified the lime features with the highest explainability scores and concatenated them with the posts, separated by SEP tokens. We then created BERT embeddings and further employed GPT decoding to determine the target stereotype. Furthermore, we utilized the T5 model by sending posts with encoded features(11 features including

LIME features) to generate predictions. For further details about the dataset(refer to Section 4.1 Dataset)

In the subsequent sections, we will delve into the details of our two approach, discussing our methodology and findings in detail.

3.1 Model 1:Binary Classifer + Bert Embeddings + GPT decoder

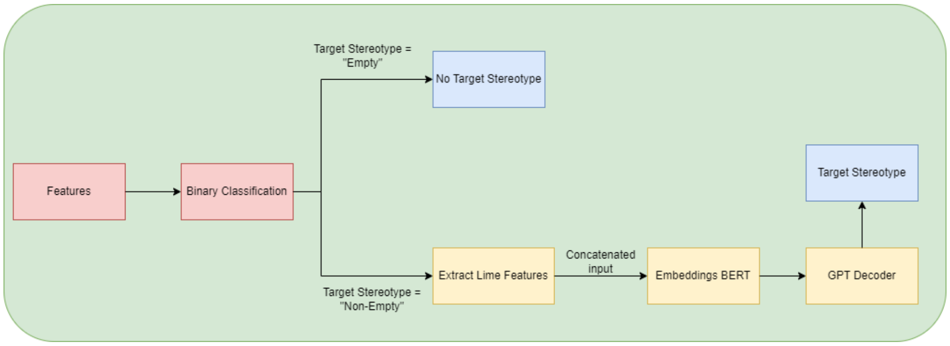

Figure 1: Overall Pipeline Model-1

3.1.1 Pre Processing

The preprocessing of posts is a crucial step to ensure that the text data is clean and properly formatted before training the GPT-2 model. This process involves several tasks aimed at enhancing the quality of the input data. Initially, the posts undergo HTML entity unescaping to ensure that any encoded HTML entities are decoded properly, allowing for accurate text representation.Subsequently, a series of regular expression patterns, as defined in the regex patterns dictionary, are applied to the posts. These patterns target various textual elements such as retweets, URLs, punctuation marks, and special punctuation characters. By replacing these patterns with spaces or other delimiters, the text is normalized and stripped of any irrelevant or distracting elements.

3.1.2 Binary Classification

In Section 4, we showcased in Figure 3 that a significant portion of the dataset's posts does not target any stereotype. Therefore, accurately measuring the target stereotype required an initial classification step to discern whether a given post does or does not target a stereotype. To achieve this, we employed a range of machine learning classifiers including Random Forest (RF), Support Vector Machine (SVM), BERT, and XGBoost. Additionally, we utilized explainability tools such as Local Interpretable Model-agnostic Explanations (LIME) to unravel the significant features contributing to the classification decisions. LIME offers insights into the internal mechanisms of complex models by elucidating individual predictions in terms of feature importance. By leveraging LIME, we were able to uncover latent features within the data. Our classification endeavors culminated in remarkable success, particularly with Random Forest achieving an exceptional F1 score of 0.99**(refer to table 1 Section 5)**. This outstanding performance underscores RF's capability to accurately discern whether a sample post targets a stereotype, thus highlighting its effectiveness in classifying posts based on their association with stereotypes.

3.1.3 Task Formulation

During training, the generation model takes a sequence of tokens as input:

$$x = \{[STR], t_1, t_2, ..., t_n, [SEP], t[G_1], t[G_2], ..., [SEP], t[S_1], t[S_2], ..., [END]\}$$

with start token $[STR]$, post tokens $t_1 : t_n$, target sterotype $t[Gi]$, and Lime Features $t[Li]$, and minimizes the cross-entropy loss

$$-\sum_l \log P(\tilde{t}_l | t_l).$$

Further these sequence of tokens are sent to compute BERT embeddings to understand contextual semantics of the input sequence of tokens.

3.1.4 Generative Framework

In the generative framework section, we delve into the intricacies of the training process, where the model's parameters are optimized through iterative refinement using the Adam optimizer. This optimizer dynamically adjusts the learning rate for each parameter, facilitating efficient convergence towards minimizing the cross-entropy loss, a pivotal component guiding the model's predictions. In the **Results section 5**, refer to training loss vs. epoch (Figure 7) .Throughout training, careful consideration is given to various parameters such as batch size, learning rate, and the number of epochs, ensuring an optimal balance between model convergence and generalization. The GPT model, leveraging its autoregressive nature, sequentially generates target stereotypes by predicting the next token in the sequence based on the preceding context. This iterative process allows the model to dynamically adjust its predictions, gradually refining its output to align with the target stereotype. By meticulously fine-tuning these parameters and optimizing the training process, we aim to enhance the model's performance in generating coherent and contextually relevant target stereotypes. Ultimately, this training methodology empowers the GPT model to effectively mitigate the impact of hate speech by generating nuanced and accurate outputs, contributing to fostering a more inclusive and respectful online discourse.

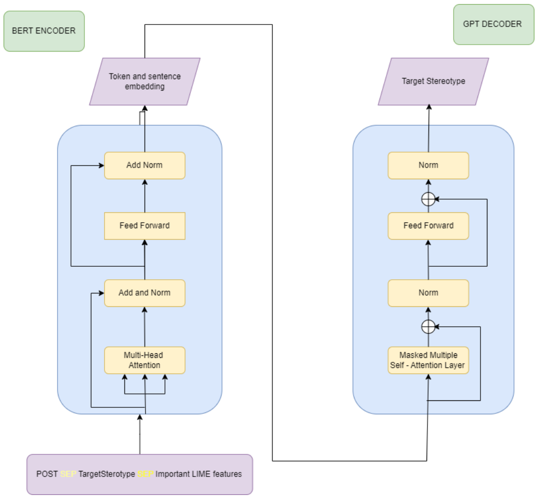
BERT ENCODER
GPT DECODER
Token and sentence embedding
Target Stereotype
Add Norm
Feed Forward
Add and Norm
Multi-Head Attention
Norm
Feed Forward
Norm
Masked Multiple Self - Attention Layer
POST SEP TargetSterotype SEP Important LIME features

Figure 2: Model 1 Architecture

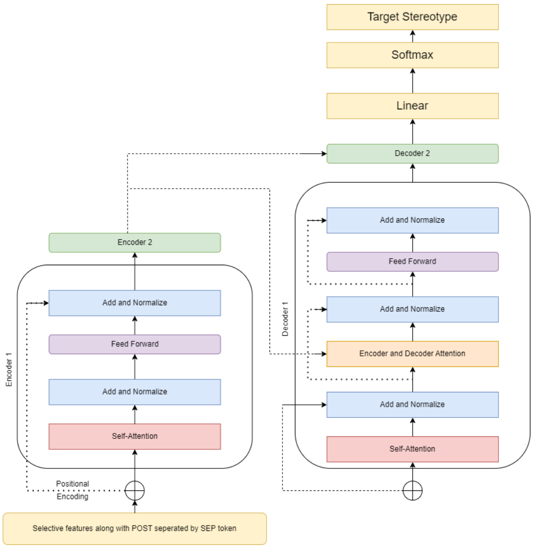

Figure 3: Model 2 Architecture

3.2 Model2 : Important Features + T5 small Model

3.2.1 Pre Processing

Similar to **section 3.1.1** , the preprocess of the posts undergo HTML entity unescaping to ensure that any encoded HTML entities are decoded properly, allowing for accurate text representation.Subsequently, a series of regular expression patterns, as defined in the regex patterns dictionary, are applied to the posts.

3.2.2 Task Formulation

During training, the generation model takes a sequence of tokens as input:

$$x = \{ \quad [POST] \quad t_1, \quad t_2, \quad ..... \quad t_n \quad [POST] \quad L_1 : V_1 \quad L_2 : V_2 \quad ..... \quad L_m : V_m \quad \}$$

where $[POST]$ encapsulates the post, Lime Features L_i are labels or columns under the dataset and V_i are their respective values for the label L_i of the sample and minimizes the cross-entropy loss

$$-\sum_l \log P(\tilde{t}_l | t_l).$$

3.2.3 Generative Framework

Our approach with the T5 model involves leveraging its text-to-text framework to generate emphasized conclusions by incorporating both post content and associated features. We preprocess each sample by encoding the post along with relevant features into a single input sequence. This combined input is then tokenized and fed into the T5 model for conditional generation. During training, we utilize the Adam optimizer with a learning rate of 5e-5 to minimize the loss function, optimizing the model's parameters to improve performance.

It entails on constructing a custom dataset class to handle the input sequences and their corresponding target stereotypes. We split the dataset into training and validation sets, with appropriate batch sizes for efficient processing. Training proceeds over multiple epochs, with each epoch iterating through the training and validation datasets to compute the loss. We monitor both training and validation losses to track the model's performance over epochs, aiming to minimize loss values to enhance the model's accuracy in generating emphasized conclusions.In the Results section, refer to training loss vs. epoch (Figure 8).

4 Dataset and Experimental Setup

For the target stereotype generation task, we utilized a dataset comprising social media posts annotated with target stereotypes. This dataset provides structured annotations of over 150,000 social media posts, spanning various demographic groups and social contexts. Each post is labeled with target stereotypes, target categories, and other relevant attributes.We have 15 features (including POST) in the dataset.

4.1 Dataset

The dataset comprises posts potentially containing offensive language, annotated with details on the nature of the offense, annotator demographics, and post origins. Fields include indicators of intent to offend, presence of sexual content, and overall offensiveness. Annotations cover annotator demographics such as age, gender, politics, and race. Posts are categorized based on target demographics and implied stereotypes. The dataset is split into training, validation, and test sets to avoid duplication. Additional information includes annotator minority status, sexual content references, speaker minority status, worker and HIT IDs, annotator politics, age, post text, target demographic details, stereotype implications, and post source.

The dataset is divided into three subsets: training, validation, and test sets. These subsets ensure that no post appears in multiple splits. The training set comprises 112,900 instances, while the validation and test sets comprise 16,738 and 17,501 instances, respectively. Each split includes the post along with its three sets of annotations. This division facilitates model training, validation, and evaluation processes while maintaining data integrity and preventing data leakage.

4.1.1 Analysis

The task is to generate the "targetStereotype" column and all columns other columns except for targetMinority, targetCategory and targetStereotype can be used. So, for the analysis, we have visualized the unique values for each column (in Fig. 3). Since some columns had some null (empty) values, we tried to find out the number of values that are not empty (in Fig. 4) for each model to get more information on which column to use and which column to skip.

We applied LIME (Local Interpretable Model-Agnostic Explanations) to find features are the most

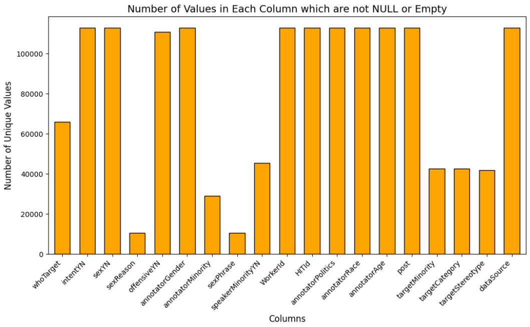

Figure 4: Number of Values in Each Column which are not NULL or Empty

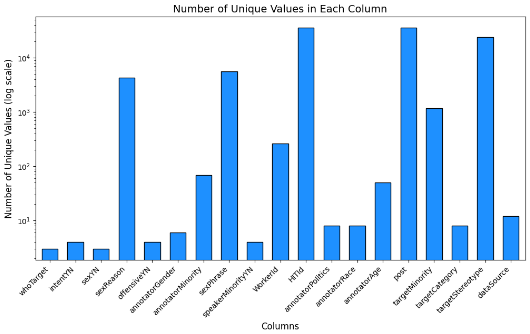

Figure 5: Number of Unique Values in Each Column

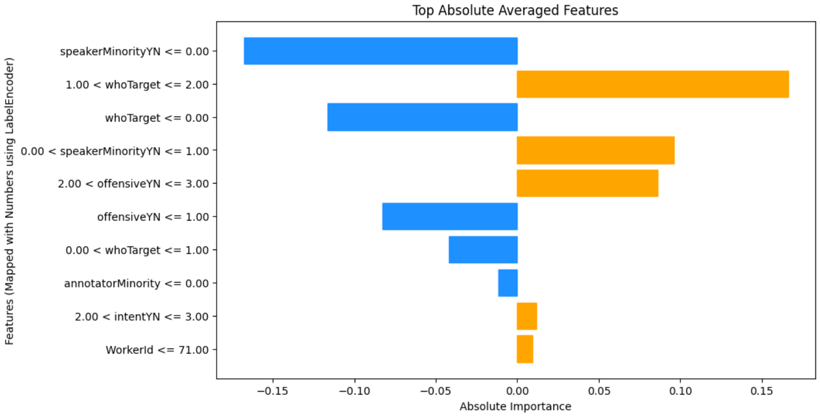

Figure 6: Features with Top Lime Scores

4.2 Experimental Setup

In our experimental setup, we focused on evaluating the performance of generated stereotypes using three evaluation metrics: BLEU, ROUGE-L, and BERTScore. we have setup our program on NVIDIA-SMI 510.54 with 128 cores and 4 GPU cores with CUDA version 11.6

(i) BLEU(n=2): One of the earliest metrics used to measure the similarity between two phrases, BLEU computes the geometric mean of n-gram precision scores with a brevity penalty for short sentences.

(ii) ROUGE-L(n=2): Originally designed for assessing summarization systems, ROUGE-L compares overlapping n-grams, word sequences, and word pairs between two phrases. We employed the ROUGE-L version, which measures the longest common subsequences.

(iii) BERTScore: This metric relies on pre-trained BERT contextual embeddings to compute similarity between two phrases. BERTScore uses a weighted aggregate of cosine similarities between tokens to determine the overall similarity.

By employing these evaluation metrics, we aimed to assess the quality and relevance of the generated stereotypes compared to the ground truth annotations in the dataset. This comprehensive evaluation framework allowed us to measure the effectiveness of our approach in capturing the intended stereotypes and target categories present in social media posts.

5 Results and Observations

5.1 Results

Results with required metrics Bert scores, BLEU, Rouge L are shown in table 1 and 2. Also the plots for Loss vs Epochs for M1 and M2 are given in figure 7 and figure 8.

Random Forest	**0.99**
BERT	0.41
XGBoost	0.81
SVM	0.57

Table 1: Binary Classification

Evaluation metric vs Model	Bart	GPT	Bert + GPT (M1)	**T5-small (M2)**
Bert score	0.62713962912	0.7464352786540985	0.62775981	**0.88446259337563**
BLEU	0.193467352182	0.183935641029783	0.16291668	**0.3314093252785033**
Rouge L	0.10253989490	0.092176580367912	0.08418227443	**0.14467106318913975**

Table 2: Evaluation metric vs Model table

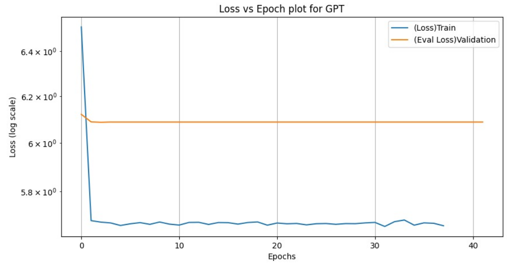

Figure 7: Loss vs Epoch plot of M1

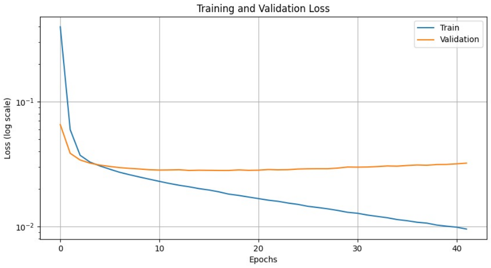

Figure 8: Loss vs Epoch plot of M2

5.2 Observations

1. We observe that BART and t5 small performed better than others. This is because BART and T5 are generally optimized for text-to-text generation tasks, with BART leveraging bidirectional and autoregressive training to capture context. At the same time, T5's design specifically targets text-to-text tasks, leading to more effective performance compared to models like GPT. Both have been fine-tuned on large datasets, boosting their text generation capabilities.

2. t5-small performed better than all other models and it achieved a Here, T5 outperforms BART possibly because its design focuses solely on text-to-text transfer and the task demands the same. T5's architecture is optimized for various text-generation tasks, allowing it to excel in diverse text-generation scenarios. Additionally, T5's training objective encourages it to learn to understand and generate text in a versatile manner, potentially leading to better performance in text-to-text generation tasks than BART, which has a more diverse set of pre-training objectives.

6 Conclusion

Hate speech, with its potential to incite violence and spread animosity, poses a significant threat to societal harmony and individual well-being. Identifying the target stereotype group of hate speech is crucial in mitigating its harmful effects. In this context, AI and NLP approaches play a pivotal role in detecting and analyzing hate speech implications. These technologies offer scalable solutions for identifying patterns and understanding the context in which hate speech occurs.

The study concludes that BART and T5 models, especially T5 small, outperformed others due to their fine-tuning for text-to-text tasks and superior architecture. Both models have extensive training on large datasets, enhancing their proficiency. Methodologically, text data preprocessing ensured cleanliness and proper formatting. Machine learning classifiers like Random Forest and Support Vector Machine successfully identified stereotype-targeted posts. Training focused on optimizing parameters for efficient convergence and generalization. T5's training involved encoding posts and relevant features into input sequences and utilizing the Adam optimizer for loss minimization, ensuring effective text generation capabilities.

7 Future Work

While our methodology demonstrates promising results, there are avenues for further improvement and exploration. Enhancing our model's accuracy and robustness could involve refining the feature engineering process and exploring alternative architectures for hate speech detection. Additionally, addressing inherent limitations, such as potential biases in training data and the risk of generating incorrect responses, remains crucial. Future research could focus on diversifying training datasets and implementing more sophisticated evaluation strategies to mitigate these challenges. Exploring various GPT-2 variants to find the most suitable for hate speech detection and generation can optimize model performance, aligning with our work's objectives.

Furthermore, understanding the nuances of hate speech across different cultural and linguistic contexts presents an exciting opportunity for future investigation. AI and NLP technologies can contribute significantly to fostering inclusive and respectful online discourse by adopting a more nuanced approach to hate speech detection and generation.

References

Maity, K., Ghosh, N., Jain, R., Saha, S., Bhattacharyya, P. *StereoHate: Towards identifying Stereotypical Bias and Target group in Hate Speech Detection.* Natural Language Engineering. 2023. DOI: 10.1017/S1351324922000555

Sahoo, N., Gupta, H., & Bhattacharyya, P. *Detecting unintended social bias in toxic language datasets.* 2022. arXiv preprint arXiv:2210.11762.

Badjatiya, P., Gupta, S., Gupta, M., & Varma, V. *Deep learning for hate speech detection in tweets.* 2017. Proceedings of the 26th International Conference on World Wide Web Companion (pp. 759-760). DOI: 10.1145/3041021.3054223

Kwon, T., Gopalan, A. *CO-STAR: Conceptualisation of Stereotypes for Analysis and Reasoning.* 2021. arXiv preprint arXiv:2112.00819.

Maity, K., Ghosh, N., Jain, R., Saha, S., Bhattacharyya, P. *StereoHate: Towards identifying Stereotypical Bias and Target group in Hate Speech Detection.* 2019. Natural Language Engineering 1. DOI: 10.1145/3308558.3313504

Malik, J. S., Pang, G., Hengel, A. V. D. *Deep learning for hate speech detection: a comparative study.* 2022. arXiv preprint arXiv:2202.09517.

ElSherief, M., Ziems, C., Muchlinski, D., Anupindi, V., Seybolt, J., De Choudhury, M., Yang, D. .Latent hatred: A benchmark for understanding implicit hate speech. 2021. arXiv preprint arXiv:2109.05322.

Mullah, N. S., Zainon, W. M. N. W. *Advances in machine learning algorithms for hate speech detection in social media: a review.* 2021. IEEE Access, 9, 88364-88376. DOI: 10.11009/ACCESS.2021.3089515

Abro, S., Shaikh, S., Khand, Z. H., Zafar, A., Khan, S., Mujtaba, G. *Automatic hate speech detection using machine learning: A comparative study.* 2020. International Journal of Advanced Computer Science and Applications, 11(8). DOI: 10.14569/IJACSA.2020.0110861