Explainable Identification of Hate Speech towards Islam using Graph Neural Networks

Abstract

Islamophobic language on online platforms fosters intolerance, making detection and elimination crucial for promoting harmony. Traditional hate speech detection models rely on NLP techniques like tokenization, part-of-speech tagging, and encoder-decoder models. However, Graph Neural Networks (GNNs), with their ability to utilize relationships between data points, offer more effective detection and greater explainability. In this work, speeches are represented as nodes and connect them with edges based on their context and similarity to develop the graph. A novel paradigm using GNNs to identify and explain hate speech towards Islam is introduced. The model leverages GNNs to understand the context and patterns of hate speech by connecting texts via pretrained NLP-generated word embeddings, achieving state-of-the-art performance and enhancing detection accuracy while providing valuable explanations. This highlights the potential of GNNs in combating online hate speech and fostering a safer, more inclusive online environment.

1 Introduction

Detecting and eliminating hate speech on social media platforms is of utmost importance for the promotion of harmony and tranquility in society. The escalating presence of hate speech specifically targeting Islam or Muslim communities on online discussion platforms is a growing concern. This form of hate speech not only fosters an environment of intolerance and hostility but can also have severe psychological impacts on individuals and communities, leading to real-world violence and discrimination.

To address this issue, researchers have increasingly turned to advanced technologies; using text-processing approaches in AI. Natural Language Processing (NLP) techniques are frequently employed for hate speech detection, with some offering severity assessment of hate speech. These methods utilize sophisticated algorithms to analyse vast amounts of textual data, identifying patterns and features indicative of hate speech. For instance, deep learning models, like recurrent neural networks (RNNs), can learn complex representations of text data, enabling them to detect subtle and context-dependent instances of hate speech. Modern NLP techniques, on the other hand, can enhance these models by providing richer linguistic insights. Tokenization, part-of-speech tagging, and named entity recognition are just a few NLP techniques that help in breaking down and understanding the text's structure and meaning. Moreover, the integration of latest NLP model and transformers, like BERT and GPT, has significantly improved the ability of models to understand context, sarcasm, and implicit hate speech, which are often challenging to detect. Another interesting approach is to use human-centric perspectives of AI using some benchmark dataset.

Researchers have tried to employ GNNs in hate speech classification, but still needs more focus on this area. Despite their potential, GNNs have not been actively employed for the purpose of interpretable identification of hate speech, particularly in Islamic contexts. Islamophobic content often exhibits close word choices and hate speakers from the same community, which GNNs can leverage to reveal and explain patterns, alongside impressive classification scores.

A novel approach employing graph neural networks for the identification and explication of hate speech directed at Islam (XG-HSI) is introduced. The dataset is pre-processed to focus on Islamic contexts, utilize pretrained NLP models for word embeddings, establish connections between texts, and employ a series of graph encoders for hate speech target identification, which achieves state-of-the-art performance.

2 Background

Graph Neural Networks (GNNs) are powerful neural networks designed for processing non-Euclidean data organized in complex, interconnected graphs. Using their ability to utilize relations between different data points, GNNs have shown tremendous promise in text classification and detection tasks. GNNs have the ability to enhance hate speech detection on social media by modeling complex relationships between users and content, capturing contextual information from interactions. They propagate information across the network, identifying coordinated and evolving hate speech patterns. We also present a case study in Section 5 to illustrate how incorporating related information enhances the process.

A general bag of words-based approach to create graphs, without LLMs is adopted. By integrating with pretrained NLP models, GNNs leverage contextual word embeddings to better understand the subtleties of hate speech. This combined approach improves the accuracy, context-awareness, and adaptability of detection systems, making them more effective in identifying hate speech directed at Islam and potentially generalizing to other targeted groups.

3 Methodology

3.1 Notations

Let a graph G = (V, E, X), where V represents nodes, E denotes edges. We also define N and M as the numbers of nodes and edges, respectively. Each node v is associated with a feature xi Ž208 RF , and the node feature matrix for the entire graph is denoted as X Ž208 RN Ŏ0d7F , where F represents the feature vector length. In our approach, each content denotes a node, contextual similarity between two nodes is denoted by an edge and word embeddings are node features of the graph. The task involves a node classification task to detect hate speech and Islamophobic content.

3.2 Data Pre-Processing

Initially, the dataset was filtered to focus on hate speech targeting Islam. Next, pretrained NLP models is applied to the text to obtain word embeddings X as node features for all nodes V. Edges E are determined using cosine similarity between embeddings with a threshold of 0.725. Subsequently, GNN is applied for the classification task.

3.3 Graph Encoder

After data pre-processing, every data point x Ž282 X undergoes a series of transformations to get output p. First, it is processed by a linear layer producing x1 (Equation 1).

$$x1 = Wx + b \tag{1}$$

Subsequently, x1 is passed into two initial graph encoders to aggregate neighborhood information, feature extraction, and yield x2, x3 utilizing G and concatenated to x23 (Equation 2,3, 4). Here in Equation 2, we aggregate features from a node's local neighborhood, to learn different characteristics. In Equation 3 and 4, we use a semi-supervised learning on graph-structured data, employing an efficient variant of convolutional neural networks that operate directly on graphs.

$$x2 = W1x1 + W2 \cdot mean_{j \in N(i)} x1 \tag{2}$$

$$x3 = W1x1 + \hat{A}x1 \tag{3}$$

$$x23 = concat(x2, x3) \tag{4}$$

Here, N is the set of neighbouring nodes. Following this, x23 is passed through another graph layer employing attention-based feature extraction, utilizing masked self-attentional layers to implicitly assign different weights to nodes in a neighbourhood, producing x4 (Equation 5 and 6).

$$x4 = \alpha_{i,i}\Theta x23_i + \sum_{j \in N(i)} \alpha_{i,j}\Theta x23_j \tag{5}$$

$$\alpha_{i,j} = \frac{exp(LeakyReLU(a^T[\Theta x23_i || \Theta x23_j]))}{\sum_{k \in N(i)} exp(LeakyReLU(a^T[\Theta x23_i || \Theta x23_k]))} \tag{6}$$

Here, ŏ3b8 refers to trainable model weights. ŏ3b1 is the attention value, calculated by the equation mentioned.

Finally, x4 is passed through a final linear layer to obtain logits pl, which are then subjected to a softmax operation to derive probabilities p (Equation 7 amd 8).

$$xc = concat(x1, x4); pl = Wxc + b \tag{7}$$

$$p = softmax(pl) \tag{8}$$

3.4 Loss Function

Cross Entropy loss is designed to minimize the difference between the predicted probabilities and true values, as follows:

$$lce = -\sum_{i=1}^{n}(p_i log(o(p_i)) + (1 - p_i) log(1 - o(p_i))) \tag{9}$$

3.5 Graph Explanation

GNNExplainer is used to derive explanations from the graph encoder network for interpreting the results and find underlying relations and causation. It works by taking a trained GNN model and its predictions as input, and returns explanations in the form of compact subgraph structures and subsets of influential node features. This model-agnostic approach can explain predictions of any GNN-based model on various graph-based machine learning tasks, including node classification, link prediction, and graph classification. GNNExplainer formulates explanations as rich subgraphs of the input graph, maximizing mutual information with the GNN's predictions. It achieves this by employing a mean field variational approximation to learn real-valued graph masks that select important subgraphs and feature masks that highlight crucial node features. Through this process, GNNExplainer offers insights into the underlying reasoning of GNN predictions, enhancing model interpretability and facilitating error analysis.

4 Experiments

4.1 Experimental Setup

Dataset. HateXplain, a benchmark hate speech dataset designed for addressing bias and interpretability is used. The dataset has hate speech targets labelled. This labelling is used to collect only Muslim-focused sentences and created a subset to work on this project. A 6:2:2 train, validation and test split is used.

Baselines. The baseline models are: CNN-GRU, BiRNN, BiRNN-HateXplain, BERT, BERT-HateXplain. Mentioned HateXplain-based models are fine-tuned on HateXplain dataset.

Implementation Details. Hugging Face transformers library is used to get embeddings from pre-trained BERT (bert-base-uncased) and BiRNN. The model is trained for 200 epochs with a learning rate of 0.001, using Adam optimizer. The experimental results in Table 1 show that our model achieves remarkable performance comparing to benchmarks with explaining occurring phenomenons.We utilized a single layer for each type of GNN, with a maximum tokenization length of 512 in the tokenizer and length of BERT embeddings (F) set to 128.

4.2 Experimental Results

Table 1 shows the performance of various models in detecting hate speech, highlighting accuracy and Macro F1 metrics. Traditional models like CNN-GRU and BiRNN show lower performance, with BiRNN-HateXplain offering slight improvements. BERT-based models perform better, particularly BERT-HateXplain. However, our proposed models, XG-HSI-BiRNN and XG-HSI-BERT, significantly outperform all others, with XG-HSI-BERT achieving the highest accuracy (0.741) and Macro F1 (0.747). These results demonstrate the superior effectiveness of our dual GNN approach in hate speech detection.

Table 1: Experimental Results (ă2191)

Model	Accuracy	Macro F1
CNN-GRU	0.628	0.604
BiRNN	0.591	0.578
BiRNN-HateXplain	0.612	0.621
BERT	0.692	0.671
BERT-HateXplain	0.693	0.681
XG-HSI-BiRNN (Ours)	0.742	0.737
XG-HSI-BERT (Ours)	0.751	0.747

5 Graph Explanation Case Study

For a given post, "How is all that awesome Muslim diversity going for you native germans? You have allowed this yourselves. If you do not stand and fight against this. You get what you asked for what you deserve!", the predicted classification was offensive towards Islam. As per the explainer, the neighbouring and self-tokens helped to classify this as offensive to Muslims are fight, Muslim diversity, brooks, rish, donald, syrian, schultz, typed. The text's association of "Muslim diversity" with potential blame and its confrontational tone in phrases like "stand and fight against this," combined with neighbouring tokens like syrians, brooks, syrians denoted negative sentiment.

6 Discussion

This study not only addresses the immediate challenge of identifying and explaining hate speech directed at Islam but also recognizes the broader impact of hate speech propagation on online platforms. The proliferation of Islamophobic language fosters intolerance, division, and hostility within communities, perpetuating harmful stereotypes and prejudices. By leveraging GNNs in our XG-HSI framework, we not only detect hate speech but also provide explanations for its occurrence, shedding light on the underlying factors driving such behaviour. GNNs excel in capturing complex relationships and patterns within data, enabling them to effectively identify instances of hate speech and elucidate the contextual nuances surrounding them. By leveraging the inherent structure of social networks and textual data, our approach offers a comprehensive understanding of how hate speech propagates in online discourse.

In future research, exploring the integration of multimodal data sources, such as images and videos, could enhance the robustness of hate speech detection models, particularly in detecting nuanced forms of Islamophobic content. Additionally, investigating the dynamic nature of online communities and incorporating temporal aspects into GNN architectures could provide deeper insights into the evolution of hate speech propagation and enable more proactive interventions to counter its spread.

7 Conclusion

Identifying and addressing Islamophobic hatred on social media is crucial for achieving harmony and peace. This research presents a novel method using GNNs to detect hate speech towards Islam. Empirical findings demonstrate that our model achieves exceptional performance, significantly outperforming all others, with XG-HSI-BERT achieving the highest accuracy (0.741) and Macro F1 (0.747). Explainability aspect of this approach is also very promising, as it provides insights into both correlations and causation. This further highlights the potential of GNNs in combating online hate speech and fostering a safer, more inclusive online environment.

Limitations

The limitations include the use of only one dataset, which, while sufficient for this initial exploration, should be expanded upon in future research to validate and extend our findings. Additionally, while Graph Neural Networks (GNNs) are known to be computationally intensive, especially with large-scale datasets, the relatively limited number of hate speech keywords suggests that GNNs may still be highly effective. Furthermore, more efficient GNN training methods are now available, which address some of the computational challenges in future applications.

Ethical Implications

Our work on using GNNs to detect hate speech targeting Islam carries significant ethical responsibilities. We focus on minimizing biases in the model to ensure fair treatment of all groups, emphasizing the need for transparency in how the model arrives at its decisions. By using interpretable GNN methods, we strive to provide clear explanations for the model's classifications, allowing for greater accountability. We also acknowledge the potential risks of misuse and take steps to prevent these, adhering to ethical guidelines that respect privacy and avoid unjust censorship.

Societal Implications

The societal impact lies in its potential to create a safer online environment by effectively identifying and mitigating Islamophobic content. By enhancing the detection accuracy and providing clear explanations for the identified hate speech, our model contributes to fostering more inclusive and respectful online communities. Additionally, our work highlights the importance of combating digital hate speech, which can lead to real-world harm. We aim to empower platforms and policymakers with tools that uphold freedom of expression while curbing harmful rhetoric, thus promoting social harmony and understanding.

Potential Risks

The application of our model presents several risks. One major concern is the potential for model misclassification, which could lead to false positives or negatives, impacting users unfairly. Additionally, there is a risk of over-reliance on automated systems, which might not capture nuanced contexts and could inadvertently suppress legitimate speech. Annotation errors can also induce bias, but as we used a previously peer-reviewed benchmark dataset, we hope those type of concerns are already addressed.

Acknowledgements

Sincere gratitude to the Computational Intelligence and Operations Laboratory (CIOL) for all their support. This work was presented at the Muslims in ML workshop (non-archival) at NeurIPS 2023, and thanks for their reviews, support, and the opportunity to present. Appreciation to all the reviewers for their valuable suggestions to improve the work.