HateNet: A Graph Convolutional Network Approach to Hate Speech Detection

Charles Duong
Brown University
Providence, RI, USA

Lei Zhang
Virginia Tech
Falls Church, VA, USA

Chang-Tien Lu
Virginia Tech
Falls Church, VA, USA

2022 IEEE International Conference on Big Data (Big Data) | 978-1-6654-8045-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/BigData55660.2022.10020510

***Abstract*—The COVID-19 pandemic has caused hate speech on online social networks to become a growing issue in recent years, affecting millions. Our work aims to improve automatic hate speech detection to prevent escalation to hate crimes. The first challenge in hate speech research is that existing datasets suffer from quite severe class imbalances. The second challenge is the sparsity of information in textual data. The third challenge is the difficulty in balancing the tradeoff between utilizing semantic similarity and noisy network language. To combat these challenges, we establish a framework for automatic short text data augmentation by using a semi-supervised hybrid of Substitution Based Augmentation and Dynamic Query Expansion (DQE), which we refer to as SubDQE, to extract more data points from a specific class from Twitter. We also propose the HateNet model, which has two main components, a Graph Convolutional Network and a Weighted Drop-Edge. First, we propose a Graph Convolutional Network (GCN) classifier, using a graph constructed from the thresholded cosine similarities between tweet embeddings to provide new insights into how ideas are connected. Second, we propose a weighted Drop-Edge based stochastic regularization technique, which removes edges randomly based on weighted probabilities assigned by the semantic similarities between Tweets. Using 3 different SubDQE-augmented datasets, we compare our HateNet model using eight different tweet embedding methods, six other baseline classification models, and seven other baseline data augmentation techniques previously used in the realm of hate speech detection. Our results show that our proposed HateNet model matches or exceeds the performance of the baseline models, as indicated by the accuracy and F1 score.**

***Index Terms*—hate speech, dynamic query expansion, graph convolutional network, social media data mining, machine learning**

I. Introduction

The growing popularity of online interactions through social media in every demographic [1] (age, race, gender, etc.) has led to both positive and negative impacts. While social media has revolutionized information sharing, it has also become a medium for increasing hateful speech [2]. The Pew Research Center [3] has reported that **41%** of Americans have personally experienced some form of online harassment and **66%** of Americans reported to have witnessed abusive or harassing behavior towards others online. On top of this, the COVID-19 pandemic has made hateful speech an increasingly worrying threat as hate, and by extension hatecrimes, toward Asian Americans and Pacific Islanders have increased as a result of many people blaming the pandemic on these ethnic groups [4]. The strong connection between hate speech and actual hate crimes [5] make the detection of hate speech a vital task. On top of being a precursor to potential hate crimes, hateful speech can have deep impacts on an individual such as heightened stress and anxiety [6], lowered academic performance and self-esteem [7], alcohol and drug use [8], and in extreme cases, suicide.

Current methods of combating hate speech are primarily through informing consumers. This involves educating children [9] and advocating on social media. Facebook, Twitter, Tiktok, and other leading social media platforms prohibit hateful speech in their respective community guidelines and also have passive reporting procedures built-in to their application [10]. However, these social media platforms are yet to utilize active hate speech detection tools. An active system is critical as only an estimated 12% of incidents are reported [11]. In this work, we leverage recent developments in natural language processing, casting the **hate speech detection problem** into the field of machine learning-powered sentiment analysis and seek to stop the inflicted verbal harm before it escalates to hate crimes.

One of the challenges facing current hate speech research is the **class imbalance problem** leading to severe class biases, hindering the performance of models (see Section IV). This is reflected in the three datasets used in this paper, **RSN** [12], **HON** [13], **HANS** [14], which all showed imbalances in the minority class. For example, **RSN** had only 26.3% of tweets as sexist whereas 73.7% of its tweets were labeled as neutral, **HON** had only 5.7% of its data labeled as hateful whereas 77.2% of its data is labeled as offensive, and **HANS** had only 5% of its data labeled as Hateful and 27.2% as abusive whereas 53.9% of its data was labeled as neutral.

The second challenge in hate speech research is the **sparsity of information in textual data**. In social media, hateful comments are a very clear minority making the data more sparse than in other event detection tasks.

The third challenge in hate speech research is **balancing the trade-off between utilizing semantic similarity and noisy network language**. On one hand, we want to utilize as much information about the semantic similarity among tweets as possible, leading to a relatively dense graph structure with more edges, and thus, more information about the relationship between tweets. On the other hand, due to the noisy Internet language usages, the semantic similarities are not always

978-1-6654-8045-1/22/$31.00 ©2022 IEEE

accurate especially between the less correlated tweets. In hate speech detection, the data is particularly noisy compared to other tasks in social media as the language used is more informal, consisting of subtle nuances and hidden subtexts in language, including sarcasm, irony, slang, nicknames, and double negatives.

In this paper, we propose **HateNet** to address these challenges. Our main contributions of this work are:

1) **Development of an active system to detect hate speech on social media.** We cast the **hate speech detection problem** into the field of machine learning-powered sentiment analysis, leveraging recent developments in natural language processing, seeking to stop the inflicted verbal harm before escalation to hate crimes.
2) **Formulation of a Short Text Data Augmentation technique using Substitution Augmentation and Dynamic Query Expansion.** We improved upon Short Text Data Augmentation by establishing a procedure to iteratively expand an existing dataset by utilizing a hybrid of the DQE algorithm and Substitution Based Augmentation in a novel way to combat the **class imbalance problem**. We successfully leverage the algorithm to generate new data points in the minority class in a semi-supervised fashion. This procedure automates most of the data collection process as well as facilitates the collection of balanced data, solving the class imbalance problems faced by almost every dataset.
3) **Construction of a graph structure of tweet embeddings and implementing a Graph Convolutional Network to capture semantic connections of tweets.** GCNs are a previously unexplored classification method in the hate speech detection task. We propose a GCN-based framework that takes in a graph generated by the thresholded semantic cosine similarities between every tweet, allowing for the effective propagation of labels across tweets with similar main ideas, providing new insights with how ideas are connected. If sentence or word embeddings can be constructed (with the aim of encoding a primitive "knowledge" or "conceptual" graph), our GCN may be able to take advantage of these connections. This helps combat the **sparsity of information in textual data problem**, and we show promising results for its application in hate speech classification.
4) **Formulation of a DropEdge-based stochastic regularization technique to prevent overfitting and oversmoothing.** We improved upon the DropEdge regularization technique by creating weighted probabilities for each graph edge, based on the semantic similarity score.
5) **Comprehensive experiments to validate the effectiveness and efficiency of the proposed techniques.** We evaluated a combination of 5 tweet embedding methods, 7 classification models, and 7 data augmentation techniques on 3 datasets for the hate speech detection task. Results showed that the proposed methods consistently outperformed competing baseline methods seen in traditional natural language learning models. By meticulously experimenting with so many comparison methods on real-world datasets, we demonstrate the performance of the HateNet in full context.

II. Related Works

Hate Speech Detection via Social Media Analysis. A large body of existing work [15]–[17] regards hate speech detection as various classification tasks and performs the analysis based on handcrafted lexcial-syntactic text features. For example, Wanner et al. [18] aimed to detect abusive speech targeting specific group characteristics by applying binary classifiers on features explicitly generated by the template-based strategies. As the growing impact on the social media platform such as Facebook and Twitter, more and more researchers [19]–[23] start to explore the ways to actively detect and moderate online hate speech. For example, Waseem et al. [12] collected a Twitter corpus using the common slurs and terms used pertaining to religious, sexual, gender, and ethnic minorities, and analyzed the impact of various extra-linguistic features in conjunction with n-grams for hate speech detection on this dataset. Cao and Lee [24] used an adversarial deep generative reinforcement learning model to detect hate speech while also addressing the class imbalance problem. Overall, none of these papers investigated using a Graph Convolutional Network (GCN) for hate speech detection. Some papers [12] were able to address the sparsity of textual data but ultimately did not address the class imbalance problem. Other papers [24] were able to address the class imbalance problem but were not successful in addressing the sparsity of textual data. Our proposed model is the first of our knowledge to utilize a Graph Convolutional Network for the hate speech detection task and also addresses both the sparsity of textual data and the class imbalance problem.

Deep Learning for Text Classification. Recently, researchers [25]–[27] discovered that the implicit features extracted by the deep learning based models could outperform the explicitly manually summarized features. For example, Djuric et al. [28] demonstrated that textual content representations generated via embedding models perform better on downstream applications than traditional term-based bag-of-words features. Going beyond the analysis on pure text data, recently, graph neural network [29], as a deep learning technique that focuses more on modeling the linguistic behaviors and dependencies between information, attracts researchers' attention. For example, Mishara et al. [30] proposed a model based on graph convolutional networks (GCNs) for abusive language detection. The introduction of GCN architecture is claimed to be able to capture and identify not only community structure but also the linguistic behavior of the users. Wang et al. [31] proposed a GCN-based classifier that aimed to detect online cyberbullying behaviors by modeling the semantic structures between tweet embeddings. All of the mentioned papers use fully connected, dense, graphs that lead to overfitting and oversmoothing. In our paper however, we have formulated a weighted DropEdge-based stochastic regularization technique that enables textual GCNs to address overfitting and oversmoothing by randomly dropping edges based on probabilities assigned by the semantic similarity scores.

Data Augmentation for Text Classification. A number of methods have been utilized in natural language processing to combat class imbalance. Upsampling and downsampling have been used as a simple technique to augment data in previous NLP tasks. Rizos et al. [32] proposed three data augmentation techniques for NLP, applying them to hate speech detection. HateGAN [24] proposed a deep generative reinforment learning model which augments the dataset during learning. Guzman-Silverio et al. [33] compared three different augmentation techniques [34]–[36] to detect aggressiveness in Mexican Spanish speech. Beddiar et al. [37] explored using back translation and paraphrasing to expand data for hate speech detection. None of the mentioned techniques are able to identify and target the most representative part of the dataset. Our paper, however, utilizes Dynamic Query Expansion as a tool to identify the most representative part of the dataset and targets augmentation toward that subsection of the dataset.

III. Problem Formulation

A. Hate Speech Detection in Graph Data

Let $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N) \in \mathbb{R}^{N \times F}$ be the set of textual posts from the social media platforms, where N is the number of posts in our input and F is the number of features that preserve the semantic meanings of the posts.

Definition I: *Semantic Similarity Graph for Online Posts.* Let each online post, $\mathbf{x}_i$, represent one node in the graph and construct a fully-connected graph, $\mathbf{G}^* = (\mathbf{V}, \mathbf{E}^*)$, of the online posts given in the dataset, $\mathbf{X}$, where $|\mathbf{V}| = N$, and $\mathbf{V}$ represents the corresponding vertex set for the set of online posts, dataset $\mathbf{X}$. Let $\mathbf{E}^*$ be the edge set for the fully-connected graph, $\mathbf{G}^*(|\mathbf{E}^*| = \binom{N}{2})$. When the filtering criteria ϵ holds (see section IV), we construct the semantic similarity graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ representing the semantic distance between the posts, where $\mathbf{E} \subseteq \mathbf{E}^*$.

Given the graph representation, now we can formulate the hate speech detection problem in a graph. We define a vector $\mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N) \in \{b_k | k = 1, 2, \ldots, K\}^N$, where b_k is the k^{th} hate speech class (racism, sexism, etc.) in the labeled dataset; K is the number of targeted hate speech classes. Thus, our hate speech detection problem is defined as follows: given the input dataset $\mathbf{X}$, the filtering criteria ϵ, and the respective labels Y, how can we find an optimal solution to accurately classify the type of hate speech activity when given a new verbally abusive online post? Mathematically, we formulate the problem as learning a function $\mathbf{F}^*$ parameterized with W, which maps $\mathbf{X}$ to $\mathbf{Y}$: $\mathbf{F}^*(\mathbf{X}) \rightarrow \mathbf{Y}$. We minimize the loss function as follows:

$$\underset{\mathbf{E},\mathbf{W}}{\arg\min}\, \mathcal{L}(\mathbf{Y}, \mathbf{F}(\mathbf{X}, W, \mathbf{E})) \tag{1}$$

B. Short Text Data Augmentation with Dynamic Query Expansion for Hate Speech Detection

We propose a Short Text Data Augmentation method to further augment the initial hate speech dataset in order to overcome the challenge of unbalanced data in online posts. The input to our Short Text Data Augmentation is the initial collection of online posts, $\mathbf{X}^k = (\mathbf{x}_1^k, \mathbf{x}_2^k, \ldots, \mathbf{x}_{p^k}^k)$, where p_k is the initial number of online posts for the k^{th} target hate speech class b_k. The initial size of the dataset is defined as $P = \sum_{k=1}^{K} p_k$. From A, $\mathbf{X} \in \mathbb{R}^{N \times F}$ is defined as the dataset we require, thus, the number of online posts expanded by our Short Text Data Augmentation is $N - P$. Let $\mathbf{X}_+^k$ denote the subspace of the target online posts which, for this task, are the posts containing the relevant hate speech queries.

Definition II: *Seed Query.* We define a seed query, $\mathbf{Q}_0$, as a manually selected and typed dependency query targeted for a certain type of event.

Definition III: *Expanded Query.* We define an expanded query, $\mathbf{Q}_k$, as a typed dependency query that is automatically generated as an output of the Dynamic Query Expansion algorithm based on a set of seed queries and a collection of online posts $\mathbf{X}^k$.

Short Text Data Augmentation Task: Given a small set of seed queries, $\mathbf{Q}_0$, and an initial collection of online posts, $\mathbf{X}^k$, the task of our Short Text Data Augmentation is to iteratively expand $\mathbf{X}_+^k$ and $\mathbf{Q}^k$ until all the relevant online posts are included.

IV. Methodology

We present our proposed model, HateNet, which addresses active hate speech detection. Figures 1 and 2 show the overall architecture of our proposed model, HateNet, comprising four main components: short text data augmentation, semantic similarity graph, graph convolution, and weighted DropEdge.

The input of our model is a set of Tweets. We start by running the Tweets through our novel Short Text Data Augmentation technique, called **SubDQE**. It starts by running a Dynamic Query Expansion (DQE) on the entire dataset to identify expanded queries that help target the imbalanced classes and representative data (Parts 1 and 2 of Figure 1). Then, a substitution technique is applied on the representative data that utilizes synonym replacement and word embedding vector closeness to generate new data (Part 3 of Figure 1). A detailed description of the short text data augmentation method can be found in Section IV A.

After the data augmentation, the augmented dataset is then used to construct a fully connected graph using the word/sentence embeddings as nodes and the semantic cosine similarities between embeddings as edges. Textual graph convolution is then applied on the graph (Figure 2). At each training epoch, our weighted DropEdge technique drops out a certain rate of edges of the input graph based on probabilities assigned using the semantic similarity scores of edges. The details of the textual graph construction and convolution and weighted DropEdge can be found in Sections IV B, IV C, and IV D.

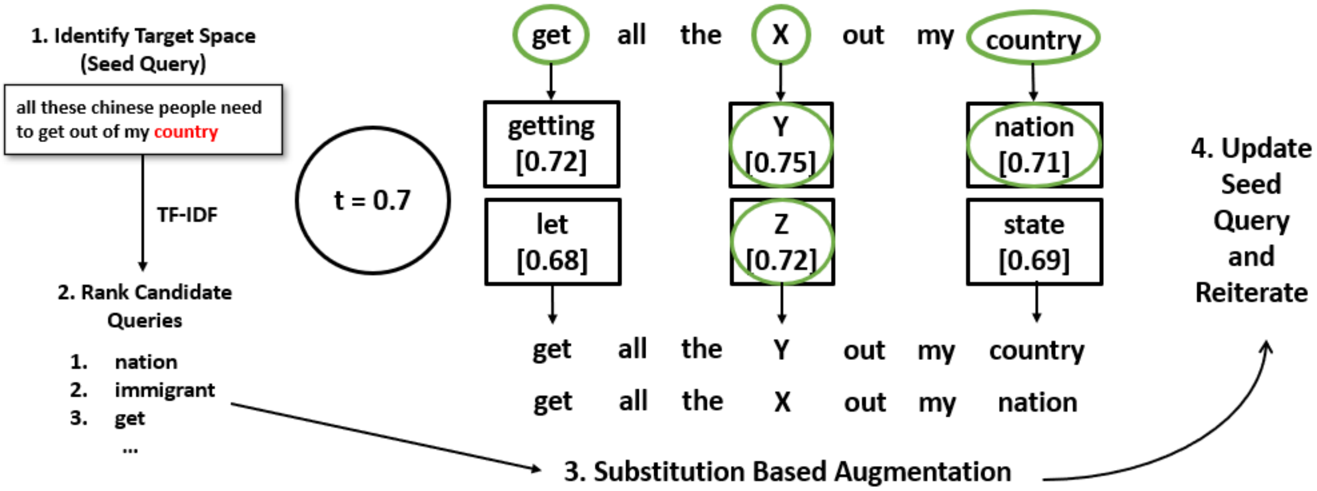
1. Identify Target Space (Seed Query)
all these chinese people need to get out of my country
TF-IDF
2. Rank Candidate Queries
1. nation
2. immigrant
3. get
...
t = 0.7
get
all
the
X
out
my
country
getting [0.72]
let [0.68]
Y [0.75]
Z [0.72]
nation [0.71]
state [0.69]
get all the Y out my country
get all the X out my nation
3. Substitution Based Augmentation
4. Update Seed Query and Reiterate

Fig. 1. Overview of Our SubDQE Process for Short Text Data Augmentation. Perturbed copies of an original short text are generated based on high cosine similarity and POS-tag match after queries are flagged. In part 3, note that although 'getting' exceeds the similarity threshold, it is a gerund, while 1get' is a regular verb.

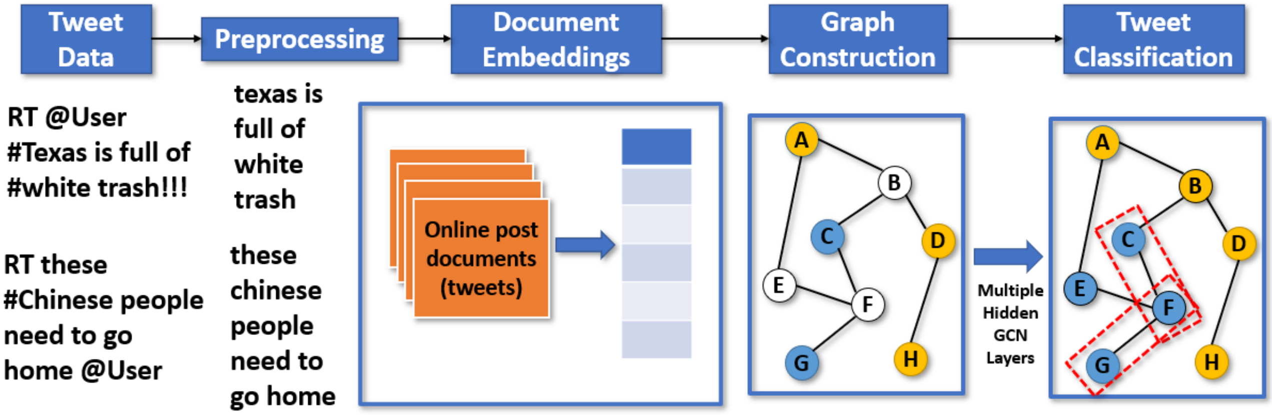
Tweet Data
Preprocessing
Document Embeddings
Graph Construction
Tweet Classification
RT @User #Texas is full of #white trash!!!
RT these #Chinese people need to go home @User
texas is full of white trash
these chinese people need to go home
Online post documents (tweets)
A
B
C
D
E
F
G
H
Multiple Hidden GCN Layers

Fig. 2. Overview of Our Proposed HateNet Framework for Hate Speech Detection.

A. Short Text Data Augmentation

Hate Speech datasets suffer from severe class imbalance including a 5.77% Hate Speech class in **HON** and a 4.96% Hateful class in **HANS** (see Section IV A). We must use data augmentation to combat this imbalance but we cannot use techniques from non-NLP domains, so we want to utilize new ones that satisfy three desiderata: change the input to the neural network (new sample), be class-conditional, where no manual labeling is required (same class), produce perturbed versions of the original sentence samples (same meaning). We propose a novel hybrid of DQE and Substitution Based Augmentation, which we call **SubDQE** (see Figure 1).

The purpose of SubDQE is to automatically identify the most representative words or features (which we call candidates) from a textual dataset. A rudimentary input of seed queries is hand-selected that vaguely captures the main idea or theme being queried for. The traditional DQE workflow has been modified to be a semi-supervised text augmentation algorithm consisting of four steps:

1) Identify Target Space. A seed query set is used as a filter to identify the most representative part of the dataset, enabling the subsequent query generation to stay focused on a specific class.

2) Rank Candidate Queries. Term-frequency inverse-document frequency (TF-IDF) weighting is applied to each word in each tweet of the target space, and the 30 highest ranking words become our candidate query set for augmentation.

3) Substitution Based Short Text Data Augmentation. This step is our modification to the DQE algorithm. After ranking candidate queries, we concatenate the top 30 candidates and expand the dataset by making relevant substitutions of words with synonyms as seen in Figure 1 Part 3. Pre-trained neural word embeddings are utilized, allowing us to determine the relative semantic similarity between each word in the vocabulary space of a text corpus. We are given a training sequence of words (our expanded query set), $\{q_t\}$ with corresponding embedding vectors $\{\mathbf{v}_t\}, \forall t \in \{1 : T\}$.

For each center query q_t, we predict the surrounding context words q_o within a radius m. For example, when using Word2Vec [38] as our embedding method, we maximize the probability of any context word given the current center word $q_{\hat{o}}$, while minimizing the probability of a random word from the vocabulary (i.e., negative sampling):

$$J_t(\theta) = \log \sigma(\mathbf{v_t}^\top \mathbf{v_o}) + \sum_{\hat{o} \sim P(q)} \log \sigma(\mathbf{v_t}^\top \mathbf{v_{\hat{o}}}), \tag{2}$$

where $P(q)$ represents a distribution that places higher sampling probabilities on less frequent words, θ is the entire array of embedding vectors, and σ is a nonlinear scoring function. Our use of neural embeddings is useful in substitution based text augmentation, as it is a central part of the main model.

This substitution based augmentation builds on the definition of a semantic similarity score between the equal-size vector representations denoted by v_i in Equation 2. We denote $c \in [0, 1]$ as the semantic similarity score between two embeddings $\mathbf{v}_i$ and $\mathbf{v}_j$: $c = \frac{\mathbf{v}_i^T \cdot \mathbf{v}_i^T}{\|\mathbf{v}_i\| \cdot \|\mathbf{v}_j\|}$ where $\|\mathbf{v}\|$ is the norm of the vector $\mathbf{v}$. For each query q_i of the input query set, this method describes a substitution that is determined by two factors. Any candidate replacement word must exceed the thresholded cosine distance, t, where $t \in [0, 1]$, *and* it must match the POS-tag assigned to the word. Our intuition for setting both of the above requirements is that two words must have been used in sufficiently equals contexts, allowing one to be replaced with the other without changing the sentence semantics, helping satisfy our previously defined desiderata.

4) Update Seed Query and Reiterate. The previous 3 steps are repeated, using the top 30 candidate queries from the current iteration as the seed query set for the next iteration. This process repeats until the difference between candidate significance values, measured by the TF-IDF weights, from the last iteration and the current iteration is within a predefined threshold value. This difference is calculated by:

$$\frac{\sum_{i \in C_t \setminus C_{t+1}} w_t(i) + \sum_{j \in C_{t+1} \setminus C_t} w_{t+1}(j)}{\sum_{k \in C_t} w_t(k) + \sum_{l \in C_{t+1}} w_{t+1}(l)}, \tag{3}$$

where t denotes the iteration, C_t denotes the set of candidates at iteration t, and w_t denotes the significance value of the candidate at iteration t.

Our semi-supervised short text data augmentation method, which we refer to as **SubDQE**, produces a high quality dataset with few, if any, outliers upon a brief visual check. The strength of our method is that it encourages the downstream task to place lower emphasis on associating single words with a label and instead place higher emphasis on capturing similar sequential patterns, i.e. the context of hate speech. Our use of the DQE algorithm and Substitution Based Augmentation is novel because it is the first of our knowledge that can (i) augment current datasets with the aim of solving class imbalance, (ii) build off of already labeled data in a semi-supervised fashion such that no manual labeling is required, (iii) keep the same meaning while emphasizing sequential patterns, (iv) identify the most representative words to target augmentation, and (v) run separate processes for each of the classes.

B. Semantic Similarity Graph for Online Posts

We construct graph structures that can capture the semantic correlations or similarities between the online posts (e.g., tweets on Twitter). The graph is constructed based on the text similarities inferred according to text embedding models trained on large corpus. According to Definition I, we describe the online posts graph with a partial graph representation $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$, where $\mathbf{A}$ denotes the adjacency matrix of the graph, and $\mathcal{V}$ is the set of vertices in the graph which represents the set of online posts $\mathbf{X}$. For online post $\mathbf{x}_i$ and $\mathbf{x}_j$ which are represented by vertices v_i and v_j respectively in $\mathcal{V}$ of graph $\mathcal{G}$, the existence of edge $i \iff j$ denoted by the variable $A_{i,j}$ in the adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ can be formulated as:

$$\mathbf{A}_{i,j} = \begin{cases} S_c(\mathbf{x}_i, \mathbf{x}_j), & \text{if condition } \zeta(\mathbf{x}_i, \mathbf{x}_j) \text{ holds,} \\ 0, & \text{otherwise,} \end{cases} \tag{4}$$

where $S_c(\mathbf{x}_i, \mathbf{x}_j)$ is the cosine similarity between the post representations; ζ is a textual similarity condition that calculates the textual similarity between the input targets $\mathbf{x}_i$ and $\mathbf{x}_j$. In our HateNet model, we select the ζ condition as: $S_c(\mathbf{x}_i, \mathbf{x}_j) \geq \epsilon$. Such edge exists only if $\mathbf{x}_i$ and $\mathbf{x}_j$ have a meaningful semantic similarity to each other, as ϵ is empirically chosen to trim insignificant edges while preserving the relational information that can be helpful for improving the performance in the downstream tasks.

C. Graph Convolution for Online Posts

We utilized a Graph Neural Network (GNN) to help learn a better semantic representation of posts using the semantic similarity graph described in the last section. Given the graph representation of the online posts $\mathbf{X}$ and $\mathbf{G} = (\mathbf{V}, \mathbf{E}, \mathbf{A})$, the spectral graph convolution is operated in the Fourier domain. An essential variable for graph convolution in the spectral domain is the Laplacian matrix $\mathbf{L}$, which is defined by $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where $\mathbf{D} \in \mathbb{R}^{N \times N}$ is the degree matrix, and $\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij}$. Then we further calculate the normalized Laplacian matrix by $\mathbf{L}_N = I_N - \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}} \in \mathbb{R}^{N \times N}$, where I_N is an identity matrix. The normalized Laplacian matrix $\mathbf{L}_N$ is symmetric and semi-positive definite. The spectral decomposition of $\mathbf{L}_N$ can be represented as $\mathbf{L}_N = \mathbf{U} \Lambda \mathbf{U}^T$, where $\mathbf{U}$ is comprised of orthogonal and normalized eigenvectors $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_N] \in \mathbb{R}^{N \times N}$ and $\Lambda = diag([\lambda_1, \ldots, \lambda_N])$ is the combination of the eigenvalues $\lambda \in \mathbb{R}^N$. Then, the convolution can be defined in the spectral domain as:

$$y = \sigma\left(\mathbf{U} g_\theta(\Lambda) \mathbf{U}^T x\right), \tag{5}$$

where x is the graph signal, y is the convolution output, g_θ is the low-pass graph filter of the convolution, and σ is the activation function. This formation is feasible of the spectral convolution; however, the computation complexity is high for large graphs. To reduce the computation complexity, a Chebyshev polynomial approximation can be applied to the low-pass filter $g_\theta(\Theta)$:

$$g_\theta(\Lambda) \approx \sum_{m=0}^{K} \theta_m T_m\left(\tilde{\Lambda}\right), \ \tilde{\Lambda} = \frac{2}{\max(\lambda)}\Lambda - I_N. \tag{6}$$

This approximation was first proposed by Hammond et al. [39]. Here the Chebyshev polynomials are recursively defined as $T_m(x) = 2xT_{m-1}(x)$. To further simplify the graph convolution and improve the efficiency, Kipf et al. [29] limit the number of order m to be 1, along with the max eigenvalue to be 2. One layer of Graph Convolution Network is now represented as

$$\mathbf{Y} = (\mathbf{D} + I_N)^{-\frac{1}{2}}(\mathbf{A} + I)(\mathbf{D} + I_N)^{-\frac{1}{2}} X\Theta. \tag{7}$$

In the prediction layer, the target is the similarly formulated graph representation $\mathbf{G}^* = (\mathbf{V}^*, \mathbf{E}^*)$ where $\mathbf{V}^*$ additionally contains the new online post observations up for prediction and $\mathbf{E}^*$ is the thresholded semantic similarities between the vertices of the updated $\mathbf{V}^*$. The input $X \in \mathbb{R}^{N\times K}$ is the hidden representation of online posts generated by the sentence-based post encoders, and the output $Y \in \{b_k | k = 1, 2, \ldots, K\}^N$ is the predicted hate speech labels based on both the contents and the connectivity of the online posts. Information sharing between the connected posts/nodes can be modeled by the filter g_θ. Thus, HateNet can be utilized as an appropriate model for classifying hate speech online posts.

D. Weighted DropEdge for Textual Graph

To help relieve the over-smoothing issue, we propose a DropEdge-based stochastic regularization technique. DropEdge has been proven to be effective on preventing over-fitting and over-smoothing in GNNs. The original DropEdge randomly removes edges from the graph by drawing independent Bernoulli random variables at each iteration. More specifically, if we denote the adjacency matrix used in the l-th layer as A^l_{drop}, then its relation with the ground truth A is

$$A^l_{drop} = A \odot Z^{(l)} \tag{8}$$

where $Z^{(l)}$ is a sparse matrix expanded by a random subset of size $|\mathbb{V}|p$ from the original edges $\mathbf{E}$, where p is a pre-defined hyperparameter of the probability to drop edges.

DropEdge does not fit well with our textual graph topology because it was designed for graphs with binary edges and doesn't consider the edge weights. However, the textual graphs as described in Section IV B are edge weighted graphs. As the edge weights in $\mathbb{V}$ is in the range of $[\epsilon, 1]$, the weight DropEdge

$$z^{(l)}_{i,j} = p(1 - \frac{e^{A_{i,j}} B_{i,j}}{\sum_{j=1}^{|\mathbb{V}|} e^{A_{i,j}} B_{i,j}}) B_{i,j} \tag{9}$$

where p is the pre-defined maximum probability to drop edges, and B is the Boolean matrix of A. Eq. (9) enforces to drop edge with higher probability on existing edges with lower weights because these edges are most likely to have misleading information on contextual similarity relationship.

E. Time Complexity Analysis

The time complexity of SubDQE is

$$O\left(l \cdot \left(|\mathcal{F}| \cdot n_{E_{TF}} + |\mathcal{T}| \left(n_{E_{TF}} + n_{E_{TT}}\right)\right)\right), \tag{10}$$

where $\mathcal{F}$ refers to feature nodes, $\mathcal{T}$ refers to tweet nodes, $n_{E_{TF}} \ll |\mathcal{F}|$ is the average number of connections between a tweet node and feature nodes, $n_{E_{TT}} \ll |\mathcal{T}|$ is the average number of connections from a tweet node to other tweet nodes, and l is the number of the iterations of SubDQE. Typically, $l \leq 10$.

The time complexity of our Weighted DropEdge GCN is

$$O\left(L\,|E|\,F^2 + LN^2F\right) \tag{11}$$

where L is the number of layers, E refers to the edges, N is the number of posts in our input data, and F is the number of features that preserve the semantic meanings of the posts.

V. Experiment

A. Experiment Setup

In this paper, we used three datasets from the literature to train and evaluate our model. These publicly available datasets were selected because they are commonly used in twitter hate speech studies and will achieve state of the art performance. Although all datasets addressed the category of hateful speech, they used different strategies of labeling the collected data. These datasets are referred to as **RSN** [12], **HON** [13], and **HANS** [14]. Note that the third dataset was published after Twitter's character limit change from 140 to 280 in 2017.

1) **Waseem and Hovy [12] dataset** includes roughly 17,000 tweet IDs labeled as Racist, Sexist, or Neutral. The annotation for these tweets were performed using CrowdFlower. Because this dataset was in the form of tweet IDs, the Twitter API[1] was used to retrieve the actual tweet content of which only 61.3% of the content could be retrieved given that many tweets were taken down or removed since the publication of the dataset. Of the tweets were retrieved, only 11 or 0.1% were labeled as racist so we decided to get rid of this class all together making the dataset consist of only sexist and neutral tweets. We refer to this dataset as **RSN**.
2) **Davidson et al. [13] dataset** includes roughly 25,000 tweets labeled as Hate speech, Offensive, or Neutral. These tweets were obtained using the hatebase[2] lexicon to filter tweets for common abusive terms, and then pulling all the tweets from all the users selected (approximately 85 million). The 25,000 tweets were a random sample taken from the overall pool of tweets and were

[1] https://developer.twitter.com/en/docs/twitter-api

[2] https://hatebase.org/

TABLE I
OVERVIEW OF DATASETS BEFORE AND AFTER SUBDQE-AUGMENTATION

Dataset	Class % Before				Total Before	Class % After				Total After
RSN	**R**	**S**	**N**				**S**	**N**		
	0.1	26.2	73.7		15,000		50	50		15,000
HON	**H**	**O**	**N**			**H**	**O**	**N**		
	5.7	77.4	16.7		15,000	26.4	40.3	33.3		15,000
HANS	**H**	**A**	**N**	**S**		**H**	**A**	**N**	**S**	
	5	27.2	53.9	14	15,000	25	25	25	25	15,000

labeled using CrowdFlower. We refer to this dataset as **HON**.

3) **Founta et al. [14] dataset** includes roughly 100,000 tweets labeled as Hateful, Abusive, Normal, and Spam. These tweets were labeled using CrowdFlower by annotators who meet certain demographic requirements. We refer to this dataset as **HANS**.

Our main contribution was utilizing a hybrid of Dynamic Query Expansion and Substitution Based Augmentation (see Section IV) to increase the number of samples of the minority class in a semi-supervised manner. We have found that it is very important to use regular downsampling for all cases in order to prevent the model from almost exclusively predicting only the majority class. At every epoch we used a different downsampled version of the training set as follows: all the samples of the minority class were included, but for the other classes we sample without replacement a number of data samples that is equal to the number of samples of the minority class. This ensured that the model receives training samples from each class at the same average frequency and also has the entire training set available to it over the entire training course.

B. Comparison Methods

The use of embedding methods was required in order to convert textual data into mathematical vectors that could be used as inputs to machine learning models. The goal was to generate representative vectors, where similar words have similar vectors. The following embedding methods were used in our experiment: Bag of Words (BOW), Term Frequency-Inverse Document Frequency (TF-IDF), word2vec [40], GloVe [41], fastText [42], BERT [43], DistilBERT (DBERT) [43], SentenceBERT (SBERT) [44].

To demonstrate the performance of HateNet in full context, six other traditional machine learning classifiers that have precedent in previous hate speech detection studies [45] were experimented on: Logistic Regression (LR), Naïve Bayes (NB), k-nearest neighbors (KNN), Support Vector Machine (SVM), XGBoost (XGB), and Multi-Layer Perceptron (MLP). All of these methods were implemented with sklearn.

Our SubDQE data augmentation is also compared with seven other data augmentation techniques previously used in hate speech detection studies: Upsampling (Up), Downsampling (Down), Substitution Augmentation (SubAug) [32], Word Position Augmentation (PosAug) [32], Neural Generative Augmentation (GenAug) [32], Back Translation (BT) [37], and Paraphrasing (Para) [37].

With our comparison methods in place, we went through five steps for each tweet embedding + classifier model combination: word embedding generation,[3] feature extraction, normalization, running/tuning with 5-fold stratified cross-validation, and experimental analysis. This process was repeated for all 3 datasets.

C. Experimental Design

The purpose of our experiment is to investigate the efficacy of different tweet embedding and classifier model combinations for detecting hate speech across three datasets. The independent variable was the combination of tweet embedding and classifier model used. The dependent variable was the resulting performance of the combination, measured by the evaluation metrics that we selected: accuracy and F1 score. The accuracy, F1 score, and Hate Recall were all used as measurements. Different baseline models and our HateNet model were tested with and without data augmentation. Our null hypothesis is that GloVe+HateNet will have no statistically significant difference between the evaluation metrics across the levels of IV. Our alternative hypothesis is that GloVe+HateNet will have a statistically significant difference between the evaluation metrics across the levels of IV.

D. Hate Speech Detection Results

Our experiments show that Sentence BERT+HateNet outperforms every other method on all 3 datasets in both accuracy and F1 scores (see Tables II III IV). Of the baseline models, SVM does very well and has proven to be generally effective in previous literature. BOW and TF-IDF are very simple embeddings that were tested: BOW only counts word frequencies and TF-IDF weights those frequencies. When combined with XGBoost, a simple classifier that uses decision trees, the results showed that simple models, while they are not the most interpretable, may prove to be as effective as complex models. The promising outcomes fo the decision-based quality of XGBoost suggests that vocabulary may be a strong discriminating factor to classify hate speech.

Apart from simple models, SBERT was the generally most effective embedding method across the classifier models. This made sense as SBERT is one of the current State-Of-The-Art sentence embedding methods for natural language

[3] For BOW, and TF-IDF, no word embeddings were generated; the output of these models were already features.

TABLE II
Performance of HateNet and Baseline Models on the HON dataset where Acc is Accuracy and M-F1 is Macro-F1 Score

HON	**LR**		**NB**		**KNN**		**SVM**		**XGB**		**MLP**		**HateNet**	
	Acc	**M-F1**	**Acc**	**M-F1**	**Acc**	**M-F1**	**Acc**	**M-F1**	**Acc**	**M-F1**	**Acc**	**M-F1**	**Acc**	**M-F1**
SBERT	0.807	0.810	0.739	0.731	0.718	0.702	0.831	0.832	0.819	0.821	0.829	0.824	**0.845**	**0.843**
BERT	0.769	0.766	0.601	0.597	0.621	0.624	0.771	0.772	0.761	0.762	0.791	0.787	0.785	0.791
DBERT	0.811	0.807	0.661	0.657	0.699	0.703	0.814	0.812	0.808	0.806	0.822	0.831	0.810	0.816
GloVe	0.791	0.796	0.621	0.623	0.645	0.648	0.820	0.819	0.817	0.814	0.801	0.803	0.809	0.811
W2V	0.761	0.759	0.633	0.634	0.633	0.639	0.812	0.816	0.813	0.817	0.810	0.808	0.791	0.789
FastText	0.751	0.755	0.545	0.543	0.599	0.604	0.800	0.796	0.801	0.806	0.781	0.777	0.772	0.776
TF-IDF	0.759	0.762	0.729	0.725	0.145	0.152	0.571	0.573	0.825	0.826	0.661	0.665	0.651	0.648
BOW	0.760	0.763	0.701	0.703	0.334	0.331	0.573	0.572	0.831	0.828	0.693	0.694	0.689	0.685

TABLE III
Performance of HateNet and Baseline Models on the HANS dataset where Acc is Accuracy and M-F1 is Macro-F1 Score

HANS	**LR**		**NB**		**KNN**		**SVM**		**XGB**		**MLP**		**HateNet**	
	Acc	**M-F1**	**Acc**	**M-F1**	**Acc**	**M-F1**	**Acc**	**M-F1**	**Acc**	**M-F1**	**Acc**	**M-F1**	**Acc**	**M-F1**
SBERT	0.837	0.841	0.771	0.767	0.746	0.739	0.863	0.865	0.852	0.854	0.841	0.837	**0.877**	**0.873**
BERT	0.801	0.798	0.634	0.631	0.654	0.651	0.801	0.805	0.790	0.793	0.814	0.816	0.823	0.821
DBERT	0.842	0.846	0.694	0.690	0.732	0.730	0.846	0.843	0.841	0.837	0.854	0.859	0.841	0.843
GloVe	0.822	0.829	0.654	0.657	0.677	0.672	0.852	0.849	0.845	0.842	0.833	0.837	0.841	0.846
W2V	0.799	0.795	0.661	0.668	0.661	0.666	0.846	0.843	0.846	0.843	0.842	0.849	0.823	0.820
FastText	0.784	0.786	0.572	0.575	0.630	0.634	0.836	0.829	0.833	0.836	0.811	0.807	0.804	0.808
TF-IDF	0.786	0.781	0.763	0.760	0.177	0.181	0.603	0.605	0.859	0.857	0.694	0.697	0.685	0.682
BOW	0.794	0.797	0.733	0.737	0.368	0.364	0.605	0.604	0.863	0.859	0.722	0.726	0.716	0.711

TABLE IV
Performance of HateNet and Baseline Models on the RSN dataset where Acc is Accuracy and M-F1 is Macro-F1 Score

RSN	**LR**		**NB**		**KNN**		**SVM**		**XGB**		**MLP**		**HateNet**	
	Acc	**M-F1**	**Acc**	**M-F1**	**Acc**	**M-F1**	**Acc**	**M-F1**	**Acc**	**M-F1**	**Acc**	**M-F1**	**Acc**	**M-F1**
SBERT	0.782	0.783	0.723	0.718	0.695	0.693	0.814	0.812	0.800	0.803	0.793	0.796	**0.824**	**0.823**
BERT	0.758	0.754	0.587	0.584	0.601	0.602	0.755	0.761	0.747	0.745	0.763	0.764	0.772	0.775
DBERT	0.796	0.793	0.641	0.645	0.683	0.684	0.796	0.799	0.798	0.792	0.802	0.806	0.796	0.798
GloVe	0.774	0.773	0.609	0.601	0.624	0.623	0.806	0.804	0.796	0.801	0.786	0.789	0.794	0.790
W2V	0.752	0.748	0.612	0.615	0.611	0.607	0.798	0.803	0.800	0.794	0.798	0.795	0.775	0.773
FastText	0.731	0.734	0.526	0.523	0.675	0.676	0.784	0.789	0.781	0.782	0.760	0.762	0.753	0.752
TF-IDF	0.734	0.739	0.714	0.712	0.126	0.119	0.552	0.556	0.807	0.802	0.645	0.641	0.632	0.638
BOW	0.745	0.741	0.684	0.686	0.312	0.315	0.557	0.554	0.819	0.810	0.674	0.671	0.663	0.665

processing and the semantic textual similarity benchmark. Further advancements in NLP embeddings can be expected to generally improve current automatic hate speech detection. Of the transformer-based methods, DBERT outperforms BERT. Of the word embedding methods, GloVe and word2vec are similar and outperform fastText as shown in Table II, even though fastText is the only method capable of embedding out-of-vocabulary words.

Our experiments also show that SubDQE is an effective data augmentation method. The **HANS** dataset originally had a 5% Hateful class and a 14% Spam class and after augmentation, is equally distributed at 25% for each class. The **HON** dataset originally had a 5.7% Hateful class, 77.4% Offensive class, and a 16.7% Neutral class and after augmentation, is distributed as 26.4%, 40.3%, and 33.3% respectively. We can observe that in Table V, each model tested had a drastically higher hate recall in the augmented **HON** dataset compared to the imbalanced dataset.

SubDQE outperforms all other baseline data augmentation methods seen in hate speech detection research as demonstrated in Table VI. Of the baseline models, Back Translation and Paraphrasing combined do very well and has been proven to be generally effective in previous literature. Separately, back translation and paraphrasing also led to an increase in accuracy and Macro-F1 score across all datasets. Upsampling and downsampling, some of the simplest data augmentation methods, actually led to lowered accuracies and macro-F1 scores across all three datasets. Of the three data augmentation methods that Rizos et al. [32] proposed, SubAug and PosAug led to a general increase in accuracy and Macro-F1 scores across all datasets but surprisingly, GenAug did not lead to much change in accuracy nor Macro-F1 score across all datasets.

E. Ablation Study

Removed Weighted DropEdge: The HateNet framework unifies several components that contribute to its effectiveness

TABLE V
CONPARISON OF MODELS ON HON WITH AND WITHOUT SUBDQE

Model	Macro-F1	Hate Recall
GloVe+HateNet	0.811	0.18
GloVe+HateNet+SubDQE	0.919	0.87
W2V+HateNet	0.789	0.20
W2V+HateNet+SubDQE	0.926	0.89
TF-IDF+SVM	0.573	0.29
TF-IDF+SVM+SubDQE	0.874	0.86

TABLE VI
COMPARISON OF DATA AUGMENTATION METHODS ON SBERT+HATENET

Method	HON		HANS		RSN	
	Acc	M-F1	Acc	M-F1	Acc	M-F1
No Aug	0.845	0.843	0.877	0.873	0.824	0.823
Up	0.781	0.783	0.812	0.820	0.765	0.761
Down	0.723	0.727	0.751	0.753	0.701	0.703
SubAug	0.856	0.852	0.885	0.887	0.834	0.836
PosAug	0.871	0.873	0.901	0.904	0.856	0.853
SGenAug	0.847	0.841	0.879	0.877	0.826	0.823
BT	0.903	0.906	0.936	0.934	0.881	0.863
Para	0.884	0.886	0.902	0.905	0.863	0.866
BT+Para	0.920	0.926	0.958	0.961	0.903	0.906
SubDQE	**0.946**	**0.948**	**0.971**	**0.973**	**0.920**	**0.926**

TABLE VII
COMPARISON WITH REMOVED DROPEDGE ON SBERT

Model	HON		HANS		RSN	
	Acc	M-F1	Acc	M-F1	Acc	M-F1
HateNet	0.845	0.843	0.877	0.873	0.824	0.823
TextGCN	0.773	0.770	0.799	0.806	0.753	0.756

TABLE VIII
HATENET RESULTS ON DOWNSIZED DATASETS

Model	HON		HANS		RSN	
	Acc	M-F1	Acc	M-F1	Acc	M-F1
SBERT	**0.711**	**0.717**	**0.746**	**0.744**	**0.696**	**0.693**
BERT	0.666	0.664	0.698	0.696	0.624	0.621
DBERT	0.705	0.702	0.733	0.739	0.664	0.661
GloVe	0.681	0.686	0.710	0.717	0.648	0.643
W2V	0.663	0.667	0.694	0.698	0.613	0.612
FastText	0.647	0.644	0.673	0.676	0.593	0.596
TF-IDF	0.502	0.504	0.533	0.536	0.464	0.468
BOW	0.520	0.528	0.556	0.551	0.475	0.472

in hate speech detection. We ran experiments on each dataset using the highest performing embedding method, SBERT, and no augmentation against a similar baseline method, TextGCN [46], that uses a GCN without our Weighted DropEdge. This resulted in a roughly 7% decrease in both accuracy and Macro-F1 score across all three datasets as demonstrated by Table VII, providing insights on how each component of our HateNet framework is indispensible to our learning framework (note that the SubDQE component was already experimented on in the previous subsection).

5,000 Tweets: We also ran experiments on a downsized version of all three datasets and the SBERT+Hatenet combination continued to produce the best accuracy and F1 score (Table VIII). The TF-IDF and BOW had the largest decline in performance across both evaluation metrics. These experiments demonstrate the robustness of the HateNet model on smaller datasets.

Case Study: We observed numerous tweets during the rise in hate crimes in 2017 from the **HON** dataset. Of the 1000 tweets from the **HON** dataset that were tested, the SBERT+HateNet combination misclassified a total of 76 tweets. Of the 76 tweets, 29 tweets came from SubDQE-generated data (38.1%), demonstrating the robustness of the SubDQE process. On analysis of the confusion matrix produced from our experiments, across the board, the most confused class was Hateful and this accounted for the majority of the error (See Table V). We noticed that some created samples do not actually make sense from a semantic point of view because the samples are completely artificially generated.

One limitation to the findings of this work is imposed by the subjectivity of the topic, as different people have different beliefs about what is offensive and what is hateful. The subjectivity is reflected by the high annotator disagreement in the **HON** dataset. Each sample was labelled by at least three people and different coders had differing opinions on what the true label of a sample should be. The fact that even amongst humans there is a relatively high annotator disagreement indicates that by attempting to model samples assuming they are properly labelled might introduce errors due to label noise as a result of lack of an objective definition.

VI. CONCLUSION

The purpose of our paper is to establish an effective hate speech detection framework to flag hateful tweets before escalation to actual hate crimes occur. First, we demonstrate that a hybrid of Dynamic Query Expansion and Substitution Based Augmentation effectively combats class imbalance which is very prevalent in hate speech datasets, and we recommend its use for other applications in social media data mining and natural language processing. Second, we improve on existing hate speech detection research, conducting meticulous experiments with a multitude of embedding method and classifier model combinations, and show similar progress to previous studies. Finally, our proposed Graph Convolutional Network framework HateNet capitalizes on inherent semantic connections between tweets while preventing overfitting and oversmoothing, and our results show that this approach matches or exceeds the performance of traditional classifiers in this domain. Our research represents a step forward for establishing an active anti-hate speech presence in social media, and a step forwards towards a safer internet for everyone.

REFERENCES

[1] B. Auxier and M. Anderson, “Social media use in 2021,” *Pew Research Center*, 2021.

[2] A. Hassan, “Hate-crime violence hits 16-year high, f.b.i. reports,” *The New York Times*, 2019.

[3] E. Vogels, “The state of online harassment,” https://www.pewresearch.org/internet/2021/01/13/the-state-of-online-harassment/, 2021, accessed: 2021-08-19.

[4] Y. Kim, “The painful history of anti-asian hate crimes in america,” *CBS News*, 2021.

[5] Z. Laub, “Hate speech on social media: Global comparisons,” *Council on Foreign Relations*, vol. 7, 2019.

[6] M. Sullaway, “Psychological perspectives on hate crime laws.” *Psychology, Public Policy, and Law*, vol. 10, no. 3, p. 250, 2004.

[7] A. F. Cabrera, A. Nora, P. T. Terenzini, E. Pascarella, and L. S. Hagedorn, “Campus racial climate and the adjustment of students to college: A comparison between white students and african-american students,” *The Journal of Higher Education*, vol. 70, no. 2, pp. 134–160, 1999.

[8] A. M. Schenk and W. J. Fremouw, “Prevalence, psychological impact, and coping of cyberbully victims among college students,” *Journal of school violence*, vol. 11, no. 1, pp. 21–37, 2012.

[9] “Online safety,” https://kidshealth.org/en/teens/internet-safety.html, 2018.

[10] R. Carroll, “Facebook gives way to campaign against hate speech on its pages,” *The Guardian UK*, May 2013.

[11] C. Devine and L. Byington, “Millions are victims of hate crimes, though many never report them,” https://publicintegrity.org/politics/millions-are-victims-of-hate-crimes-though-many-never-report-them/, August 2018, accessed September 1, 2021.

[12] Z. Waseem and D. Hovy, “Hateful symbols or hateful people? predictive features for hate speech detection on twitter,” in *Proceedings of the NAACL student research workshop*, 2016, pp. 88–93.

[13] T. Davidson, D. Warmsley, M. Macy, and I. Weber, “Automated hate speech detection and the problem of offensive language,” in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 11, no. 1, 2017.

[14] A.-M. Founta, C. Djouvas, D. Chatzakou, I. Leontiadis, J. Blackburn, G. Stringhini, A. Vakali, M. Sirivianos, and N. Kourtellis, “Large scale crowdsourcing and characterization of twitter abusive behavior,” in *11th International Conference on Web and Social Media, ICWSM 2018*. AAAI Press, 2018.

[15] A. H. Razavi, D. Inkpen, S. Uritsky, and S. Matwin, “Offensive language detection using multi-level classification,” in *Canadian Conference on Artificial Intelligence*. Springer, 2010, pp. 16–27.

[16] J. Liscombe, J. Venditti, and J. Hirschberg, “Classifying subject ratings of emotional speech using acoustic features,” in *Eighth European Conference on Speech Communication and Technology*, 2003.

[17] J. Qian, M. ElSherief, E. Belding, and W. Y. Wang, “Hierarchical cvae for fine-grained hate speech classification,” *arXiv preprint arXiv:1809.00088*, 2018.

[18] W. Warner and J. Hirschberg, “Detecting hate speech on the world wide web,” in *Proceedings of the second workshop on language in social media*, 2012, pp. 19–26.

[19] L. Natasha, “Facebook, google, twitter commit to hate speech action in germany,” http://techcrunch.com/2015/12/16/ germany-fights-hate-speech-on- social-media/, 2015.

[20] M. Geir, “Zuckerberg in germany: No place for hate speech on facebook,” http://abcnews.go.com/Technology/ wireStory/zuckerberg-place-hate- speech-facebook-37217309, 2016.

[21] J. C. Wong, “Mark zuckerberg tells facebook staff to stop defacing black lives matter slogans,” http://www.theguardian.com/ technology/2016/feb/25/mark- zuckerberg-facebook-defacing-black-lives-matter-signs, 2016.

[22] S. Sood, J. Antin, and E. Churchill, “Profanity use in online communities,” in *Proceedings of the SIGCHI conference on human factors in computing systems*, 2012, pp. 1481–1490.

[23] S. O. Sood, J. Antin, and E. Churchill, “Using crowdsourcing to improve profanity detection,” in *2012 AAAI Spring Symposium Series*, 2012.

[24] R. Cao and R. K.-W. Lee, “Hategan: Adversarial generative-based data augmentation for hate speech detection,” in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020, pp. 6327–6338.

[25] C. Nobata, J. Tetreault, A. Thomas, Y. Mehdad, and Y. Chang, “Abusive language detection in online user content,” in *Proceedings of the 25th international conference on world wide web*, 2016, pp. 145–153.

[26] J. Pavlopoulos, P. Malakasiotis, and I. Androutsopoulos, “Deeper attention to abusive user content moderation,” in *Proceedings of the 2017 conference on empirical methods in natural language processing*, 2017, pp. 1125–1135.

[27] J. Qian, M. ElSherief, E. M. Belding, and W. Y. Wang, “Leveraging intra-user and inter-user representation learning for automated hate speech detection,” *arXiv preprint arXiv:1804.03124*, 2018.

[28] N. Djuric, J. Zhou, R. Morris, M. Grbovic, V. Radosavljevic, and N. Bhamidipati, “Hate speech detection with comment embeddings,” in *Proceedings of the 24th international conference on world wide web*, 2015, pp. 29–30.

[29] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” *arXiv preprint arXiv:1609.02907*, 2016.

[30] P. Mishra, M. Del Tredici, H. Yannakoudakis, and E. Shutova, “Abusive language detection with graph convolutional networks,” *arXiv preprint arXiv:1904.04073*, 2019.

[31] J. Wang, K. Fu, and C.-T. Lu, “Sosnet: A graph convolutional network approach to fine-grained cyberbullying detection,” in *2020 IEEE International Conference on Big Data (Big Data)*. IEEE, 2020, pp. 1699–1708.

[32] G. Rizos, K. Hemker, and B. Schuller, “Augment to prevent: short-text data augmentation in deep learning for hate-speech classification,” in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 991–1000.

[33] M. Guzman-Silverio, Á. Balderas-Paredes, and A. P. López-Monroy, “Transformers and data augmentation for aggressiveness detection in mexican spanish.” in *IberLEF@ SEPLN*, 2020, pp. 293–302.

[34] J. Wei and K. Zou, “Eda: Easy data augmentation techniques for boosting performance on text classification tasks,” *arXiv preprint arXiv:1901.11196*, 2019.

[35] Q. Xie, Z. Dai, E. Hovy, T. Luong, and Q. Le, “Unsupervised data augmentation for consistency training,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 6256–6268, 2020.

[36] D. Jin, Z. Jin, J. T. Zhou, and P. Szolovits, “Is bert really robust? a strong baseline for natural language attack on text classification and entailment,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 05, 2020, pp. 8018–8025.

[37] D. R. Beddiar, M. S. Jahan, and M. Oussalah, “Data expansion using back translation and paraphrasing for hate speech detection,” *Online Social Networks and Media*, vol. 24, p. 100153, 2021.

[38] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *arXiv preprint arXiv:1301.3781*, 2013.

[39] D. K. Hammond, P. Vandergheynst, and R. Gribonval, “Wavelets on graphs via spectral graph theory,” *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129–150, 2011.

[40] R. Rehurek and P. Sojka, “Software framework for topic modelling with large corpora,” in *In Proceedings of the LREC 2010 workshop on new challenges for NLP frameworks*. Citeseer, 2010.

[41] J. Pennington, R. Socher, and C. D. Manning, “Glove: Global vectors for word representation,” in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[42] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov, “Learning word vectors for 157 languages,” *arXiv preprint arXiv:1802.06893*, 2018.

[43] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz *et al.*, “Huggingface’s transformers: State-of-the-art natural language processing,” *arXiv preprint arXiv:1910.03771*, 2019.

[44] N. Reimers and I. Gurevych, “Sentence-bert: Sentence embeddings using siamese bert-networks,” *arXiv preprint arXiv:1908.10084*, 2019.

[45] H. Rosa, N. Pereira, R. Ribeiro, P. C. Ferreira, J. P. Carvalho, S. Oliveira, L. Coheur, P. Paulino, A. V. Simão, and I. Trancoso, “Automatic cyberbullying detection: A systematic review,” *Computers in Human Behavior*, vol. 93, pp. 333–345, 2019.

[46] L. Yao, C. Mao, and Y. Luo, “Graph convolutional networks for text classification,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 7370–7377.