# Towards Explainable Hate Speech Detection

**Happy Khairunnisa Sariyanto[1], Diclehan Ulucan[2], Oguzhan Ulucan[2], Marc Ebner[2]**
Institute of Mathematics and Computer Science, University of Greifswald
17489 Greifswald, Germany
[1]happy.sariyanto@stud.uni-greifswald.de
[2]{diclehan.ulucan, oguzhan.ulucan, marc.ebner}@uni-greifswald.de

## Abstract

Recent advancements in deep learning have significantly enhanced the efficiency and accuracy of natural language processing (NLP) tasks. However, these models often require substantial computational resources, which remains a major drawback. Reducing the complexity of deep learning architectures, and exploring simpler yet effective approaches can lead to cost-efficient NLP solutions. This is also a step towards explainable AI, i.e., uncovering how a particular task is carried out. For this analysis, we chose the task of hate speech detection. We address hate speech detection by introducing a model that employs a weighted sum of valence, arousal, and dominance (VAD) scores for classification. To determine the optimal weights and classification strategies, we analyze hate speech and non-hate speech words based on both their individual and summed VAD-values. Our experimental results demonstrate that this straightforward approach can compete with state-of-the-art neural network methods, including GPT-based models, in detecting hate speech.

## 1 Introduction

Natural language processing is based on concepts and studies in various fields such as computer science, linguistics, and artificial intelligence, and aims to give computer systems the ability to understand, generate, and interpret human language (Chowdhary, 2020). Tasks such as speech recognition, text summarization, and question answering greatly benefit from various NLP algorithms (Kamath et al., 2019; Awasthi et al., 2021; Gupta and Gupta, 2012). In recent years, significant advancements have been made in natural language processing thanks to the developments in the field of deep learning (Khurana et al., 2023). In particular, the introduction of the transformer architecture has led to improvements in text generation, machine translation, and text summarization (Vaswani et al., 2017; Patwardhan et al., 2023). However, despite their accuracy, deep learning models require substantial computational resources and long training times. Although transfer learning can reduce training costs, the high computational demands remain a significant drawback (Yin and Zubiaga, 2021; Sharir et al., 2020).

We can avoid the high cost associated with expensive hardware requirements and time-consuming training procedures if a traditional algorithm can achieve similar or even better results while solving an NLP task. Motivated by this idea, and recognizing the value of exploring simple yet effective approaches alongside proposing new methods to enhance existing techniques, we examine whether an existing concept can be modified to outperform or compete with neural networks-based strategies. For the analysis, we select the text classification task of hate speech detection since numerous complex learning-based methods are present in this field and it holds many challenges reflecting problems encountered in other NLP tasks as well (Zimmerman et al., 2018). One of the great challenges of hate speech detection is the usage of informal language (Reyes et al., 2012). Informal language is more figurative which might be the reason of preferring neural network models over traditional algorithms since neural networks-based models achieve high performance in understanding the meaning of a word within its context. Another challenge is the fact that perceiving content as harmful is a relatively subjective task. Exactly defining hate speech is troublesome (Hietanen and Eddebo, 2023). Several studies classify a text as hate speech if it negatively targets individuals or groups, and hurts or degrades them because of the characteristics of a group they belong to such as sexuality, religion, or nationality, and thus spreads or shows hate towards a certain group (Chiril et al., 2022; Davidson et al., 2017; Cao et al., 2020). Based on the idea that hate speech demonstrates negative

*Findings of the Association for Computational Linguistics: ACL 2025*, pages 12883–12893
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

sentiments, we present a simple learning-free algorithm that uses a sum of valence, arousal, and dominance values (Mohammad, 2018) to classify a given text as hate speech or non-hate speech. According to our experiments, our simple yet effective method can compete with neural networks-based methods, including recent large language models.

The remainder of the paper is organized as follows. In Sec. 2, we give a brief summary of methods utilized in our experiments. In Sec. 3, we present the VAD algorithm, and in Sec. 4, we demonstrate our experimental results. We conclude with a summary and discussion of future directions in Sec. 5 and address the limitations of our study in Sec. 6.

## 2 Related Work

Before neural networks became widely used in the field of hate speech detection, mostly traditional machine learning methods such as Naive Bayes were used (Waseem and Hovy, 2016; Gitari et al., 2015; Chatzakou et al., 2017). As described by Cao et al. (2020), these early approaches typically rely on feature extraction methods such as bag-of-words (Cambria and White, 2014). For instance, the model introduced by Gitari et al. (2015) applies rule-based sentiment analysis techniques to detect hate speech.

In the remainder of this section, we briefly review only the models, and lexicons and datasets which we utilize in our experiments. For more detailed information about hate speech detection, we kindly ask the reader to refer to comprehensive surveys (Yin and Zubiaga, 2021; Fortuna and Nunes, 2018; MacAvaney et al., 2019; Alkomah and Ma, 2022).

### 2.1 Models

The study of Mathew et al. (2021) introduces different methods for the task of hate speech detection, including HateXplain which is based on the BERT model (Devlin et al., 2019). Their work highlights the importance of identifying which text passages contribute to the classification decision, noting that different models often focus on different tokens. The study of Kralj Novak et al. (2022) argues that the assumption of a so-called gold standard, i.e., exactly one correct solution exists, is inadequate as it disregards the aspect of subjectivity and disagreement in hate speech detection. Therefore, a BERT model is trained for English texts on three so-called diamond standard datasets that allow the consideration of disagreement by specifying the percentage of annotators that chose to label the text as hate speech or not. The study of Antypas and Camacho Collados (2023) investigates the generalization of models on unseen data and discusses how dataset formation can inadvertently bias models toward certain types of hate speech, i.e., it points out the tendency to focus on a certain type of hate speech in datasets. To address this, they constructed an ensemble of hate speech datasets, finding that training on such an ensemble improves classification accuracy. The multilingual toolkit called pysentimiento (Pérez et al., 2021) has different models for opinion mining and NLP tasks in social contexts, e.g., alongside hate speech detection there are models for sentiment analysis and irony detection. The models are fine-tuned pre-trained models, and the hate speech detection method uses the pretrained BERTweet model (Nguyen et al., 2020).

### 2.2 Lexicons and Datasets

The NRC-VAD Lexicon (Mohammad, 2018) contains $2000$ words of the English language with their respective valence, arousal and dominance values. These scores, originally in the range $[0, 1]$, are manually assigned by human annotators, with values near $0.5$ indicating neutrality. Values closer to $1$ indicate that there is an intense association with the category the value belongs to, whereas values closer to $0$ represent a low degree of association. Since low associations indicate that the word is more associated with the opposing category, we shift the range of the values from $[0, 1]$ to $[-0.5, 0.5]$. For a word $w$, $f_i(w)$ with $i \in V, A, D$ denotes the corresponding VAD value from the range $[-0.5, 0.5]$.

The Hurtlex Lexicon (Elisa Bassignanaand and Patti, 2018) is a multilingual hate speech lexicon that is based on the Italian hate speech lexicon Le Parole per Ferire (De Mauro, 2016). Hurtlex extends Le Parole per Ferire and contains $8228$ words grouped into three main hate speech categories. These categories are further divided into several subcategories, i.e., the main category called negative stereotypes includes a subcategory for ethnic curse words and another subcategory for degrading terms related to jobs. We use the English lexicon of Hurtlex in our study.

The HatEval2019 dataset (Basile et al., 2019) is formed by using tweets. The tweets are gathered by searching for keywords. These keywords

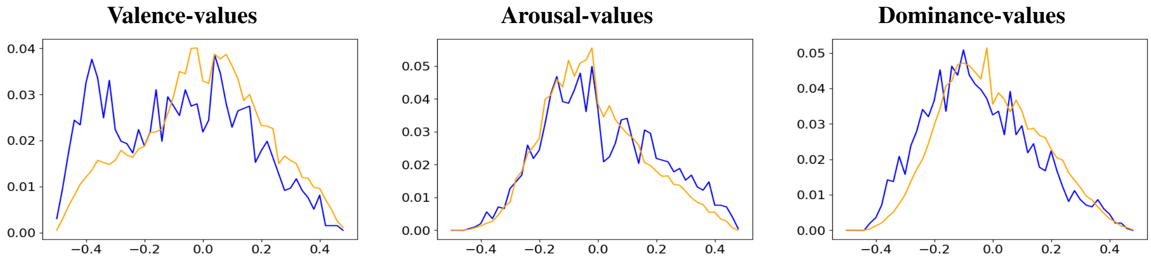

Figure 1: Visualization of the percentage of hate speech words and non-hate speech words based on individual VAD dimensions. The horizontal axis represents the individual value of a word in the given dimension, and the vertical axis presents the percentage of non-hate speech words (orange) or hate speech words (blue). To compute the percentage in the vertical axis, the words in the NRC-VAD Lexicon are sorted according to their $V$, $A$, and $D$ values into bins of length $0.02$ in an interval of $[-0.5, 0.5]$. Then, for each dimension, all words in each bin are classified according to the Hurtlex Lexicon to determine whether they are hate speech or not. To compute the percentage of hate speech words (blue), for each bin, the number of hate speech words in that bin is divided by the total number of all hate speech words. Similarly, for the percentage of non-hate speech words (orange), the number of non-hate speech words in each bin is divided by the total number of all non-hate speech words.

include both neutral words as well as words with high polarity. HatEval2019 is composed of several sets formed for different subtasks. Due to computational constraints, we use only subtask A, which includes $1000$ tweets labeled as hate speech ($427$ tweets, label $1$) or non-hate speech ($573$ tweets, label $0$). It is worth mentioning that due to the formation aim of the dataset, tweets are labeled as hate speech only if they target women or immigrants. Therefore, a hate speech detection model designed to point out any kind of hate speech may classify tweets from this dataset that are labeled as $0$ as hate speech. The dataset by Davidson et al. (2017) also contains tweets, with $24783$ entries labeled as hate speech ($19,790$ tweets), offensive but not hate speech ($3,419$ tweets), and neither offensive nor hate speech ($1,251$ tweets). In our study, we consider the texts classified as offensive but not hate speech also as hate speech since it is pointed out by Davidson et al. (Davidson et al., 2017) that hate speech detection models achieve better results when offensive but non-hate speech texts are considered hate speech as well. The Multilingual and Multi-Aspect Hate Speech dataset (MLMA) (Ousidhoum et al., 2019) is a dataset of tweets with multiple labels. It includes tweets in three languages, namely, Arabic, English and French, with the aim of encouraging analysis of shared hate speech patterns across languages. In our experiments, we used only the English subset of the dataset, which consists of $5,647$ tweets. Each tweet is annotated with five different sets of labels: directness, hostility type, target attribute, target group, and the annotator's sentiment. Tweets are labeled with all applicable tags, thus a single tweet may have more than one label within a category. For the binary classification in our experiments, we are only interested in the label set for the hostility type which consists of the labels *abusive*, *hateful*, *offensive*, *disrespectful*, *fearful* and *normal*. As the directness label is only applied if the hostility type includes a label other than the normal label, we considered the $661$ tweets solely labeled *normal* as non-hate speech and all other $4986$ tweets as hate speech. HateCheck (Röttger et al., 2020) is a synthetic dataset consisting of $3,728$ English texts that aims to facilitate the investigation of model weaknesses by providing functional test suites that target different potential weak points such as detecting when profanities are used in non-hate speech contexts. The dataset has a binary labeling with $2,563$ texts labeled with *hateful* for hate speech and $1,165$ texts labeled with *non-hateful* for non-hate speech. The Dynamically Generated Hate Speech (DynaHate) dataset (Vidgen et al., 2020) is a synthetic dataset created dynamically through four rounds of human-and-model interaction. In the first round, annotators were tasked with creating texts that would challenge the weaknesses of a RoBERTa model (Liu et al., 2019), using direct feedback from the model during the process. In the subsequent rounds, texts were modified to preserve their structure as much as possible while flipping the label from hate speech to non-hate speech. DynaHate consists of $41,144$ texts, of which $32,924$ texts are labeled as the training set. $4,100$ texts of the data is assigned to the development set and $4,120$ texts to the test set. For our experiments, we used only the test set, which contains $2,268$ texts labeled as *hate* (hate speech) and $1,852$ texts labeled as *nothate* (non-hate speech).

## 3 Proposed Approach

Our approach is based on the VAD model which is also widely used in psychology to measure emotions (Mohammad, 2018). The VAD model represents emotions in a three-dimensional space. The first dimension, valence, quantifies the positivity or negativity of an emotion. The second dimension, arousal, measures the activeness or passiveness of the emotion. The third dimension, dominance, quantifies the degree of control one feels over a situation. For instance, calmness can be characterized by high valence, low arousal, and high dominance, whereas anger is typically associated with low valence, high arousal, and high dominance.

In the remainder of this section, we first analyze the VAD values of commonly used words in hate speech, and then we introduce our method, called VAD-Baseline.

### 3.1 Analysis of VAD values of Common Words in Hate Speech

In our method, we use the NRC-VAD Lexicon (Mohammad, 2018; Vishnubhotla and Mohammad, 2022). To classify the words as hate speech or not, we first group the words in the NRC-VAD Lexicon according to their frequency of usage in hate speech, while we consider a word as hate speech if it is included in the Hurtlex Lexicon (Elisa Bassignanaand and Patti, 2018). Figure 1 shows an analysis of the VAD values for the two groups, with each dimension examined separately. As it can be observed, the distribution of both groups is very similar, particularly in terms of arousal and dominance values. The difference is more noticeable for the valence values which might be a result of the fact that hate speech words are usually associated with negative emotions, while non-hate speech words are more likely to be neutral or positive (Mohammad, 2018; Chiril et al., 2022; Davidson et al., 2017; Cao et al., 2020). Also, in Fig. 1, it is observable that the valence and dominance values of hate speech words are usually more negative than those of non-hate speech words, and the arousal value of hate speech words is more likely to be positive than that of non-hate speech words.

Afterwards, we analyze the weighted sums of the valence-, arousal- and dominance-values by assigning weights of either $1$ or $-1$ to each dimension. We apply possible combinations of weights to increase the differences between the groups. As the distributions of the individual dimensions of the groups are similar, it is not surprising that the distributions of the summed values are similar as well. We obtain the most noticeable differences for the weight combinations of $[1, -1, 1]$ and $[-1, 1, -1]$ for valence, arousal, and dominance, respectively. While for the former combination, the summed values of hate speech words tend to be more negative than those of non-hate speech words, for the latter they tend to be more positive, which also coincides with our observations on the individual dimensions of each group.

Since it is more intuitive to associate hate speech with negativity, our initial combination of weights, i.e., $[1, -1, 1]$, is modified as described below, so that VAD values indicating a higher ratio of hate speech words have a greater impact on the weighted sum. To derive these new modified weights, we pass the associated VAD value of a word $w$ through a unit step function as follows:

$$g_i(w) = \begin{cases} 0.6, & \text{if } f_V(w) < \Theta_V \text{ and } i = V, \\ -1.6, & \text{if } f_A(w) > \Theta_A \text{ and } i = A, \\ 1, & \text{if } f_D(w) < \Theta_D \text{ and } i = D, \\ \theta_i, & \text{otherwise,} \end{cases} \tag{1}$$

where $i \in \{V, A, D\}$, and $f_i$ and $g_i$ denote the VAD-value and the result of the gate function, respectively. The parameter $\Theta_i$ represents the threshold value used to identify hate speech, while $\theta_i$ denotes the weight assigned when a non-hate speech word is processed.

Figure 2 illustrates the corresponding step functions. Specifically, high $V$ and $D$ values (beyond certain thresholds) are suppressed by assigning them weights below 1, while high arousal values receive a negative weight. The procedure for selecting the cut-off thresholds is detailed in Sec. 4.3. The resulting summed word-level VAD values are shown in Fig. 3.

### 3.2 A Word-Level Approach based on Joy and Anger

We also explore a weighting strategy inspired by Bălan et al. (2020), who demonstrated that the most common base emotions in hate speech are joy and anger. As suggested, we multiply the respective individual VAD values of joy ($j$) and anger ($a$)

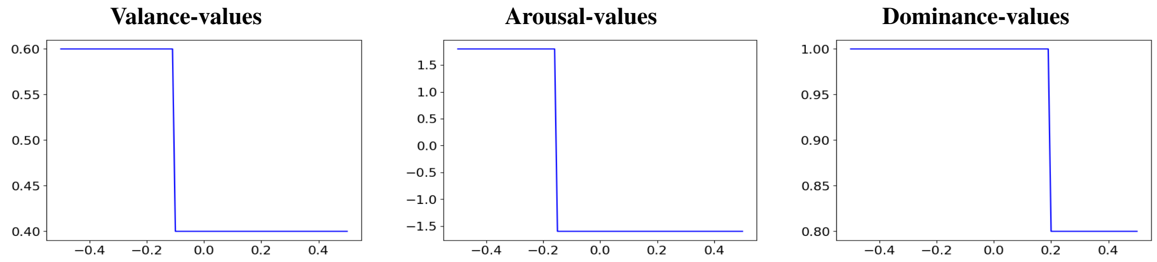

Figure 2: Word-level weight functions for different values.

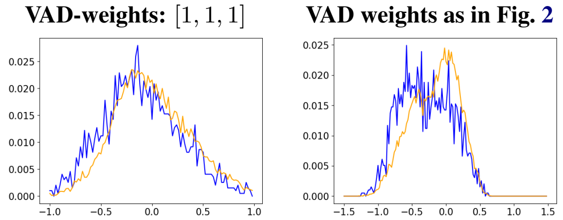

Figure 3: Visualization of the percentage of hate speech words and non-hate speech words based on the summed VAD values with different word-level weights. While the plot on the left-hand side shows an unweighted summation, the right-hand side represents the change in the distribution of hate speech/non-hate speech words when the step functions in Fig. 2 are used. The horizontal axis represents the summed VAD value of a word, i.e., $f_V(w) + f_A(w) + f_D(w)$, while the vertical axis presents the percentage of non-hate speech words (orange) or hate speech words (blue) similar to Fig. 1.

to obtain a new vector $F$ of individual VAD values as follows:

$$\mathbf{F} = [f_{j_V} \cdot f_{a_V},\ \ f_{j_A} \cdot f_{a_A},\ \ f_{j_D} \cdot f_{a_D}]. \tag{2}$$

Then, we use the step functions given in Fig. 2 to weight the components of $\mathbf{F}$, and sum these new VAD-values as follows:

$$\begin{aligned}\beta = g_V(f_{j_V} \cdot f_{a_V}) \cdot (f_{j_V} \cdot f_{a_V})\\ + g_A(f_{j_A} \cdot f_{a_A}) \cdot (f_{j_A} \cdot f_{a_A})\\ + g_D(f_{j_D} \cdot f_{a_D}) \cdot (f_{j_D} \cdot f_{a_D}).\end{aligned} \tag{3}$$

As a result, we obtain a value $\beta$ which is closer to the mean of the summed word-level VAD value of hate speech words compared to the mean of the summed word-level VAD value of non-hate speech words.

### 3.3 Formation of the VAD-Baseline

To form our VAD-Baseline, we introduce two different strategies on the summation of the VAD values of each word in a given text. We have two stages to compute the summed VAD value of a given text. Firstly, we obtain the accumulated VAD value of each word in the corpus by summing the VAD components per word. Secondly, we sum these values for the entire text.

In order to determine the most effective weighting strategy, we utilize different weight combinations on the word-level and text-level. We use the weights that we determined in Sec. 3.1 for the word-level weights of each individual VAD value. On the other hand, to determine the text-level weights, we carry out practical experiments, which are explained in Sec. 4.3. As we demonstrate in Fig. 4, we have 5 different functions $\hat{g}$ that weight the texts in a different manner. Our first function $\hat{g}_0$ does not affect any value in the summation on text-level, thus $sum_{\hat{g}_0}$ only applies word-level weighting. We apply this function to have a simple starting point that weights all emotional dimensions equally. Our second function $\hat{g}_1$ assigns higher weights to values in the range $R_1 = [-0.82, -0.72]$ and suppresses other values. In other words, through $\hat{g}_1$, we increase the weight if the word-level value is in a range where the rate of hate speech words is higher than that of non-hate speech words. In our third function $\hat{g}_2$, we suppress word-level values in the range $R_2 = [-0.28, 0.5]$ where they are close to zero. In this range, the distributions for hate speech words and non-hate speech words are very similar or the rate of non-hate speech words is higher than that of hate-speech words. Our fourth function $\hat{g}_3$ combines the strategies of $\hat{g}_1$ and $\hat{g}_2$ by applying weights as in $\hat{g}_2$ for values in $R_1$ and using $\hat{g}_1$ for all other values. Our fifth function $\hat{g}_4$ is related to the anger and joy weighting approach. We use the difference between the summed word-level values (Fig. 3) and the summed word level VAD value of anger and joy (Eqn. 3). $\hat{g}_4$ assigns weights by suppressing values whose difference with the summed word level VAD value of anger and joy is greater than $0.03$, which is approximately the

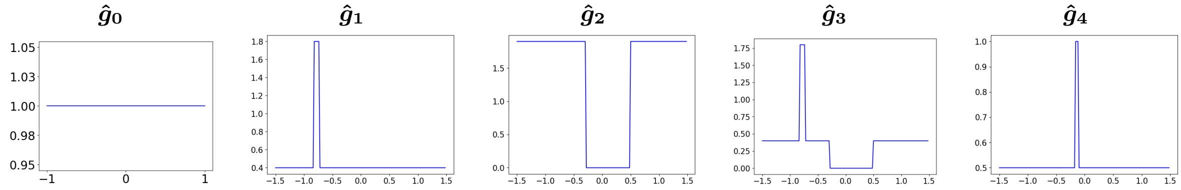

Figure 4: Text-level weights. We consider 5 distinct functions which weight the texts in a different manner.

difference between the summed word-level VAD values of anger and joy and the average of hate speech words.

We combine the text-level weight functions with the word-level weights as follows. For a given text, we first determine its words that have a VAD value. Then, for each of these words, we compute the word-level summation $w_{sum}$ by using our word-level weights shown in Fig. 2 as follows:

$$w_{sum}(w) = g_V(w) \cdot f_V(w) + g_A(w) \cdot f_A(w) + g_D(w) \cdot f_D(w). \quad (4)$$

Lastly, we apply a text-level weight function to the word-level summation to obtain the text-level summation $t_{sum}$ as follows:

$$t_{sum}(text) = \frac{1}{K} \sum_{k}^{K} \hat{g}(w_{sum}(w_k)) \cdot w_{sum}(w_k), \quad (5)$$

where $text$ is the given input text, and $K$ is the number of words having a VAD value in the given text with $k = 1, 2, 3, ..., K$.

After we compute the summed VAD values at the text-level, we apply two different classification operators, $class$, to decide whether a given text should be classified as hate speech. The first approach, $class_1$, categorizes a text as hate speech if its summed VAD value is negative, whereas the second approach, $class_2$, classifies a text as hate speech if the difference of its summed text-level VAD value to the summed word-level VAD value of anger and joy, $\beta$, is smaller than a certain threshold (practically determined as 0.1):

$$\text{class}_1(\text{text}) = \begin{cases} 1, & \text{if } t_{\text{sum}}(\text{text}) < 0, \\ 0, & \text{otherwise}, \end{cases}$$
$$\text{class}_2(\text{text}) = \begin{cases} 1, & \text{if } t_{\text{sum}}(\text{text}) - \beta < 0.1, \\ 0, & \text{otherwise}. \end{cases} \quad (6)$$

## 4 Experiments

In this section, we first present our experimental setup. Then, we provide our experimental results and discuss the results. Lastly, we detail our parameter selection procedure.

### 4.1 Experimental Setup

We briefly introduce the evaluation and tokenization strategies utilized in our experiments. To analyze the performance of our VAD-Baseline method, we use three metrics, namely, precision, recall, and F1-score (Jardine and van Rijsbergen, 1971; Van Rijsbergen, 1974). While precision provides us the score for the correctly predicted positive labels among all the positive values, recall gives information about the correctly classified actual positives, and F1-score demonstrates the harmonic mean of recall and precision.

The parameters we utilize in our method are extracted from the validation set of the HatEval2019 (Basile et al., 2019) dataset. After obtaining these parameters, we conduct comprehensive experiments to evaluate the method and assess its generalizability. These experiments are carried out on 5 datasets, namely, the evaluation set of HatEval2019 (Basile et al., 2019), Davidson (Davidson et al., 2017), HateCheck (Röttger et al., 2020), DynaHate (Vidgen et al., 2020), and MLMA (Ousidhoum et al., 2019). To tokenize the texts in these datasets, we cannot apply a simple method such as tokenizing the words based on spacing, since in some tweets spacing between words is ignored. Therefore, we utilize the tokenizer from the roBERTa model (Antypas and Camacho Collados, 2023; Zhuang et al., 2021), and apply a post-processing step to discard special tokens (e.g., end-of-string markers or symbols indicating preceding spaces). Furthermore, we retain only those tokens corresponding to words present in the NRC-VAD Lexicon (Mohammad, 2018)

### 4.2 Experimental Results and Discussion

We evaluate the performance of our VAD-Baseline method against the pysentimiento (Pérez et al., 2021), roBERTa Hate Speech (Antypas and Camacho Collados, 2023; Camacho-Collados et al., 2012), Hate Speech EN (Kralj Novak et al., 2022), and HateXplain (Mathew et al., 2021) models. These methods (described in Sec.2.1) are not fine-tuned during our experiments. Moreover, we investigate the performance of current GPT models (OpenAI, 2024), i.e., gpt3.5-turbo, gpt4o-mini, and gpt4o, on both benchmarks.

We provide the results in Table 1 where we also analyze the effects of different weight combinations and classification strategies on our VAD-Baseline method. Overall, our simple yet effective approach competes with neural network-based methods and GPT models, and in some cases, even outperforms them. It is important to stress that we do not aim to design an algorithm that outperforms the state-of-the-art but to show that with simple modifications existing methods can compete and even outperform them. This is a step towards explainable AI. The goal is to uncover the underlying computations. Thus, it is desirable to investigate and enhance existing techniques to obtain accurate results with low-cost methods.

As already noted by Antypas and Camacho Collados (2023), neural network-based models face challenges on unseen data, i.e., models might not be generalizable, and without fine-tuning they do not necessarily generalize better than traditional algorithms which can also be observed in Table 1. Our VAD-Baseline method performs more accurately than several of the neural networks-based models in terms of different metrics. For instance, on the HatEval2019 Evaluation Set, the combination of $sum_{\hat{g}_2}$ and $class_2$, achieves the best performance in terms of precision among all methods. Also, on both the HatEval2019 Evaluation and Davidson datasets, the combination of $sum_{\hat{g}_2}$ and $class_1$, and $sum_{\hat{g}_3}$ and $class_1$ output the second-best scores in terms of recall, while they present the best F1-score on the Davidson dataset. Additionally, most of the combinations with $class_1$ are among the $5$ best-performing methods in terms of recall and F1-score. Particularly, the combinations that outperform the neural networks-based methods and even large language models might provide new perspectives on how to improve these existing strategies with simple and explainable modifications. Furthermore, the combinations of different weights and classes provide consistent results across the datasets which demonstrates the generalizablity of our method.

As we can see, through different weights and classification strategies we obtain distinct statistical results. Although one could select the best combination for each dataset individually, an adaptive selection of weights, classification strategies, and threshold ranges may further enhance the efficiency of our VAD-Baseline algorithm (see Sec. 6). Nonetheless, the current experiments show that a simple yet effective approach can indeed be useful for this task. Hence, the improvement of existing traditional methods can benefit the field of natural language processing, and the observations we make while enhancing the classical algorithms can also provide us new perspectives for the improvement of neural networks-based algorithms including large language models.

As a final note, it is worth mentioning that since our approach depends on lexicons that list particular hate and non-hate speech terms, it may have difficulty detecting implicit hate speech that requires contextual interpretation. Thus, while our approach provides an explainable method, modifying it to capture implicit hate speech is a valuable future direction. For instance, one can create a hybrid model that uses the weights we proposed, and neural networks to capture contextual information. This might offer a straightforward path to improving the effectiveness of our method.

### 4.3 Parameter Selection

We would like to briefly mention how we determined the most effective weights, ranges, and thresholds. We carried out comprehensive grid search experiments on the HatEval2019 Validation Set (Basile et al., 2019), where we analyzed different combinations of parameters. First of all, we examined the impact of different cut-off thresholds on the word-level weight functions that are mentioned in Sec. 3.1. We tested thresholds in a range of $[-0.3, 0.3]$ with a step size of $0.05$, and observed that the best results are obtained when the cut-off thresholds shown in Fig. 2 are used. Furthermore, we investigated different ranges, i.e., $R_1$ and $R_2$, to determine the text-level weight functions given in Fig. 4. We first partitioned the interval $[-1.5, 1.5]$ into smaller segments of length $0.02$. For each segment, we identified those where hate speech words outnumbered non-hate speech

| | | HatEval2019 Val. Set | | | HatEval2019 Eval. Set | | | Davidson | | | HateCheck | | | DynaHate | | | MLMA | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Algorithms** | **precision** | **recall** | **F1-score** | **precision** | **recall** | **F1-score** | **precision** | **recall** | **F1-score** | **precision** | **recall** | **F1-score** | **precision** | **recall** | **F1-score** | **precision** | **recall** | **F1-score** |
| **Learning-based** | gpt3.5-turbo | 0.549 | 0.983 | 0.705 | 0.784 | 0.983 | 0.872 | 0.998 | 0.064 | 0.119 | 0.809 | 0.999 | 0.894 | 0.639 | 0.991 | 0.778 | 0.895 | 0.995 | 0.942 |
| | gpt4o-mini | 0.689 | 0.702 | 0.695 | 0.781 | 0.536 | 0.635 | 0.994 | 0.667 | 0.798 | 0.900 | 0.998 | 0.946 | 0.833 | 0.897 | 0.864 | 0.909 | 0.895 | 0.902 |
| | gpt4o | 0.663 | 0.750 | 0.704 | 0.785 | 0.499 | 0.609 | 0.995 | 0.496 | 0.662 | 0.889 | 0.997 | 0.941 | 0.844 | 0.911 | 0.876 | 0.914 | 0.836 | 0.873 |
| | pysentimiento | 0.710 | 0.850 | 0.774 | 0.810 | 0.834 | 0.821 | 0.986 | 0.605 | 0.749 | 0.820 | 0.388 | 0.527 | 0.710 | 0.327 | 0.448 | 0.897 | 0.110 | 0.196 |
| | roBERTa Hate Speech | 0.829 | 0.853 | 0.841 | 0.809 | 0.549 | 0.654 | 0.991 | 0.701 | 0.821 | 0.907 | 0.618 | 0.735 | 0.771 | 0.431 | 0.553 | 0.886 | 0.997 | 0.938 |
| | Hate Speech EN | 0.569 | 0.658 | 0.610 | 0.768 | 0.746 | 0.757 | 0.797 | 0.943 | 0.864 | 0.772 | 0.870 | 0.818 | 0.590 | 0.664 | 0.624 | 0.901 | 0.631 | 0.742 |
| | HateXplain | 0.600 | 0.007 | 0.014 | 0.750 | 0.371 | 0.499 | 0.972 | 0.589 | 0.733 | 0.730 | 0.165 | 0.270 | 0.682 | 0.284 | 0.401 | 0.901 | 0.740 | 0.812 |
| **VAD-Baseline** | $sum_{\hat{g}_0}$ and $class_2$ | 0.419 | 0.468 | 0.441 | 0.780 | 0.891 | 0.831 | 0.800 | 0.315 | 0.452 | 0.679 | 0.513 | 0.585 | 0.533 | 0.641 | 0.581 | 0.226 | 0.532 | 0.317 |
| | $sum_{\hat{g}_0}$ and $class_1$ | 0.448 | 0.857 | 0.588 | 0.778 | 0.838 | 0.807 | 0.871 | 0.849 | 0.860 | 0.671 | 0.572 | 0.618 | 0.524 | 0.675 | 0.590 | 0.228 | 0.732 | 0.348 |
| | $sum_{\hat{g}_1}$ and $class_2$ | 0.457 | 0.817 | 0.586 | 0.780 | 0.891 | 0.831 | 0.760 | 0.839 | 0.798 | 0.658 | 0.661 | 0.660 | 0.548 | 0.798 | 0.650 | 0.875 | 0.684 | 0.768 |
| | $sum_{\hat{g}_1}$ and $class_1$ | 0.446 | 0.876 | 0.591 | 0.779 | 0.853 | 0.814 | 0.848 | 0.874 | 0.861 | 0.668 | 0.579 | 0.620 | 0.519 | 0.688 | 0.592 | 0.885 | 0.735 | 0.804 |
| | $sum_{\hat{g}_2}$ and $class_2$ | 0.351 | 0.276 | 0.309 | 0.812 | 0.318 | 0.457 | 0.671 | 0.051 | 0.095 | 0.680 | 0.464 | 0.552 | 0.579 | 0.445 | 0.503 | 0.877 | 0.319 | 0.468 |
| | $sum_{\hat{g}_2}$ and $class_1$ | 0.446 | 0.958 | 0.609 | 0.785 | 0.942 | 0.856 | 0.843 | 0.901 | 0.871 | 0.671 | 0.654 | 0.662 | 0.519 | 0.789 | 0.626 | 0.883 | 0.764 | 0.819 |
| | $sum_{\hat{g}_3}$ and $class_2$ | 0.475 | 0.700 | 0.566 | 0.811 | 0.322 | 0.461 | 0.839 | 0.633 | 0.722 | 0.681 | 0.468 | 0.555 | 0.579 | 0.449 | 0.506 | 0.877 | 0.321 | 0.470 |
| | $sum_{\hat{g}_3}$ and $class_1$ | 0.446 | 0.958 | 0.609 | 0.785 | 0.941 | 0.856 | 0.842 | 0.901 | 0.871 | 0.671 | 0.654 | 0.662 | 0.519 | 0.789 | 0.626 | 0.883 | 0.764 | 0.819 |
| | $sum_{\hat{g}_4}$ and $class_2$ | 0.454 | 0.541 | 0.494 | 0.776 | 0.805 | 0.791 | 0.809 | 0.121 | 0.211 | 0.656 | 0.512 | 0.575 | 0.524 | 0.648 | 0.580 | 0.886 | 0.693 | 0.778 |
| | $sum_{\hat{g}_4}$ and $class_1$ | 0.456 | 0.649 | 0.535 | 0.768 | 0.624 | 0.688 | 0.858 | 0.750 | 0.800 | 0.647 | 0.406 | 0.499 | 0.525 | 0.464 | 0.493 | 0.888 | 0.612 | 0.724 |

Table 1: Comparison of the performance of the variants of VAD-Baseline with learning-based strategies. The five best results are highlighted using color coding as follows, first: purple, second: blue, third: green, forth: yellow, and fifth: orange.

words ($R_1$) and those where non-hate speech words predominated ($R_2$). Then, we examined all possible unions of these segments to find the shortest continuous ranges where at least $90\%$ of the bins within the given range include a higher rate of hate speech words for $R_1$ and at least $80\%$ of the bins within the given range include a higher rate of non-hate speech words for $R_2$. Consequently, we determined $R_1 = [-0.82, -0.72]$ and $R_2 = [-0.28, 0.5]$. Lastly, we analyzed the impact of different weight values at both the word and text levels. For word-level weights, we tested V- and D-values within the range $[0.2, 2]$. For A-values, we applied different ranges based on the threshold: values below the threshold were tested in the range $[0.2, 2]$, while values above the threshold were tested in the range $[-2, -0.2]$. In both cases, we used a step size of $0.2$ across all considered threshold values. The best results were achieved using a specific combination of weights and thresholds, as illustrated in Fig. 2 and Eqn. 1. For text-level weights, we investigated $\hat{g}_1$ by testing values in the range $[1, 2]$ with a step size of $0.1$ for values within $R_1$ and values in the range $[0.1, 1]$ with the same step size for values outside $R_1$. Similarly, for $\hat{g}_2$, weights were explored within $[0.1, 1]$ for values in $R_2$ and within $[1, 2]$ for values outside $R_2$, both with a step size of $0.1$. For $\hat{g}_4$, weights in the range $[1, 2]$ with a step size of $0.1$ were tested for cases where the difference between the summed word-level VAD values of anger and joy was below $0.03$. Otherwise, weights in the range $[0.1, 1]$ with the same step size were used. Based on these experiments, the optimal text-level weights are presented in Fig. 4. It is worth mentioning that as we present in Table 1, overall, the parameters generalize well on the HatEval2019 Evaluation Set, Davidson, HateCheck, DynaHate, and MLMA datasets.

## 5 Conclusion

Natural language processing has greatly benefited from learning-based approaches, particularly transformer models, which have enabled highly accurate performance in tasks such as question answering and text summarization. However, these come with high computational costs, mainly due to their training processes, which often require expensive hardware and a significant amount of time. On the other hand, traditional algorithms do not require any training phase, making them generally more computationally efficient than neural networks-based models. Motivated by this fact and taking into account that it is important to explore explainable simple yet effective approaches to improve existing techniques alongside developing new methods, in this paper, we investigate whether low-cost traditional algorithms can compete or even outperform neural networks-based models in the field of natural language processing. We chose hate speech detection for our investigation because it reflects challenges common to many NLP tasks, including informal language use and subjectivity, and has been the focus of numerous complex learning-based models. Since hate speech is characterized by negative sentiments, we propose a learning-free method that leverages valence, arousal, and dominance values to classify texts as hate speech or not. In the experiments, our VAD-Baseline algorithm demonstrates that simple yet effective methods can compete with recent neural networks-based models. Hence, we argue that further analysis of low-cost explainable methods can help us to obtain robust algorithms while avoiding the computational burden of extensive training. Furthermore, neural networks-based models can benefit from the observations we obtain from improving classical methods.

As future work, we plan to refine the VAD-Baseline algorithm by making the selection of weights and classification strategies adaptive, as well as enhancing it to address implicit hate speech. Also, we will investigate different natural language processing tasks and existing traditional methods to analyze whether we can obtain accurate results without high computational costs.

## 6 Limitations

During our investigation, we observed that our method may sometimes label non-hate speech texts as hate speech. The reason for this could be that increasing the weights for words with negative values not only increases the impact of hate speech terms but also causes words with negative polarity in general to have larger weights. Thus, non-hate speech texts that convey a negative sentiment such as sadness or frustration can be falsely classified as hate speech. Also, for both word-level and text-level summation there are various ways to obtain the same value through summation which may be a reason for the similar distributions of the summed VAD-values for hate speech words and non-hate speech words. For instance, on word-level, there may be non-hate speech words that do not have similar individual VAD-values as hate speech words, yet have a negative summed VAD-value.

As shown in Table 1, different combinations of weights and classification methods result in distinct outcomes. Selecting the weight combinations and classification approaches adaptively would enhance the effectiveness of our algorithm which we consider as future work.

Hitherto, our method was evaluated only on English texts because it relies on the NRC-VAD Lexicon. Adapting the approach for other languages may require modifications to use different VAD lexicons. Furthermore, even for English texts it is important to keep the lexicons up-to-date since informal language is dynamic and evolves over time. Thus, hate speech patterns, styles, and frequently used words also change. As future work, we will explore the usability of our method in another language by utilizing a lexicon in that language.

Also, since our method relies on lexicons, including specific hate speech and non-hate speech words, it may struggle with implicit hate speech messages that require contextual understanding. To address this issue, future work could explore including additional linguistic or contextual cues beyond lexicon-based features. Similarly, while our method provides an interpretable framework, further research could enhance its ability to capture implicit hate speech through external knowledge sources or contextual embeddings.

Additionally, our method is not robust against intentionally obscured text, such as deliberate typos or other changes in syntax, that is a widely known challenge in this field for both traditional and learning-based models (Gitari et al., 2015; Gröndahl et al., 2018). The VAD-Baseline could be extended with additional lexicons to address these issues, as well as challenges posed by the use of irony and emojis.

## Ethics Statement

In this study, we do not use private data or non-public information. We rely exclusively on publicly available datasets and models, each released under specific licenses. Hurtlex is distributed under the CC BY-NC-SA 4.0 license, while HatEval2019 and HateCheck are available under CC BY-NC 4.0. The Davidson dataset, Hate Speech EN, MLMA and the TweetNLP package are provided under the MIT License. HateXplain is released under the Apache 2.0 license. Additionally, we use NRC-VAD and DynaHate, which do not have an explicitly stated license but permit usage in research when the associated papers are cited. pysentimiento is an open-source library available for non-commercial and scientific research purposes, with models trained on third-party datasets that are subject to their respective licenses.

Note that as the topic is hate speech detection, offensive content is part of the research question and therefore a part of the datasets. In the HatEval2019, Davidson and MLMA datasets user information is either obscured or removed entirely, while the DynaHate und HateCheck datasets are synthetically formed and do not contain any user information.

## References

Fatimah Alkomah and Xiaogang Ma. 2022. A literature review of textual hate speech detection methods and datasets. *Information*, 13(6):273.

Dimosthenis Antypas and Jose Camacho Collados. 2023. Robust hate speech detection in social media: A cross-dataset empirical evaluation. In *The 7th Workshop on Online Abuse and Harms*, pages 231–242.

Ishitva Awasthi, Kuntal Gupta, Prabjot Singh Bhogal, Sahejpreet Singh Anand, and Piyush Kumar Soni. 2021. Natural language processing (nlp) based text summarization-a survey. In *6th International Conference on Inventive Computation Technologies*, pages 1310–1317.

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. 2019. Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 54–63.

Oana Bălan, Gabriela Moise, Livia Petrescu, Alin Moldoveanu, Marius Leordeanu, and Florica Moldoveanu. 2020. Emotion classification based on biophysical signals and machine learning techniques. *Symmetry*, 12(1).

Jose Camacho-Collados, Kiamehr Rezaee, Talayeh Riahi, Asahi Ushio, Daniel Loureiro, Dimosthenis Antypas, Joanne Boisson, Luis Espinosa-Anke, Fangyu Liu, Eugenio Martínez-Cámara, et al. 2012. TweetNLP: Cutting-Edge Natural Language Processing for Social Media. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*.

Erik Cambria and Bebo White. 2014. Jumping nlp curves: A review of natural language processing research. *IEEE Computational Intelligence Magazine*, 9(2):48–57.

Rui Cao, Roy Ka-Wei Lee, and Tuan-Anh Hoang. 2020. Deephate: Hate speech detection via multi-faceted text representations. In *Proceedings of the 12th ACM Conference on Web Science*, pages 11–20.

Despoina Chatzakou, Nicolas Kourtellis, Jeremy Blackburn, Emiliano De Cristofaro, Gianluca Stringhini, and Athena Vakali. 2017. Mean birds: Detecting aggression and bullying on twitter. In *Proceedings of the 2017 ACM on Web Science Conference*, pages 13–22.

Patricia Chiril, Endang Wahyu Pamungkas, Farah Benamara, Véronique Moriceau, and Viviana Patti. 2022. Emotionally informed hate speech detection: a multi-target perspective. *Cognitive Computation*, pages 1–31.

KR Chowdhary. 2020. Natural language processing. *Fundamentals of Artificial Intelligence*, pages 603–649.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the International AAAI Conference on Web and Social Media*, pages 512–515.

Tullio De Mauro. 2016. Le parole per ferire. *Internazionale*, 27(9):2016.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *North American Chapter of the Association for Computational Linguistics*, pages 4171–4186.

Valerio Basile Elisa Bassignanaand and Viviana Patti. 2018. Hurtlex: A multilingual lexicon of words to hurt. In *CEUR Workshop Proceedings*, pages 1–6.

Paula Fortuna and Sérgio Nunes. 2018. A survey on automatic detection of hate speech in text. *ACM Computing Surveys*, 51(4):1–30.

Njagi Dennis Gitari, Zhang Zuping, Hanyurwimfura Damien, and Jun Long. 2015. A lexicon-based approach for hate speech detection. *International Journal of Multimedia and Ubiquitous Engineering*, 10(4):215–230.

Tommi Gröndahl, Luca Pajola, Mika Juuti, Mauro Conti, and N. Asokan. 2018. All You Need is "Love": Evading Hate Speech Detection. In *Proceedings of the 11th ACM Workshop on Artificial Intelligence and Security*, page 2–12.

Poonam Gupta and Vishal Gupta. 2012. A survey of text question answering techniques. *International Journal of Computer Applications*, 53(4).

Mika Hietanen and Johan Eddebo. 2023. Towards a definition of hate speech—with a focus on online contexts. *Journal of Communication Inquiry*, 47(4):440–458.

Nick Jardine and Cornelis Joost van Rijsbergen. 1971. The use of hierarchic clustering in information retrieval. *Information Storage and Retrieval*, 7(5):217–240.

Uday Kamath, John Liu, and James Whitaker. 2019. *Deep learning for NLP and speech recognition*, volume 84. Springer.

Diksha Khurana, Aditya Koli, Kiran Khatter, and Sukhdev Singh. 2023. Natural language processing: State of the art, current trends and challenges. *Multimedia Tools and Applications*, 82(3):3713–3744.

Petra Kralj Novak, Teresa Scantamburlo, Andraž Pelicon, Matteo Cinelli, Igor Mozetič, and Fabiana Zollo. 2022. *Handling Disagreement in Hate Speech Modelling*. Springer.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Sean MacAvaney, Hao-Ren Yao, Eugene Yang, Katina Russell, Nazli Goharian, and Ophir Frieder. 2019. Hate speech detection: Challenges and solutions. *PloS One*, 14(8):e0221152.

Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. 2021. Hatexplain: A benchmark dataset for explainable hate speech detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(17):14867–14875.

Saif Mohammad. 2018. Obtaining reliable human ratings of valence, arousal, and dominance for 20,000 english words. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 174–184.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. BERTweet: A pre-trained language model for English Tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14.

OpenAI. 2024. OpenAI platform: Models. `https://platform.openai.com/docs/models/gp`, (Last accessed: 09.02.2025).

Nedjma Ousidhoum, Zizheng Lin, Hongming Zhang, Yangqiu Song, and Dit-Yan Yeung. 2019. Multilingual and multi-aspect hate speech analysis. *arXiv preprint arXiv:1908.11049*.

Narendra Patwardhan, Stefano Marrone, and Carlo Sansone. 2023. Transformers in the real world: A survey on nlp applications. *Information*, 14(4):242.

Juan Manuel Pérez, Mariela Rajngewerc, Juan Carlos Giudici, Damián A. Furman, Franco Luque, Laura Alonso Alemany, and María Vanina Martínez. 2021. pysentimiento: a python toolkit for opinion mining and social nlp tasks. *arXiv*, arXiv:2106.09462.

Antonio Reyes, Paolo Rosso, and Davide Buscaldi. 2012. From humor recognition to irony detection: The figurative language of social media. *Data & Knowledge Engineering*, 74:1–12. Applications of Natural Language to Information Systems.

Paul Röttger, Bertram Vidgen, Dong Nguyen, Zeerak Waseem, Helen Margetts, and Janet B Pierrehumbert. 2020. Hatecheck: Functional tests for hate speech detection models. *arXiv preprint arXiv:2012.15606*.

Or Sharir, Barak Peleg, and Yoav Shoham. 2020. The cost of training nlp models: A concise overview. *arXiv*, arXiv:2004.08900.

Cornelis Joost Van Rijsbergen. 1974. Foundation of evaluation. *Journal of Documentation*, 30(4):365–373.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*.

Bertie Vidgen, Tristan Thrush, Zeerak Waseem, and Douwe Kiela. 2020. Learning from the worst: Dynamically generated datasets to improve online hate detection. *arXiv preprint arXiv:2012.15761*.

Krishnapriya Vishnubhotla and Saif M. Mohammad. 2022. Tweet emotion dynamics: Emotion word usage in tweets from us and canada. In *Proceedings of the 13th International Conference on Language Resources and Evaluation*, pages 4162–4176.

Zeerak Waseem and Dirk Hovy. 2016. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL Student Research Workshop*, pages 88–93.

Wenjie Yin and Arkaitz Zubiaga. 2021. Towards generalisable hate speech detection: a review on obstacles and solutions. *PeerJ Computer Science*, 7:e598.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. A robustly optimized BERT pre-training approach with post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227.

Steven Zimmerman, Udo Kruschwitz, and Chris Fox. 2018. Improving hate speech detection with deep learning ensembles. In *Proceedings of the 11th International Conference on Language Resources and Evaluation*, pages 2546–2553.