Information Processing and Management 60 (2023) 103433

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

Systematic keyword and bias analyses in hate speech detection

Gretel Liz De la Peña Sarracén *, Paolo Rosso

Universitat Politècnica de València, Camino de Vera, s/n, 46022, Valencia, Spain

ARTICLE INFO

Keywords:
Hate speech detection
Keyword extraction
Bias analysis
Bias mitigation

ABSTRACT

Hate speech detection refers broadly to the automatic identification of language that may be considered discriminatory against certain groups of people. The goal is to help online platforms to identify and remove harmful content. Humans are usually capable of detecting hatred in critical cases, such as when the hatred is non-explicit, but how do computer models address this situation? In this work, we aim to contribute to the understanding of ethical issues related to hate speech by analysing two transformer-based models trained to detect hate speech. Our study focuses on analysing the relationship between these models and a set of hateful keywords extracted from the three well-known datasets. For the extraction of the keywords, we propose a metric that takes into account the division among classes to favour the most common words in hateful contexts. In our experiments, we first compared the overlap between the extracted keywords with the words to which the models pay the most attention in decision-making. On the other hand, we investigate the bias of the models towards the extracted keywords. For the bias analysis, we characterize and use two metrics and evaluate two strategies to try to mitigate the bias. Surprisingly, we show that over 50% of the salient words of the models are not hateful and that there is a higher number of hateful words among the extracted keywords. However, we show that the models appear to be biased towards the extracted keywords. Experimental results suggest that fitting models with hateful texts that do not contain any of the keywords can reduce bias and improve the performance of the models.

1. Introduction

In recent years, much research has been carried out to deal with the negative impact of hate speech on online social media. There is a lot of debate about the definition of hate speech and what can be considered a hateful message. In most cases, hate speech is understood as a language that attacks or belittles, incites violence or hatred against groups based on certain characteristics such as physical appearance, religion, gender identity, or others, and it can occur with different linguistic styles, even in subtle forms or when humour is used (Fortuna & Nunes, 2018). This definition highlights the subjective factor in the task of identifying hatred. In subtle cases, whether a message attacks or discriminates depends on the recipient's perspective. While this task can be difficult for humans, computational models face an even greater challenge.

Hate Speech Detection (HSD) is the use of natural language processing and machine learning techniques to automatically identify hate speech. The goal is to detect and remove harmful content on online platforms and social media to create a safer and more inclusive environment for all users. HSD is usually treated as a binary classification problem between the class of hateful texts and the class of non-hateful texts, and, models are typically trained on datasets of labelled texts to recognize features indicative of hate speech. Alkomah and Ma (2022) recently provided a review of textual hate speech detection systems and pointed out the

* Corresponding author.
E-mail addresses: gredela@posgrado.upv.es (G.L. De la Peña Sarracén), prosso@dsic.upv.es (P. Rosso).

https://doi.org/10.1016/j.ipm.2023.103433
Received 6 February 2023; Received in revised form 18 April 2023; Accepted 6 June 2023
Available online 17 June 2023

widespread use of transformers-based machine learning models. These are complex models that have shown outstanding results in many natural language processing tasks (Latif et al., 2023). The question in HSD is, what do the models learn to identify hatred? We have a fairly intuitive hypothesis. We suppose that the models pay more attention to potential hateful patterns that are common in hateful contexts.

Research Questions and Contributions. In this work, we examine two transformers-based models trained on hate speech (HSD models) with three popular collections of tweets in English to investigate our hypothesis. Our focus is on the relationship between hateful keywords and the words to which these models pay more attention to.[1] We aim to answer the following research questions.

RQ1: Do the HSD models pay mostly attention to hateful words?
We rely on Captum, a library proposed by Kokhlikyan et al. (2020) to interpret the results of deep learning models. This tool allows us to obtain the weight that the models give to each element of a text in the decision-making process. Then, we rank the words within a collection of texts and consider that the models pay more attention to the top words.
RQ2: Is it possible to identify the words to which the HSD models pay more attention using simple statistics?
Comparing the highest weighted words for the HSD models with frequent words in hateful contexts, we show that we can hardly predict the words that the HSD models pay more attention to. Surprisingly, most hateful words seem to be extracted with simple statistics. To extract this second group of words, we propose an unsupervised keyword extraction method. The idea is to penalize words with high frequency in the class of non-hateful texts, even though they appear frequently in the other class. This allows us to focus only on hateful contexts.
RQ3: Should we focus on mitigating HSD model bias towards hateful keywords?
Complementing the study of the relationship between hateful keywords and the words that the HSD models focus on, we analyse the effect of attempting to mitigate the bias of these models towards hateful words. First, we calculate the bias of the HSD models with one metric inspired by the concept of fairness and one metric introduced by Borkan, Dixon, Sorensen, Thain, and Vasserman (2019), which is based on the ROC-AUC score. We then use two mechanisms to attempt to reduce the bias: forcing the models to fine-tune with (1) non-hateful texts with hateful keywords and (2) hateful texts without hateful keywords. Finally, we evaluate how these mechanisms can affect the performance of the HSD models.
Two critical cases in HSD motivate our investigation of this mitigation: (1) non-hateful texts containing hateful words. For instance, *"oh shit, I accidentally blocked you. whoops"*, is a non-hateful text in the popular dataset of Waseem and Hovy (2016). However, this example may be misclassified as hateful by associating *shit* with hateful content. This association can be done by an over-generalized training of models from datasets where this word is much more frequent in hateful contexts. (2) Hateful texts without hateful words. HSD is pretty simple in texts with explicit hate, but we consider here the possibility of coming across texts in which hate is expressed without hateful words (Frenda, Patti, & Rosso, 2022; Sanchez-Junquera et al., 2021).

The contribution of this paper is twofold: first, we propose and evaluate an unsupervised keyword extraction method to automatically detect relevant words for one class in a text collection[2]. In particular, we use this method for the HSD task in order to extract hateful keywords. Second, we analyse the effect of bias mitigation for critical cases of hate speech detection.

The rest of this paper is organized as follows. Section 2 summarizes the related work and Section 3 introduces the text collections that we use in this work. Sections 4–6 present our studies and findings for the research questions respectively. In particular, the description of our proposal to extract keywords is in Section 5. Finally, Section 7 presents limitations and ethical concerns of our research work, and Section 8 concludes the paper.

2. Related work

Different strategies have been proposed for hate speech detection, which ranges from methods based on linguistic characteristics to machine learning techniques. Alkomah and Ma (2022), Poletto, Basile, Sanguinetti, Bosco, and Patti (2021) and Velankar, Patil, and Joshi (2022) provide reviews on methods and datasets. However, most of the systems proposed in recent years focus on transformer-based models, showing outstanding results.

Recently, Malik, Pang, and Hengel (2022) presented an empirical comparison of 14 shallow and deep models for hate speech detection on three benchmarks of different data characteristics. The experimental results showed that transformer-based hate speech detectors have promising performance, although pointing out that they still have some weak points. Shishah and Fajri (2022) compared current approaches in hate speech detection in order to analyse the influence of different approaches and their applicability in the real world. The study was conducted on eight hate speech datasets and showed that a transformer model approach is able to outmatch many of the previous hate speech detection models by significant G-Means and F1 scores. In addition, the last competitions on hate speech detection are evidence of the prestige of transformer-based models, as the top places are generally based on these types of models. For example, the Hate Speech Detection (HaSpeeDe 2) shared task in 2020 (Manuela et al., 2020) was the second edition of the shared task HaSpeeDe in 2018 (Bosco et al., 2018). In the first edition, the best systems were fundamentally based on deep learning methods such as Convolutional Neural Networks and Recurrent Neural Networks. While in the second edition transformers-based models were the popular choice. Lavergne, Saini, Kovács, and Murphy (2020) used BERT, ALBERTo and UmBERTo language models to reach the first position in HaSpeeDe 2.

[1] NOTE: This paper contains examples of potentially explicit offencive content. They do not represent the views of the authors.
[2] We make available a python package [https://pypi.org/project/hmrf/].

Table 1
Essential information of the text collections.

Collection	Focus	Size	# hateful texts	% hateful tweets
Hateval	Misogyny/Racism	10,000	4,209	42.09%
W&H	Sexism/Racism	10,574	2,783	26.32%
Founta	Hate Speech	56,470	3,635	6.44%

In this paper, we study two transformer-based models to evaluate the relationship between the decision-making of these models and hateful keywords. We investigate how the models are biased towards the keywords and evaluate two strategies to mitigate the bias. The strategies that we use are based on fine-tuning and the main effort is in filtering the data that is used to fit the models. Following, we position our work w.r.t. other studies on bias in hate speech detection.

Bias analysis in hate speech detection. Wiegand, Ruppenhofer, and Kleinbauer (2019) analysed how high-ranking scores in biased datasets that contain mostly implicit abuse, are due to bias modelling in those datasets. In our work, we analyse this report by studying the relationship between hateful keywords and transformers-based models. Balkir, Nejadgholi, Fraser, and Kiritchenko (2022) presented a feature attribution method for explaining text classifiers in the context of hate speech detection. The authors showed that different values of necessity and sufficiency for identity terms correspond to different kinds of false positive errors, exposing sources of classifier bias against marginalized groups. They studied the bias with mere mentions of identity terms that result in false positive predictions. We also intend to evaluate how the mentions of hateful keywords influence the models, but we characterize the bias with two well-defined metrics that allow us to study the bias quantitatively.

Bias mitigation. Nozza, Volpetti, and Fersini (2019) analysed the bias in misogyny identification and evaluated the mitigation of bias by four strategies based on terms with the most imbalanced class distributions. The experimental results showed the ability of the bias mitigation strategy to reduce the bias of the misogyny detection model proposed by the authors of the work. While this is an interesting result, the impact of bias reduction for classification needs to be investigated. In our work, together with the evaluation of the bias reduction, we add the study of the performance variation. Xia, Field, and Tsvetkov (2020) stated the bias in annotated training data causes text to often be mislabelled as hate speech with a high false positive rate by current hate speech classifiers. The authors used adversarial training to mitigate this bias and show that the false positive rate seems to reduce while minimally affecting the performance of hate speech classification. We not only assess the performance of the models but also analyse how the bias varies with strategies aimed at reducing bias. Mozafari, Farahbakhsh, and Crespi (2020) introduced a bias mitigation mechanism by using a regularization method to re-weight input samples. The objective was to decrease the effects of highly correlated n-grams of the training set with class labels. The results showed the existence of a racial bias in trained classifiers. The authors also showed the bias was reduced with the bias mitigation mechanism. To evaluate this mechanism, the authors employed a cross-domain approach in which they use the trained classifiers on a dataset to predict the labels of two new datasets. Unlike that way of measuring bias reduction, we rely on two metrics for a quantitative evaluation.

3. Datasets

In this work we use three text collections with English tweets: HatEval (Basile et al., 2019), Waseem & Hovy (W&H) (Waseem & Hovy, 2016) and Founta (Founta et al., 2018).

- **HatEval:** It is the dataset used for Task 5 of SemEval 2019.[3] The objective of that task was the detection of hate speech against immigrants and women in Spanish and English tweets. The tweets were collected by monitoring potential victims of hatred, downloading the history of identified haters, and filtering tweets with terms related to hate speech. This collection is composed of 9000 tweets for training and 1000 tweets for development.
- **Waseem & Hovy (W&H):** A popular dataset referenced in several studies (Arango, Pérez, & Poblete, 2019; Fortuna & Nunes, 2018; Gröndahl, Pajola, Juuti, Conti, & Asokan, 2018; Poletto et al., 2021; Schmidt & Wiegand, 2017). It is composed of tweets annotated as sexist, racist, or non-hate. It is available as a list of identifiers. It contains 16,906 tweets, of which 3378 are labelled as sexist and 1970 as racist. In the construction of this dataset, some tweets were first collected with a manual search of common terms related to religion, sex, gender, and ethnic minorities. Then, the most frequent terms from the hateful tweets of this first set of tweets were used to collect the rest of the tweets.
- **Founta:** It is a dataset that contains tweets annotated as hateful, abusive, spam, or normal. The data was collected by random sampling and some heuristics to boost the proportion of abusive texts. The boosted random sampling technique relies on increasing minority classes to address the problem of bias in the entire text set. This collection is composed of 3635 tweets tagged as hateful, 10,122 tagged as abusive, 13,404 tagged as spam, and 52,835 tagged as normal.

Table 1 shows the essential information of the data we used in our study. Note that we used all tweets from Hateval, both training and development. From W&H we could not download all the tweets, we only used those that were accessible given their identifier. Lastly, from Founta we only used tweets tagged as hateful or normal. We considered the 'normal' texts as the non-hateful class.

[3] https://competitions.codalab.org/competitions/19935

Table 2
Links to the pre-trained models used in this work.

Model	Architecture	Paper	URL
BERT	bert	Aluru, Mathew, Saha, and Mukherjee (2020)	https://huggingface.co/Hate-speech-CNERG/dehatebert-mono-english
ROBERTA	roberta-base	Vidgen, Thrush, Waseem, and Kiela (2021)	https://huggingface.co/facebook/roberta-hate-speech-dynabench-r4-target

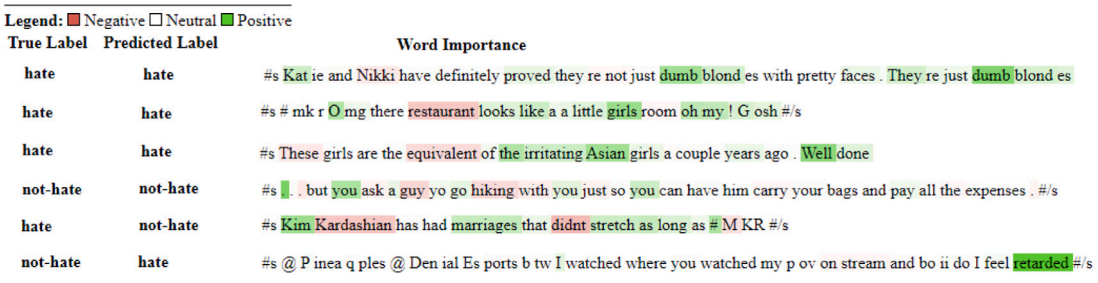


Fig. 1. Attributions with Transformers-Interpret for randomly selected texts from the text collections. The positive class (green) corresponds to the class of hateful texts. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

4. Transformer-based models for hate speech detection: Analysis of salient words

In this section, we will address **RQ1** with the aim of gaining insights into the behaviour of the HSD models by exploring the weight that they assign to words in the decision-making process. We face one of the most important problems in eXplainable Artificial Intelligence (XAI) since these models are often considered as black boxes. However, there is a branch of research on explainability that is focused on facilitating the identification of different features that contribute to the results of complex models for natural language processing. Danilevsky et al. (2020) present a taxonomy with five main explanation techniques: feature importance, surrogate model, example-driven, provenance-based, and declarative induction. We focus on the importance of the features, to investigate the scores of the different features used to generate the final prediction. We rely on Captum,[4] a library that offers several attribution algorithms that allow us to understand the importance of input features. This tool was introduced by Kokhlikyan et al. (2020) and has been broadly used to explain transformer-based models.

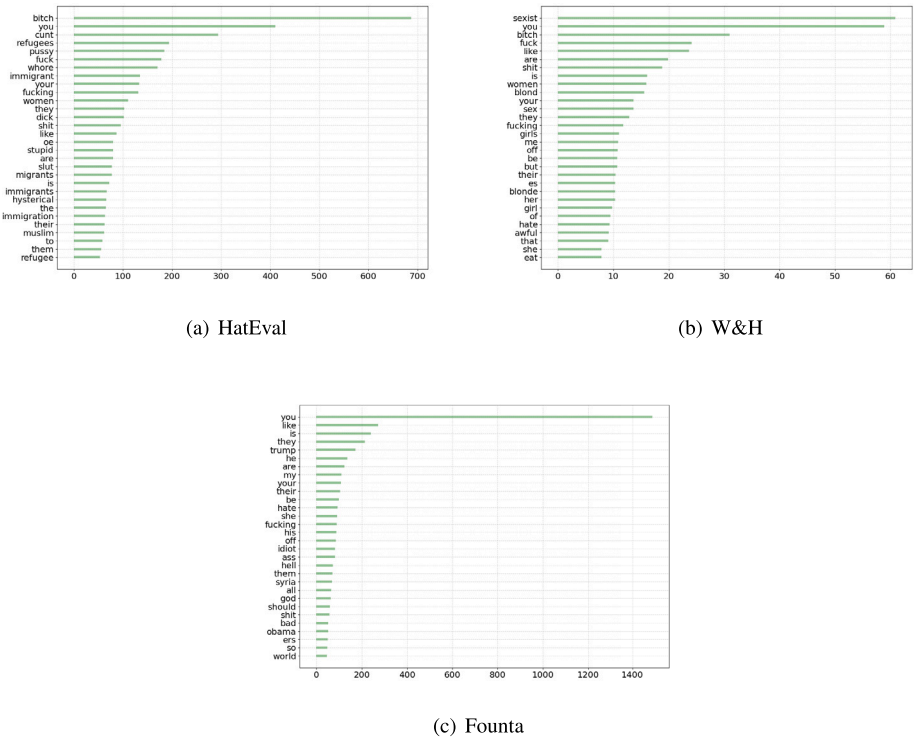
We use Transformers-Interpret, a model interpretation library for PyTorch that wraps Captum with the Huggingface transformer package. Unlike Captum, this tool focuses solely on natural language processing, making the interpretation of the HSD task easier. In particular, we employ the sequence classification explainer method that allows us to compute the attribution of the terms in a text using a model. Positive attribution numbers indicate that a word contributes positively towards the predicted class, while negative numbers indicate that a word contributes negatively towards the predicted class. However, attribution explanations are not limited to the predicted class and we force the method to obtain the attributions w.r.t the hateful class. Thus, the method returns a list of tuples containing words and their associated attribution scores for the hateful class.

For our experiments, we use the transformer-based models shown in Table 2. They are trained to detect hate speech and are accessible in HuggingFace for English.

Findings. Fig. 1 shows some examples taken randomly from the text collections and classified with ROBERTA. Words highlighted in green indicate a positive attribution to the hateful class, while red indicates a negative attribution. The first four examples are well-classified hateful texts. Note that some words with a positive attribution are not hateful, but they are important to understand the text as hateful. For example, in the case of text number 3, the word "Well" has a high positive attribution (dark green). While this is not a hateful word, in the context of this text it is used to support a hateful message. Surprisingly, the model has been able to understand it. This suggests that the model, beyond learning hateful words, learns to identify hateful contexts. Despite these interesting results, we observe some cases where the model fails. Let us look at the last two examples. Text number 5 is a case of subtle hatred where there are no explicit hateful words and the model fails to detect the text as hateful. In addition, example number 6 is a non-hateful text that contains a hateful word and is also misclassified. In other words, although this model appears to be quite robust to identify hatred in subtle cases, it may fail in the critical cases exemplified by Examples 5 and 6. In Section 6 we analyse this issue in more detail by examining the bias of the model towards a set of hateful words.

Once we obtain a model attribution for the words in each text, we calculate the general word score in a text collection. The score is the sum of the attributions that the words receive in the texts of the collection in which they appear. Note that the higher the attribution that the model gives to a word in each instance, the higher its score is. Similarly, the score of the words to which the model gives negative attribution is small. Fig. 2 shows the top 30 words of the ranking generated with ROBERTA in each text collection. The way to access the analogous result for BERT is in Appendix.

[4] https://captum.ai/



(a) HatEval (b) W&H

(c) Founta

Fig. 2. Ranking of the words ROBERTA pays the most attention in each text collection.

Table 3
Hate speech detection results in terms of F1-score. Numbers in bold indicate the best significant result with a significance level of .05.

Model	HateEval	W&H	Founta
BERT	**0.8004**	0.6034	0.6188
ROBERTA	0.7157	**0.7236**	**0.7180**

We observe that the number of hateful words represents less than 50% of the salient words of the models. This suggests that the models seem to pay equal or more attention to the target of hate (girls, immigrants, etc.) than to hateful words. This corresponds to what we saw in Fig. 1, where this behaviour was relevant in one example of subtle hatred. However, we also identify some cases where models can fail. We then evaluate the performance of the models in each text collection and Table 3 shows the results in terms of F1-score (see Powers, 2015 to understand the F1 metric).

For the analysis of statistical analysis, we used McNemar's test as Dietterich (1998) recommends. This is a paired non-parametric statistical hypothesis test that allows to evaluate once each model for comparison. The default assumption of this test is that two models disagree with the same amount. Then, if the null hypothesis is rejected, it suggests that there is evidence to say that the models disagree in different ways.

Table 3 shows the best results per text collection in bold, indicating that there is a statistically significant difference in the disagreements between BERT and ROBERTA.

The comparison between these models is beyond the scope of this paper, so the comparison between models with statistic analysis becomes more important in Section 6. In this section, the most relevant is to note that the results in the three datasets with ROBERTA are similar, although among the salient words in HatEval there seems to be a higher amount of hateful words. It does not seem that there is a direct relationship between the number of hateful words the HSD models pay more attention and their performance.

5. Analysis of hateful keyword

Once we have analysed the salient words of the models, we may address **RQ2** from Section 1 extracting hateful keywords from the text collections taking into account some statistics. We then compare the result with the words the HSD models pay more attention to. For keywords extraction, we propose the Harmonic Mean of Relative Frequencies (HMRF), a measure that takes into account the frequency of words in hateful texts along with their frequency in the rest of the texts. The idea is different from the well-known TFIDF since we try to favour words that maximize the difference between their frequency in the hateful texts and their

frequency in the rest of the texts. HMRF is also different from the Polarized Weirdness Index (PWI) of words used by Poletto et al. (2021), which is based on Ahmad et al. (1999). PWI is the ratio of the relative frequency of a word in a class against its relative frequency in the other class. We consider instead the distribution of words when we calculate and combine the frequency to score each word.

5.1. Method for automatic keyword extraction

The first step of our method is to eliminate the stopwords to consider only words with semantic weight. Then, from the rest of the words we only take into account the nouns, adjectives, and verbs. The next step consists of identifying the most relevant words in each class of texts. We refer to the relevance of a word based on its relative frequency in the class. We then characterized the keywords as the words with the largest difference between the relevance in the set of hateful texts and the relevance in the set of non-hateful texts. In the third step, we expand the list of words with phrases.

We start from the idea that potentially hateful words are more likely to appear in hateful texts. However, we have considered that if we analyse only the relevance of words within hateful texts, we will probably select as keywords those that are more common due to the topic in the collection of texts and not because they are words frequently used to transmit hate.

To deal with this issue we consider not only the relevance of each word in the set of hateful texts but also in the set of non-hateful texts. The procedure is then to search for words that are very relevant in the hateful texts and not very relevant in the rest of the texts at the same time. In this sense, we use the concept of 'little relevant words in non-hateful texts'. That way, we propose a measure that allows ranking the words to make the most significant ones. This strategy also helps us to discard words that may indicate hate but are frequently used in non-hateful texts in a given context. For example, if a text collection has been built from a thread of posts about feminism, the word 'feminist' is likely to appear frequently not only in the hateful texts but also in the rest of the texts. Therefore, we prefer not to select that word as a relevant word and look for other more discriminating words that indicate hate in that particular context.

Harmonic mean of relative frequencies

The Harmonic Mean of Relative Frequencies (HMRF) is the measure we propose to assign a score to each word w. Basically, we calculate the score of w using the harmonic mean of two relative frequencies of w into a set of texts S. The relative frequencies are (1) the frequency of w only considering the texts of S (Eq. (1)), and (2) the frequency of w in S with respect to its frequency in the entire collection of texts C (Eq. (2)). The variable k identifies all possible words in the text set and the indicator $\mathbf{1}_{(t)}(d)$ defines the number of times that the word w appears in a text t.

$$f_1^S(w) = \frac{\sum_{t \in S} \mathbf{1}_{(w)}(t)}{\sum_k \sum_{t \in S} \mathbf{1}_{(k)}(t)} \tag{1}$$

$$f_2^S(w) = \frac{\sum_{t \in S} \mathbf{1}_{(w)}(t)}{\sum_{t \in C} \mathbf{1}_{(w)}(t)} \tag{2}$$

Then, we use the cumulative distribution function (CDF)[5] on the relative frequencies. This is a distribution function of a random variable X: $F_X(x) = P(X \leq x)$. So, $CDF(f_1^S)$ indicates the ratio of words that will take a value of f_1^S less than or equal to $f_1^S(w)$. Similarly, $CDF(f_2^S(w))$ indicates the ratio of words with a value of f_2^S equal to or lower than $f_2^S(w)$. Thus, by using CDF, it is possible to see where the value of either $f_1^S(w)$ or $f_2^S(w)$ lies in the distribution of the words in a cumulative way.

Finally, we use the harmonic mean (Sheldon, Paul, & Wade, 2001) to combine both $CDF(f_1^S(w))$ and $CDF(f_2^S(w))$. It gives the greatest weight to the smallest item of a series, and the impact of large outliers is mitigated. Eq. (3) specifies how the harmonic mean is used to obtain the final score for w.

$$HMRF_S(w) = \frac{2 * CDF(f_1^S(w)) * CDF(f_2^S(w))}{CDF(f_1^S(w)) + CDF(f_2^S(w))} \tag{3}$$

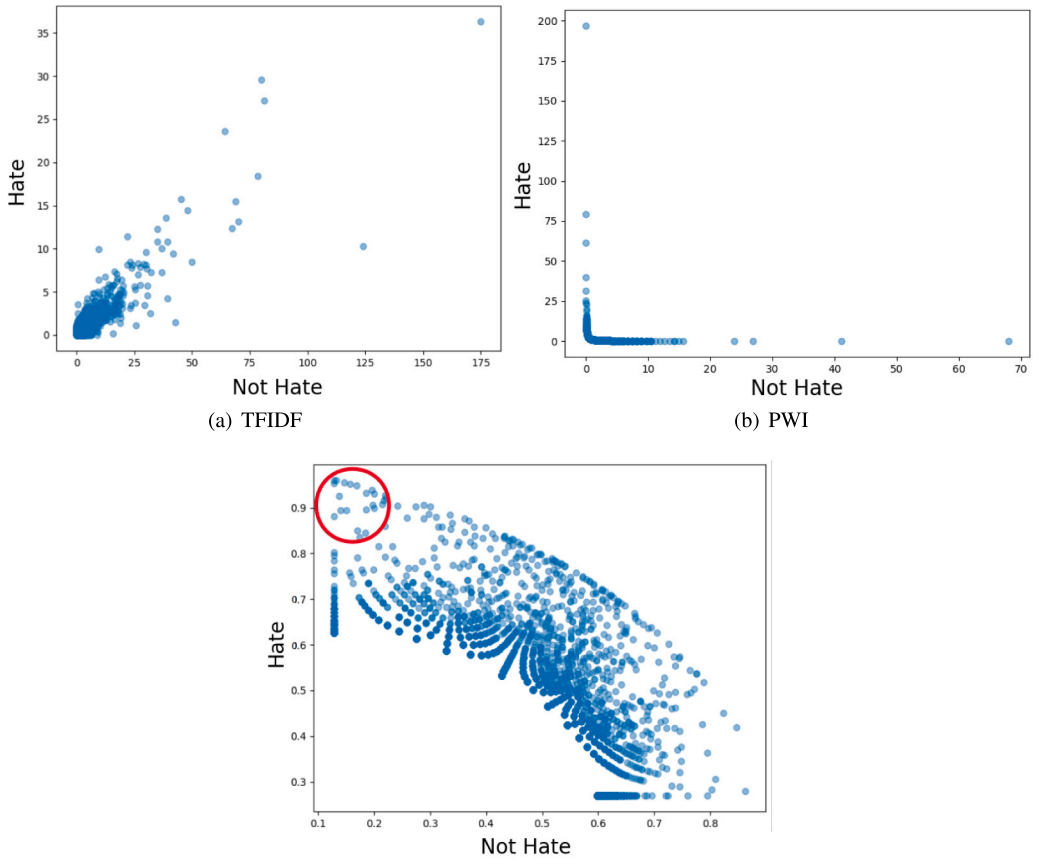
In text collections for HSD, we generally have the sets of hateful (H) and non-hateful (N) texts. Thus, the set S refers to each of the sets H and N, and the set C to {H ∪ N}. In this way, w is represented by the tuple ($\text{HMRF}_N(w)$,$\text{HMRF}_H(w)$).

Fig. 3 shows word-tuple representations extracted from the HatEval dataset (Basile et al., 2019) as points on a plane. Figs. 3(a) and 3(b) show the points according to TFIDF and PWI respectively. While Fig. 3(c) shows the points according to the HMRF measure. The words that interest us as hateful keywords are in the circle in the upper left corner of this last figure.

As we can see, PWI allows us to easily identify relevant words in hateful texts compared to TFIDF. However, HMRF provides a clearer idea of the distribution of words. Note that with PWI all the points with high values in the hateful axis have the same value in the other axis. With HMRF in contrast we obtain a distribution in which we can see not just those points that are more relevant in the hateful texts, but also which of them are less relevant in the non-hateful texts.

In order to extract the keywords, we order the words descendingly according to the HMRF_H - HMRF_N difference. For the words with the same score, we establish a ranking, so that the most relevant word is the one with the highest HMRF_H.

[5] https://www.sciencedirect.com/topics/mathematics/cumulative-distribution-function



(a) TFIDF (b) PWI

(c) HMRF

Fig. 3. Representation of terms.

Phrases construction

In addition, we include the possibility of expanding the keywords list generated in the previous step, by adding phrases composed of more than one word.[6] We take into account the concept of collocation considering only the words that are already in the list. i.e. the phrases are identified with two or more hateful keywords that commonly co-occur in context. In order to obtain the phrases we only use hateful texts and a strategy to analyse the co-occurrence of words.

The strategy is based on pointwise mutual information (PMI). This measures how much more likely the words in a sequence $W = (w_1, \ldots, w_n)$ co-occur than if they occur independently (Eq. (4)). Where n is the size of the sequence and P is its probability in the text set.

$$PMI(W) = log_2 \frac{P(w_1, \ldots, w_n)}{\prod_{i=1}^{n} P(w_i)} \tag{4}$$

We consider bigrams and trigrams when generating the phrases. In the case of bigrams we consider the structures <adjective,noun> and <noun,noun>, while for trigram we consider the structures <adjective,ALL,noun> and <noun,ALL,noun>. ALL refers to any word, including those out of the hateful keyword list. Note that in these patterns we only consider adjectives and nouns. This is because the connotation of an adjective can vary depending on the noun it modifies. This dependency is usually less strong in the case of other parts of speech such as verbs. For example, the word 'f*ck' is usually enough to express hate, regardless of the words with which it co-occurs. The objective of these patterns is to limit the number of phrases.

[6] In the scope of this work, we use 'term' to refer to a word or a phrase.

Table 4
Sets of keywords extracted with HMRF and YAKE.

Collection	HMRF	YAKE
HateEval	deportthemall, illegal, enddaca, bitch, suck, aliens, housegop, hoes, citizens, stoptheinvasion, borders, sendthemback, bitch suck	free speech time, illegal immigrants, trump, bitch, immigrant, woman, illegal immigration, immigrant children illegal muslim migrant
W&H	football, sports, sexist, female, woman, feminist, feminism, bitch, drivers, girl, womenagainstfeminism, girl call, female sports, female football call, rights, equal rights	mkr, kat, sexist, woman, call girls, people, mkr kat, mkr kat kat, mkr kat call, call me sexist, mkr katie, mkr hey kat, mkr god kat, mkr krazy-eyes kat, mkr omg
Founta	hate, racist, liar, hated, feminist, bombing, refugees, disgusting, terrorists, retarded, bitches, bastards, bombs, gay, hating, evil, kill, hoes, blacks, missiles, blacks hating, evil bastards, hate hoes, hate bitches, feminist bitches	youtube video, transponder snail, day, today, people, love, time, trump, video, found a transponder, isis calls trump, good, make, world health day, happy birthday, great, found, happy trump loves russia,

Table 5
Percentage of occurrence of keywords per class. HS refers to the class of hateful texts and N-HS to the class of non-hateful texts. The largest difference between the percentages of the classes is in bold.

Class	HateEval		W&H		Founta	
	N-HS	HS	N-HS	HS	N-HS	HS
HMRF	**48.41**	**70.28**	**48.30**	**76.67**	**25.36**	**56.95**
YAKE	36.75	45.74	56.80	60.92	24.56	18.74

5.2. Experimental setup

Taking advantage of the study of the relationship between keywords extracted with HMRF and the salient words of the HSD models, we include another set of keywords obtained with an alternative strategy: keywords extracted with YAKE (Campos et al., 2020), a general-purpose method for keyword extraction.

Thus, we compare five sets of words in each text collection:

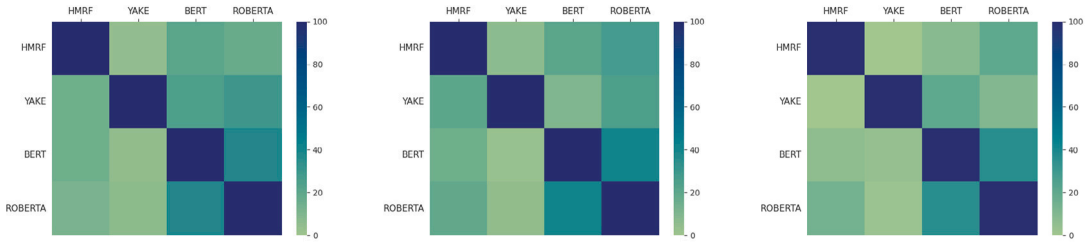
- **BERT:** Salient words for the BERT model.
- **ROBERTA:** Salient words for the ROBERTA model.
- **HMRF:** Keywords extracted with our method.
- **YAKE:** Keywords extracted with a method that does not consider the division between classes. Experiments carried out on different text collections report that this method outperforms state-of-the-art methods such as TFIDF, KP-Miner, RAKE, TextRank, SingleRank, ExpandRank, TopicRank, TopicalPageRank, PositionRank and MultipartiteRank (Campos et al., 2020).

5.3. Discussion

Analysis of the results of our method. We first compared the behaviour of HMRF and YAKE taking into account two aspects that are expected from a keyword extraction strategy. We analysed whether the extracted words reflect the focus of the source dataset. Next, we analysed if the extracted keywords are really potentially hateful terms (words and phrases). Table 4 shows some examples of terms extracted with HMRF and YAKE for each text collection. We set the number of hateful keywords to 20. Note that the final size of the sets of terms increases as phrases are added. We extract all possible phrases. Thus, the final amount of extracted terms are not fixed.

At first glance, we can see little overlap between both sets of keywords (using HMRF and YAKE). However, in most cases, the keywords seem to reflect the focus of the source text collection. In Hateval, for example, where the focus is misogyny and racism, keywords like ‘bitch’, ‘deportthemall’, and ‘immigrants’ reflect what is expected. An exception is the case of W&H with YAKE. Most of the terms are very frequent in the texts, but they are not English words. Perhaps, it would be convenient to carry out a text pre-processing. Alternatively, our penalization of very frequent terms in non-hateful texts seems to deal with this problem. Note that with HMRF for the same collection (W&H), the terms better reflect the focus of the texts.

Taking a closer look at the extracted words, we can see that most of them are actually potentially hateful words. In the case of the Founta collection, YAKE extracts keywords that do not express hatred at all, such as ‘love’, ‘good’, ‘happy birthday’, etc. By contrast, HMRF manages to extract more hateful terms for the same collection. These terms mostly make more sense to express hatred such as ‘hate’, ‘bastard’, ‘kill’, etc. Table 5 confirms this, by showing the percentage of occurrence of the words of each set in the classes of the text collections. As expected, the difference between the occurrence of the words in each class is greater when considering the words extracted with HMRF.



(a) HatEval (b) W&H (c) Founta

Fig. 4. Heatmap of the overlap between each pair of word sets. Each cell represents the percentage of overlap taking as the total the size of the group of words on the X-axis.

This can suggest that the set of words extracted without taking into account the division between classes (YAKE) is more similar to the set of salient words of the HSD models. Remember that in Section 4 we saw that the highest percentage of words to which HSD models pay the most attention are usually not hateful words. To investigate this assumption, let us analyse the overlap between each pair of word sets.

Overlap among keywords sets. Fig. 4 shows the overlap between each pair of word sets. Each cell represents the percentage of overlap calculated w.r.t to the set in columns, i.e. the size of the set of columns is taken as the total to calculate the percentage that represents the overlap. Let us focus on the first and second columns of each heatmap, which correspond to the keywords extracted with HMRF and YAKE, respectively. Although YAKE extracts a set of keywords that seems to be more similar to the salient words of the HSD models, we can see that the percentage of keywords extracted with HMRF that is salient (for the HSD models) is higher. However, the overlap of the salient words and the keywords by HMRF is not large enough to state that with our method we can predict the words to which the HSD models will pay the most attention.

6. Bias mitigation

Motivated by the low similarity between the keywords extracted with our method and the salient words of the HSD models observed in Section 5, in this section we investigate **RQ3** of Section 1: our goal is to evaluate the role of hateful keywords, extracted with HMRF, in hate speech detection with HSD models. We first analyse the bias of the models towards the extracted keywords. Following the course of the analysis in the previous section, we consider both the keywords extracted with HMRF and the keywords extracted with YAKE. We then use two strategies to try to mitigate the bias and assess how bias varies. The strategies are based on fine-tuning with data in which the occurrence of the keywords is taken into account. Finally, we investigate the relationship between the bias variation and the performance of the HSD models.

6.1. Experimental setup

Bias estimation. Various types of bias have been defined in the literature, as well as a number of problems related to classification models (Garrido-Muñoz, Montejo-Ráez, Martínez-Santiago, & Ureña-López, 2021). We focus on the model bias w.r.t HSD. In our case, a model is considered biased when it tends to make more errors towards a class due to the presence of keywords. This phenomenon mainly occurs when there is much more representation of the keywords in the class of hateful texts, and models learn to classify as hateful, text with those keywords.

Let us formalize the model bias for our study, as well as the way in which we evaluate its impact on HSD. We first follow the idea of Garrido-Muñoz et al. (2021), which uses the 'fairness' concept. The authors of the paper argue that fairness is equivalent to zero-bias systems in machine learning. Thus, it allows us to formalize and quantify the bias in some way.

Formalization (bias based on fairness)

Given the distribution $< X, K, Y, \hat{Y} >$, referring X to instances from a text collection, K to one keyword, Y to the true classes of the instances and $\hat{Y}$ to the predicted classes by a model. Here $\hat{Y} = 1$ means a classification in the class of hateful texts, while $\hat{Y} = 0$ means a classification in the class of non-hateful texts. For K, $K = 1$ means the presence of the keyword in a text and $K = 0$ means the absence. Then, according to the concept of fairness, bias can be defined as Eq. (5).

$$bias = |P(\hat{Y} = 1|K = 1) - P(\hat{Y} = 1|K = 0)| \tag{5}$$

Note that this is equivalent to equal positive probabilities for when the keyword is present and when it is not. Thus, equal probabilities is a good estimator of bias (fairness), such that the higher the value of this metric, the higher the bias. Consider that

this is not a cognitive bias, rather this is related to the estimation of parameters in statistical modelling. We follow the Eqs. (6) and (7) to calculate the probabilities.

$$P(\hat{Y} = 1|K = 1) = \frac{\#\text{ Texts containing K and classified as hateful}}{\#\text{ Texts containing K}} \tag{6}$$

$$P(\hat{Y} = 1|K = 0) = \frac{\#\text{ Texts not containing K and classified as hateful}}{\#\text{ Texts not containing K}} \tag{7}$$

Bias based on ROC-AUC metrics

Note that the bias based on fairness only takes into account the instances classified as hateful. Alternatively, we use the metrics introduced in Borkan et al. (2019), which was used in the competition 'Jigsaw Unintended Bias in Toxicity Classification'.[7] This metric considers both texts classified as hateful and texts classified as non-hateful, by using three sub-metrics based on the ROC-AUC[8] score on three specific subsets of the test for each keyword, such that each metric captures a different aspect of bias:

- **Subgroup** AUC: The test set is restricted to only the examples that contain the specific keyword. A low value in this metric means the model does a bad job to distinguish between hateful and non-hateful texts that contain the keyword.
- **BPSN** (Background Positive, Subgroup Negative) AUC: Test set is restricted to the non-hateful examples that contain the keyword and the hateful examples that do not. A low value in this metric means that the model confuses non-hateful examples that contain the keyword with hateful examples that do not. That is, the model predicts higher hateful scores than it should for non-hateful examples containing the keyword.
- **BNSP** (Background Negative, Subgroup Positive) AUC: Test set is restricted to the hateful examples that contain the keyword and the non-hateful examples that do not. A low value means that the model confuses hateful examples that contain the keyword with non-hateful examples that do not. That is, the model predicts lower hateful scores than it should for hateful examples containing the keyword.

We calculate the bias per keyword and combine them with the following generalized mean:

$$M_p(n) = \left(\frac{1}{K} \sum_{k=1}^{K} m_{k,n}^p \right)^{\frac{1}{p}} \tag{8}$$

$m_{k,n}$ is the bias metric calculated for keyword k and metric n. K is the number of keywords (subgroups).

We set p to −5 just like in the competition, where the objective was to encourage competitors to improve the model for the subgroups with the lowest model performance. Finally, we combine the overall AUC ($AUC_{overall}$) with the generalized mean to calculate the model score. $AUC_{overall}$ refers to ROC-AUC for the full test set. Here, the lower the score, the higher the bias.

$$score = w_0 AUC_{overall} + \sum_{n=1}^{N} w_n M_p(n) \tag{9}$$

N is the number of metrics (**Subgroup**, **BPSN**, **BNSP**).

$w_i, i = \overline{0, N}$ is a weight for the relative importance of each metric. We set all four w to 0.25.

In summary, we use the following two metrics to estimate the bias in our experiment:

- **b1**: Bias based on ROC-AUC metrics. The lower the value of this metric, the higher the bias.
- **b2**: Bias based on fairness. The higher the value of this metric, the higher the bias.

Strategies for bias mitigation

In order to mitigate the bias, we rely on fine-tuning the HSD models. The goal is to make a small fit in the parameters of the models with a very small learning rate. We focus this fit towards a specific set of keywords when choosing the data for fine-tuning. In this sense, we follow the following strategies:

- V1: Data only contains hateful texts without keywords.
- V2: Data only contains non-hateful texts with keywords.
- V3: Data contains random texts.

The first two strategies are aligned with the critical cases that we discussed in Section 1. We want to fit the HSD models for those cases considering the sets of keywords that we are studying, i.e. HMRF and YAKE. Thus, we have to analyse the behaviour of the fine-tuned models in 4 variants:

- $V1_{HMRF}$: Strategies V1 taking HMRF as the set of keywords.
- $V1_{YAKE}$: Strategies V1 taking YAKE as the set of keywords.

[7] https://www.kaggle.com/competitions/jigsaw-unintended-bias-in-toxicity-classification/overview

[8] https://onlinelibrary.wiley.com/doi/10.1111/j.1466-8238.2007.00358.x

Table 6
Estimated biases for the not fitted model (Original) and the fine-tuned models ($V1_{HMRF}$, $V1_{YAKE}$, $V2_{HMRF}$, $V2_{YAKE}$, V3) for BERT and ROBERTA. We report bias based on ROC-AUC metrics (**b1**) and bias based on fairness (**b2**) towards the keywords specified in the columns (HMRF and YAKE).

	HateEval				W&H				Founta			
	HMRF		YAKE		HMRF		YAKE		HMRF		YAKE	
	b1	b2	b1	b2	b1	b2	b1	b2	b1	b2	b1	b2
BERT												
Original	0.712	0.344	0.754	0.219	0.548	0.186	0.583	0.136	0.469	0.207	0.578	0.016
$V1_{HMRF}$	0.705	0.332	–	–	0.563	0.203	–	–	0.498	0.275	–	–
$V1_{YAKE}$	–	–	0.735	0.240	–	–	0.598	0.167	–	–	0.616	0.020
$V2_{HMRF}$	0.710	0.348	–	–	0.531	0.172	–	–	0.563	0.190	–	–
$V2_{YAKE}$	–	–	0.754	0.219	–	–	0.578	0.133	–	–	0.575	0.015
V3	0.710	0.339	0.747	0.219	0.552	0.189	0.580	0.129	0.474	0.214	0.585	0.015
ROBERTA												
Original	0.536	0.243	0.665	0.158	0.661	0.284	0.682	0.225	0.613	0.317	0.679	0.021
$V1_{HMRF}$	0.541	0.239	–	–	0.701	0.294	–	–	0.640	0.363	–	–
$V1_{YAKE}$	–	–	0.671	0.167	–	–	0.708	0.237	–	–	0.711	0.031
$V2_{HMRF}$	0.534	0.244	–	–	0.627	0.259	–	–	0.606	0.313	–	–
$V2_{YAKE}$	–	–	0.665	0.158	–	–	0.680	0.220	–	–	0.680	0.019
V3	0.536	0.245	0.661	0.166	0.674	0.289	0.690	0.228	0.629	0.333	0.696	0.028

- $V2_{HMRF}$: Strategies V2 taking HMRF as the set of keywords.
- $V2_{YAKE}$: Strategies V2 taking YAKE as the set of keywords.

We also include the third strategy to compare the results of the first two strategies and assess if they are relevant to mitigate the bias in relation to this traditional fine-tuning strategy.

Fine-tuning and evaluation details. Selecting data for fine-tuning in one dataset, we chose 2500 instances of each of the other datasets. For example, for the evaluation of a model fitted with $V1_{HMRF}$ in HatEval, we select 2500 hateful texts with keywords of HMRF from W&H and 2500 from Founta.

We experiment with the two models introduced in Section 4 (BERT and ROBERTA). We train with batches of size 16 in 3 epochs and evaluate the whole dataset with batches of 16 instances. We optimize all models with Adam (Kingma and Ba, 2015) and a learning rate of 1×10^{-7}. The performance of the models is reported in terms of F1-score, along with the P value of McNemar's statistical test introduced in Section 4. This last test was used to compare if each variant of a model varies significantly with a significance level $\alpha = .05$ i.e. we compare each fine-tuned variant with the original model (where no fit is made).

6.2. Results and discussion

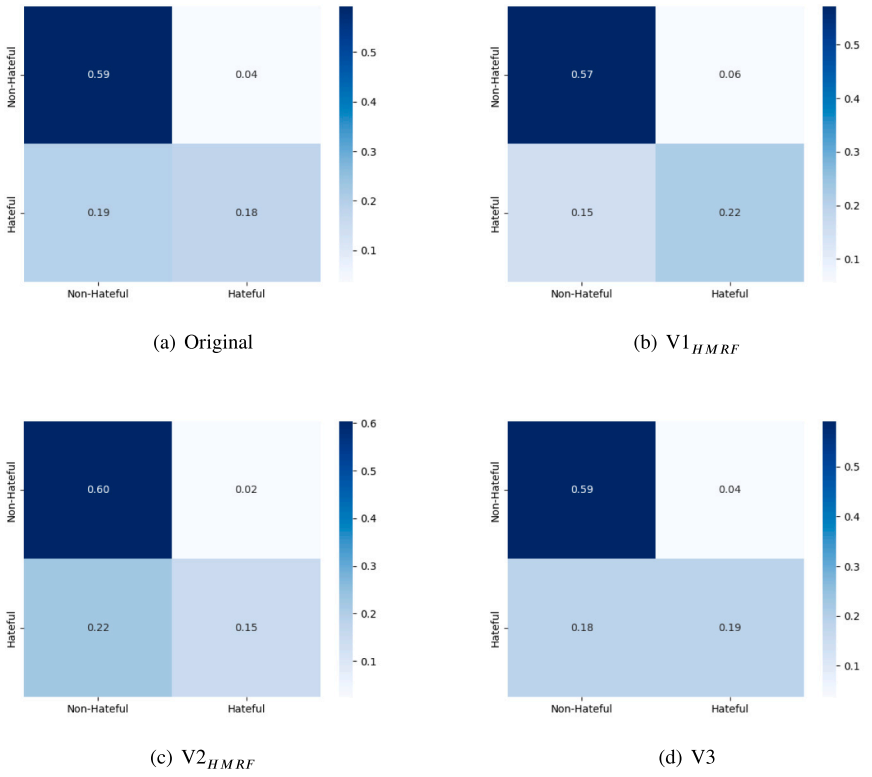
A summary of the estimated biases, per model and dataset, is provided in Table 6.
Following is a list of the most important findings when analysing the table:

- The bias of the Original model is greater towards HMRF keywords than towards YAKE keywords. This corroborates the finding of the previous section, where we noticed that the percentage of overlap between the salient words of the HSD models with the HMRF keywords is greater than with those of YAKE.
- In general we do not observe a pattern in the variation of the bias with V3.
- V1 seems to reduce the bias **b1** (increase the value) with respect to the Original model. We do not observe this behaviour in HatEval with BERT. This suggests that using hateful instances without keywords makes the model fit to identify hatred in cases where those keywords are not present.
- V2 seems to increase the bias **b1** (reduce the value) with respect to the Original model. We do not observe this behaviour in Founta with BERT.
- The bias **b2** seems to have the opposite behaviour: V1 tends to increase **b2** with respect to the Original model, while V2 tends to reduce it.

Once we have analysed the variation of the bias towards the keywords extracted with HMRF and YAKE, we need to evaluate how this influences the variation of the performance of the HSD models. Table 7 summarizes the results for all models in terms of F1-score and P value. We do the statistical analysis between each variant after the fine-tuning and the Original model. We observe that only in one case ($V1_{HMRF}$ in HatEval) the results of the models are not significantly different. Therefore, it makes sense to analyse the variation of F1 in relation to the variation of bias. Taking into account the analysed bias variation, we notice that in most cases, less bias **b1** (greater value) corresponds to a greater F1 value and that greater bias **b2** (greater value) corresponds to a greater F1 value. We have seen that V1 tends to reduce **b1** and increase **b2**, therefore V1 seems to be good for improving the value of F1. Likewise, we observe that in most cases, more bias **b1** (lower value) corresponds to a lower F1 value and that lower bias **b2** (lower value) corresponds to a lower F1 value. V2 tends to increase **b1** and decrease **b2**, therefore V2 seems to worsen the value

Table 7
Performance of the not fitted model (Original) and the fine-tuned models ($V1_{HMRF}$, $V1_{YAKE}$, $V2_{HMRF}$, $V2_{YAKE}$, V3) for BERT and ROBERTA. We consider $\alpha = .05$.

	HateEval		W&H		Founta	
	F1	P value	F1	P value	F1	P value
BERT						
Original	**0.8004**	–	0.6034	–	0.6188	–
$V1_{HMRF}$	0.7927	P=.13	0.6161	P<.001	**0.6378**	P<.001
$V1_{YAKE}$	0.7890	P<.001	**0.6199**	P<.001	0.6375	P<.001
$V2_{HMRF}$	0.7994	P<.001	0.5572	P<.001	0.6124	P<.001
$V2_{YAKE}$	0.8003	P<.001	0.5995	P<.001	0.6147	P<.001
V3	0.7975	P=.01	0.6111	P<.001	0.6213	P<.001
ROBERTA						
Original	0.7157	–	0.7236	–	0.7180	–
$V1_{HMRF}$	**0.7170**	P<.001	**0.7588**	P<.001	0.7157	P<.001
$V1_{YAKE}$	0.7161	P<.001	0.7560	P<.001	0.7153	P<.001
$V2_{HMRF}$	0.7140	P<.001	0.6806	P<.001	0.7185	P<.001
$V2_{YAKE}$	0.7157	P<.001	0.7196	P=.02	0.7189	P<.001
V3	0.7165	P<.001	0.7346	P<.001	**0.7213**	P<.001



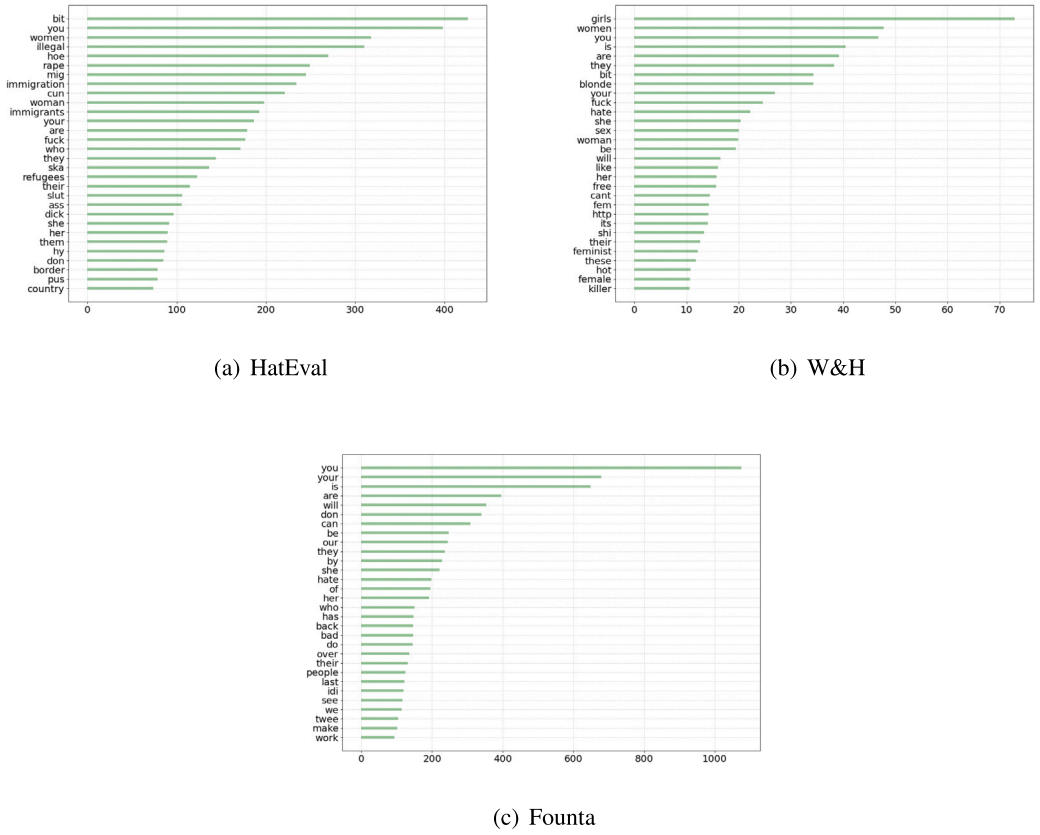
(a) Original (b) $V1_{HMRF}$

(c) $V2_{HMRF}$ (d) V3

Fig. 5. Confusion matrices for classification with ROBERTA in W&H. Rows represent the actual labels and columns the predicted labels.

of F1. However, note that there are unexpected cases, where the behaviour is different. Therefore, we cannot assert that there is a correlation.

Finally, let us remember that **b2** only takes into account instances classified as hateful. With V1 this class is favoured, so the number of texts classified as hateful will tend to increase (see an example in Fig. 5). The opposite happens with V2, whereby disfavouring the class of hateful texts, the difference between the number of true positives and false positives will tend to be less. This also helps us to understand how V1 can favour the values of F1 by improving the positive class (hateful texts).



(a) HatEval (b) W&H

(c) Founta

Fig. 6. Ranking of the words to which BERT pays the most attention in each text collection.

7. Limitations and ethical concerns

In this work, we have provided some insights regarding the relationship between keywords extracted from text collection and the salient words of two transformers-based models trained for HSD. There are two limitations that we want to discuss in this section. First, note that we rely on an interpretability model to determine the salient words of the models. Therefore, our analysis at this point depends on the results of this interpretability model. On the other hand, our HMRF metric is based on penalizing very frequent words in the negative class of the source text collection. This helps us find potential hateful keywords, but it can also rule out words with hate content. This phenomenon should not be very common and we are interested in hateful words that are not used frequently in non-hate texts. However, we must bear in mind that this can be a weak point in some tasks.

Our study may have some ethical concerns as it focuses on hate speech detection. Please, note that our goal is limited to assisting in this effort to help online platforms to identify and remove hateful content.

8. Conclusions

Transformer-based models have arrived to mark an important leap in different natural language processing tasks. Among them, hate speech detection has been favoured in recent years. However, one problem we face is understanding what these models learn to detect hatred. In this work, we focused on studying the relationship between a set of keywords extracted from three text collections and the salient words of two transformers-based models pre-trained in hate speech detection. For the extraction of the keywords, we proposed HMRF, a metric that focuses on extracting very frequent keywords in the class of hateful texts and at the same time less frequent in the other class. First of all, we noted that HMRF manages to extract a large number of hateful words, unlike other leading keyword extraction methods. Then, we observed that there is not much similarity between the keywords extracted with HMRF and the words that the transformer-based models pay the most attention to. We even noticed that the set of salient words from the models has fewer hateful words than those extracted with HMRF. Finally and most importantly, we analysed the bias of the models towards the HMRF keywords with two types of metrics and evaluated two strategies to try to mitigate the bias. The experimental results suggested that the bias towards hateful keywords can be reduced when fine-tuning the models with hateful texts where the keywords are not present and that this reduction may imply an improvement in the F1. This finding provides an incentive for future research efforts to analyse the bias towards hateful keywords taken from external resources such as HurtLex (Bassignana, Basile,

& Patti, 2018), a lexicon that contains hateful words that are independent of a specific text collection. In addition, we suggest a deeper analysis of the selection of data size for the fine-tuning strategy that showed good results in reducing bias. We believe that it can influence the variation of the bias and the performance of the models that we observed with the size of the data that we used.

Data availability

I have shared the link to my data.

Acknowledgements

This work was done in the framework of the research project on Fairness and Transparency for equitable NLP applications in social media, funded by MCIN/AEI/10.13039/501100011033 and by ERDF, EU A way of making EuropePI.

Appendix

See Fig. 6.

References

Ahmad, K., Gillam, L., Tostevin, L., et al. (1999). University of surrey participation in TREC8: Weirdness indexing for logical document extrapolation and retrieval (WILDER). In *TREC* (pp. 1–8).
Alkomah, F., & Ma, X. (2022). A literature review of textual hate speech detection methods and datasets. *Information*, *13*(6), 273.
Aluru, S. S., Mathew, B., Saha, P., & Mukherjee, A. (2020). Deep learning models for multilingual hate speech detection. arXiv preprint arXiv:2004.06465.
Arango, A., Pérez, J., & Poblete, B. (2019). Hate speech detection is not as easy as you may think: A closer look at model validation. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval* (pp. 45–54).
Balkir, E., Nejadgholi, I., Fraser, K. C., & Kiritchenko, S. (2022). Necessity and sufficiency for explaining text classifiers: A case study in hate speech detection. In *Proceedings of the 2022 conference of the North American chapter of the association for computational linguistics: human language technologies* (pp. 2672–2686).
Basile, V., Bosco, C., Fersini, E., Debora, N., Patti, V., Pardo, F. M. R., et al. (2019). Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter. In *13th International workshop on semantic evaluation* (pp. 54–63). Association for Computational Linguistics.
Bassignana, E., Basile, V., & Patti, V. (2018). Hurtlex: A multilingual lexicon of words to hurt. In *5th Italian conference on computational linguistics, vol. 2253* (pp. 1–6). CEUR-WS.
Borkan, D., Dixon, L., Sorensen, J., Thain, N., & Vasserman, L. (2019). Nuanced metrics for measuring unintended bias with real data for text classification. In *Companion proceedings of the 2019 world wide web conference* (pp. 491–500).
Bosco, C., Felice, D., Poletto, F., Sanguinetti, M., Maurizio, T., et al. (2018). Overview of the evalita 2018 hate speech detection task. In *Ceur workshop proceedings, vol. 2263* (pp. 1–9). CEUR.
Campos, R., Mangaravite, V., Pasquali, A., Jorge, A., Nunes, C., & Jatowt, A. (2020). YAKE! Keyword extraction from single documents using multiple local features. *Information Sciences*, *509*, 257–289.
Danilevsky, M., Qian, K., Aharonov, R., Katsis, Y., Kawas, B., & Sen, P. (2020). A survey of the state of explainable AI for natural language processing. arXiv preprint arXiv:2010.00711.
Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, *10*(7), 1895–1923.
Fortuna, P., & Nunes, S. (2018). A survey on automatic detection of hate speech in text. *ACM Computing Surveys*, *51*(4), 1–30.
Founta, A., Djouvas, C., Chatzakou, D., Leontiadis, I., Blackburn, J., Stringhini, G., et al. (2018). Large scale crowdsourcing and characterization of Twitter abusive behavior. In *Proceedings of the international AAAI conference on web and social media, vol. 12, no. 1*.
Frenda, S., Patti, V., & Rosso, P. (2022). Killing me softly: Creative and cognitive aspects of implicitness in abusive language online. *Natural Language Engineering (JNLE)*, 1–22.
Garrido-Muñoz, I., Montejo-Ráez, A., Martínez-Santiago, F., & Ureña-López, L. A. (2021). A survey on bias in deep NLP. *Applied Sciences*, *11*(7), 3184.
Gröndahl, T., Pajola, L., Juuti, M., Conti, M., & Asokan, N. (2018). All you need is "Love" evading hate speech detection. In *Proceedings of the 11th ACM workshop on artificial intelligence and security* (pp. 2–12).
Kokhlikyan, N., Miglani, V., Martin, M., Wang, E., Alsallakh, B., Reynolds, J., et al. (2020). Captum: A unified and generic model interpretability library for pytorch. arXiv preprint arXiv:2009.07896.
Latif, S., Zaidi, A., Cuayahuitl, H., Shamshad, F., Shoukat, M., & Qadir, J. (2023). Transformers in speech processing: A survey. arXiv preprint arXiv:2303.11607.
Lavergne, E., Saini, R., Kovács, G., & Murphy, K. (2020). Thenorth@ haspeede 2: Bert-based language model fine-tuning for italian hate speech detection. In *7th Evaluation campaign of natural language processing and speech tools for Italian. Final workshop, vol. 2765*.
Malik, J. S., Pang, G., & Hengel, A. v. d. (2022). Deep learning for hate speech detection: a comparative study. arXiv preprint arXiv:2202.09517.
Manuela, S., Gloria, C., Di Nuovo, E., Frenda, S., Stranisci, M. A., Bosco, C., et al. (2020). Haspeede 2@ evalita2020: Overview of the evalita 2020 hate speech detection task. In *Proceedings of the seventh evaluation campaign of natural language processing and speech tools for Italian. Final workshop* (pp. 1–9). CEUR.
Mozafari, M., Farahbakhsh, R., & Crespi, N. (2020). Hate speech detection and racial bias mitigation in social media based on BERT model. *PLoS One*, *15*(8), Article e0237861.
Nozza, D., Volpetti, C., & Fersini, E. (2019). Unintended bias in misogyny detection. In *IEEE/WIC/ACM international conference on web intelligence* (pp. 149–155).
Poletto, F., Basile, V., Sanguinetti, M., Bosco, C., & Patti, V. (2021). Resources and benchmark corpora for hate speech detection: A systematic review. *Language Resources and Evaluation*, *55*, 477–523.
Powers, D. M. (2015). What the F-measure doesn't measure: Features, flaws, fallacies and fixes. arXiv preprint arXiv:1503.06410.
Sánchez-Junquera, J., Rosso, P., Montes, M., Chulvi, B., et al. (2021). Masking and BERT-based models for stereotype identication. *Procesamiento Del Lenguaje Natural (SEPLN)*, *67*, 83–94.
Schmidt, A., & Wiegand, M. (2017). A survey on hate speech detection using natural language processing. In *Proceedings of the fifth international workshop on natural language processing for social media* (pp. 1–10).
Sheldon, A., Paul, B., & Wade, R. (2001). Harmonic function theory. *Graduate Texts in Mathematics*, *137*.
Shishah, W., & Fajri, R. M. (2022). Large comparative study of recent computational approach in automatic hate speech detection. *TEM Journal*, *11*(1), 82.
Velankar, A., Patil, H., & Joshi, R. (2022). A review of challenges in machine learning based automated hate speech detection. arXiv preprint arXiv:2209.05294.
Vidgen, B., Thrush, T., Waseem, Z., & Kiela, D. (2021). Learning from the worst: Dynamically generated datasets to improve online hate detection. In *ACL*.
Waseem, Z., & Hovy, D. (2016). Hateful symbols or hateful people? Predictive features for hate speech detection on Twitter. In *Proceedings of the NAACL student research workshop* (pp. 88–93). Association for Computational Linguistics.
Wiegand, M., Ruppenhofer, J., & Kleinbauer, T. (2019). Detection of abusive language: The problem of biased datasets. In *Proceedings of the 2019 conference of the North American chapter of the Association for Computational Linguistics: Human language technologies, volume 1 (Long and short papers)* (pp. 602–608).
Xia, M., Field, A., & Tsvetkov, Y. (2020). Demoting racial bias in hate speech detection. *SocialNLP*.