Universitat Politècnica de València

Departamento de Sistemas Informáticos y Computación

PhD Thesis

Gretel Liz De la Peña Sarracén

On the Keyword Extraction and Bias Analysis, Graph-based Exploration and Data Augmentation for Abusive Language Detection in Low-Resource Settings

Advisor
Prof. Paolo Rosso
Universitat Politècnica de València, Spain

January, 2024

Acknowledgments

I would like to express my deepest gratitude to my supervisor, Prof. Paolo Rosso, for his unwavering support, invaluable guidance, and insightful feedback throughout the course of this research. Their expertise and mentorship have been instrumental in shaping the direction of this thesis.

I am also indebted to the people who invited and advised me during my internship in Manheim, Germany: Prof. Simone Paolo Ponzetto (University of Mannheim), Dr. Robert Litschko (LMU Munich) and Dr. Goran Glavaš (University of Würzburg).

Many thanks go to my colleagues and fellow researchers at the Universitat Politècnica de València, whose collaboration and support have been both enjoyable and intellectually enriching. The synergy between academia and industry fostered by Symanto has been instrumental in achieving the objectives of this thesis.

I extend my appreciation to the reviewers of this thesis Dr. Arkaitz Zubiaga (Queen Mary University of London), Prof. Rafael Valencia (Universidad de Murcia), and Dr. Tommaso Caselli (Rijksuniveristeit Groningen); thank you very much for your work on my thesis. I would also like to thank the members of the PhD committee of this thesis Dr. Arkaitz Zubiaga (Queen Mary University of London), Dra. Sara Tonelli (Fondazione Bruno Kessler), and Dra. Aitziber Atutxa (University of the Basque Country).

Special thanks go to my family for their tireless encouragement and understanding. Their love and support have been my pillars throughout this challenging yet rewarding endeavor. Also to the friends who have taken an interest in my emotional and physical well-being during my doctoral research.

Last but not least, I would like to thank all the participants and individuals who generously contributed their time. I am truly grateful.

Abstract

Abusive language detection is a task that has become increasingly important in the modern digital age, where communication takes place via various online platforms. The increase in online interactions has led to an increase in the occurrence of abusive language. Addressing such content is crucial to maintaining a safe and inclusive online environment. However, this task faces several challenges that make it a complex and ongoing area of research and development. In particular, detecting abusive language in environments with sparse data poses an additional challenge, since the development of accurate automated systems often requires large annotated datasets.

In this thesis we investigate different aspects of abusive language detection, paying particular attention to environments with limited data. First, we study the bias toward abusive keywords in models trained for abusive language detection. To this end, we propose two methods for extracting potentially abusive keywords from datasets. We then evaluate the bias toward the extracted keywords and how this bias can be modified in order to influence abusive language detection performance. The analysis and conclusions of this work reveal evidence that it is possible to mitigate the bias and that such a reduction can positively affect the performance of the models. However, we notice that it is not possible to establish a similar correspondence between bias mitigation and model performance in low-resource settings with the studied bias mitigation techniques.

Second, we investigate the use of models based on graph neural networks to detect abusive language. On the one hand, we propose a text representation framework designed with the aim of obtaining a representation space in which abusive texts can be easily distinguished from other texts. On the other hand, we evaluate the ability of models based on convolutional graph neural networks to classify abusive texts. The next part of our research focuses on analyzing how data augmentation can influence the performance of abusive language detection. To this end, we investigate two well-known techniques based on the principle of vicinal risk minimization and propose a variant for one of them. In addition, we evaluate simple techniques based on the operations of synonym replacement, random insertion, random swap, and random deletion.

The contributions of this thesis highlight the potential of models based on graph neural networks and data augmentation techniques to improve abusive language detection, especially in low-resource settings. These contributions have been published in several international conferences and journals.

Resumen

La detección del lenguaje abusivo es una tarea que se ha vuelto cada vez más importante en la era digital moderna, donde la comunicación se produce a través de diversas plataformas en línea. El aumento de las interacciones en estas plataformas ha provocado un aumento de la aparición del lenguaje abusivo. Abordar dicho contenido es crucial para mantener un entorno en línea seguro e inclusivo. Sin embargo, esta tarea enfrenta varios desafíos que la convierten en un área compleja y que demanda de continua investigación y desarrollo. En particular, detectar lenguaje abusivo en entornos con escasez de datos presenta desafíos adicionales debido a que el desarrollo de sistemas automáticos precisos a menudo requiere de grandes conjuntos de datos anotados.

En esta tesis investigamos diferentes aspectos de la detección del lenguaje abusivo, prestando especial atención a entornos con datos limitados. Primero, estudiamos el sesgo hacia palabras clave abusivas en modelos entrenados para la detección del lenguaje abusivo. Con este propósito, proponemos dos métodos para extraer palabras clave potencialmente abusivas de colecciones de textos. Luego evaluamos el sesgo hacia las palabras clave extraídas y cómo se puede modificar este sesgo para influir en el rendimiento de la detección del lenguaje abusivo. El análisis y las conclusiones de este trabajo revelan evidencia de que es posible mitigar el sesgo y que dicha reducción puede afectar positivamente el desempeño de los modelos. Sin embargo, notamos que no es posible establecer una correspondencia similar entre la variación del sesgo y el desempeño de los modelos cuando hay escasez datos con las técnicas de reducción del sesgo estudiadas.

En segundo lugar, investigamos el uso de redes neuronales basadas en grafos para detectar lenguaje abusivo. Por un lado, proponemos una estrategia de representación de textos diseñada con el objetivo de obtener un espacio de representación en el que los textos abusivos puedan distinguirse fácilmente de otros textos. Por otro lado, evaluamos la capacidad de redes neuronales convolucionales basadas en grafos para clasificar textos abusivos. La siguiente parte de nuestra investigación se centra en analizar cómo el aumento de datos puede influir en el rendimiento de la detección del lenguaje abusivo. Para ello, investigamos dos técnicas bien conocidas basadas en el principio de minimización del riesgo en la vecindad de instancias originales y proponemos una variante para una de ellas. Además, evaluamos técnicas simples basadas en el reemplazo de sinónimos, inserción aleatoria, intercambio aleatorio y eliminación aleatoria de palabras. Las contribuciones de esta tesis ponen de manifiesto el potencial de las redes neuronales basadas en grafos y de las técnicas de aumento de datos para mejorar la detección del lenguaje abusivo, especialmente cuando hay limitación de datos. Estas contribuciones han sido publicadas en conferencias y revistas internacionales.

Resum

La detecció del llenguatge abusiu és una tasca que s'ha tornat cada vegada més important en l'era digital moderna, on la comunicació es produïx a través de diverses plataformes en línia. L'augment de les interaccions en estes plataformes ha provocat un augment de l'aparició de llenguatge abusiu. Abordar este contingut és crucial per a mantindre un entorn en línia segur i inclusiu. No obstant això, esta tasca enfronta diversos desafiaments que la convertixen en una àrea complexa i contínua de recerca i desenvolupament. En particular, detectar llenguatge abusiu en entorns amb escassetat de dades presenta desafiaments addicionals pel fet que el desenvolupament de sistemes automàtics precisos sovint requerix de grans conjunts de dades anotades.

En esta tesi investiguem diferents aspectes de la detecció del llenguatge abusiu, prestant especial atenció a entorns amb dades limitades. Primer, estudiem el biaix cap a paraules clau abusives en models entrenats per a la detecció de llenguatge abusiu. Amb este propòsit, proposem dos mètodes per a extraure paraules clau potencialment abusives de col·leccions de textos. Després avaluem el biaix cap a les paraules clau extretes i com es pot modificar este biaix per a influir en el rendiment de la detecció de llenguatge abusiu. L'anàlisi i les conclusions d'este treball revelen evidència que és possible mitigar el biaix i que esta reducció pot afectar positivament l'acompliment dels models. No obstant això, notem que no és possible establir una correspondència similar entre la variació del biaix i l'acompliment dels models quan hi ha escassetat dades amb les tècniques de reducció del biaix estudiades.

En segon lloc, investiguem l'ús de xarxes neuronals basades en grafs per a detectar llenguatge abusiu. D'una banda, proposem una estratègia de representació textual dissenyada amb l'objectiu d'obtindre un espai de representació en el qual els textos abusius puguen distingir-se fàcilment d'altres textos. D'altra banda, avaluem la capacitat de models basats en xarxes neuronals convolucionals basades en grafs per a classificar textos abusius. La següent part de la nostra investigació se centra en analitzar com l'augment de dades pot influir en el rendiment de la detecció del llenguatge abusiu. Per a això, investiguem dues tècniques ben conegudes basades en el principi de minimització del risc en el veïnatge d'instàncies originals i proposem una variant per a una d'elles. A més, avaluem tècniques simples basades en el reemplaçament de sinònims, inserció aleatòria, intercanvi aleatori i eliminació aleatòria de paraules. Les contribucions d'esta tesi destaquen el potencial de les xarxes neuronals basades en grafs i de les tècniques d'augment de dades per a millorar la detecció del llenguatge abusiu, especialment quan hi ha limitació de dades. Estes contribucions han sigut publicades en revistes i conferències internacionals.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

1.1 Problem Description

In the contemporary digital era, the ubiquitous presence of social media and online platforms has led to the proliferation of abusive language as a complex and critical problem. Abusive language is a multifaceted phenomenon that encloses not only explicit threats and derogatory comments but also more subtle forms of expression that can have extensive implications for individuals and society as a whole. The use of abusive language, including hate speech and different forms of offensive or harmful content, produces a significant challenge to maintaining healthy online communities and promoting inclusive digital spaces. Therefore, understanding the impact and prevalence of abusive language in the current digital scenario is not only a critical social concern but also a topic of active research in fields such as natural language processing, psychology, and sociology.

According to various research (Waseem and Hovy, 2016; Fortuna and Nunes, 2018a; Malmasi and Zampieri, 2018; Jurgens et al., 2019; Caselli et al., 2021a; Kiritchenko et al., 2021; Pamungkas et al., 2023), the abusive language phenomenon covers a wide spectrum of abusive behaviors, including offense, online aggression, hate speech, abuse, stereotyping, cyberbullying and doxxing. Poletto et al. (2021) also expose these concepts and illustrate their connections in Figure 1.1. Currently, most approaches emphasize offensive language (Pradhan et al., 2020), abusive language (Vidgen and Derczynski, 2020), and hate speech (Yin and Zubiaga, 2021; Alkomah and Ma, 2022) for their social impact. Although they are communicative categories that belong to the same phenomenon, they differ in features and targets. Offensive language may be used to provoke a negative emotional reaction, but it might not necessarily target specific groups or identities. Hate speech, on the other hand, promotes prejudice and discrimination against specific groups based on their characteristics. Abusive language encompasses a wide range of harmful language, but may not be based on the same prejudices as

hate speech.

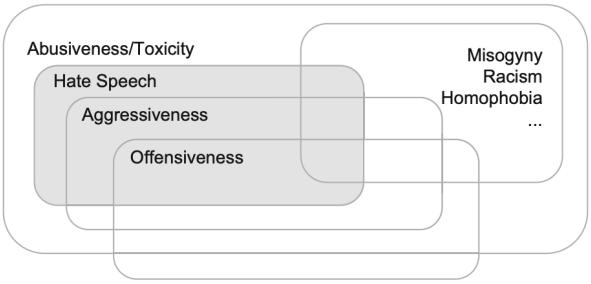


Figure 1.1: Abusive language phenomena (source Poletto et al. (2021)).

A major challenge in the study of the abusive language phenomenon is low-resource settings. Typically, these settings have significant limitations in terms of available data. The lack of sufficient labeled data hinders the training of robust models capable of distinguishing between abusive and non-abusive content. Moreover, abusive language can be subtle and context-dependent (Sánchez-Junquera et al., 2021a; Frenda et al., 2023; Merlo et al., 2023). Detecting such abuses is particularly difficult when working with limited data and resources. Therefore, ingenious techniques are essential solutions to adapt to specific linguistic contexts.

In this thesis, we aim to explore the abusive language phenomena as the task of abusive language detection in low-resource settings. Our research is focused on the concepts of offensive language, hate speech, and abusive language in general. We address this study from three perspectives: 1) analysis of bias toward abusive keywords in models, 2) graph-based models, and 3) data augmentation (see Section 1.5). These perspectives involve different techniques that allow us to analyze the problem of data limitation.

The research is structured as follows. First, we propose two techniques to extract keywords from text collections and use them in the context of abusive language detection. One of the methods leverages the BERT multi-head self-attention mechanism, while the other uses some statistics from the collections. However, both methods focus on identifying terms that are highly relevant to abusive texts but irrelevant to other texts. We then analyze the bias of the models toward these terms, and the conclusions provide insight into how this bias can affect the performance of abusive language detection. The next part of our research focuses on graph neural networks. We first propose a strategy based on graph auto-encoders to generate a text representation (embeddings), and the findings show its capability to discriminate abusive language. Then, we use a model based on convolutional graph neural networks to evaluate the effectiveness of graph-based models compared to state-of-the-art models in low-resource settings. Finally, we study data augmentation techniques to diversify the data in low-resource settings. We

evaluate two strategies and propose a variant based on the principle of vicinal risk minimization (Chapelle et al., 2000).

The rest of this chapter overviews reference methods for the detection of abusive language, hate speech, and offensive language, as well as for abusive language detection in low-resource settings. Next, we motivate our work and present our objectives. Finally, we present our research questions, contributions, and the structure of this thesis.

1.2 Abusive Language and Hate Speech

Abusive language and hate speech involve the use of harmful language but generally differ in their scope, intent, and the specific types of harm they can cause. Abusive language can be directed at an individual or group for reasons like personal conflicts or disagreements. This does not necessarily involve targeting people based on their characteristics (Swamy et al., 2019; Clarke and Grieve, 2017). While hate speech is specifically directed at groups based on their inherent characteristics, such as race, religion, or ethnicity. This promotes discrimination and prejudice against these groups (Waltman and Mattheis, 2017; ElSherief et al., 2018; Bilewicz and Soral, 2020). Consequently, abusive language can cause emotional distress and harm to the target but does not necessarily lead to discrimination against a specific group. On the other side, hate speech can contribute to a broader culture of discrimination and can incite violence or harassment in specific communities.

Although there are differences between these two concepts, they are closely related concepts. In the field of natural language processing, hate speech detection (HSD) and abusive language detection (ALD) share commonalities in terms of their objectives and methodologies. They both fall under the broader category of text classification tasks aimed at identifying harmful language in textual data (MacAvaney et al., 2019; Putri et al., 2020; Anand et al., 2023). Therefore, it is usual to find that many works for hate speech detection also encompass or reference abusive language detection. This overlap occurs because hate speech is a subset of abusive language (see Figure 1.1), and many of the techniques and models developed for HSD can be applied to a wider range of ALD tasks. Furthermore, the linguistic features and patterns associated with hate speech often overlap with those of other abusive languages. Models trained to identify hate speech are likely to recognize common linguistic cues shared with other forms of abusive language. Due to the limited availability of data labeled for other subcategories of abusive language, research can focus on the detection of hate speech acknowledging that studies can also serve as a basis for detecting abusive language more broadly. The distinction between HSD and ALD depends on the specific goals and context of the research or application.

In this thesis, both concepts are discussed and studied, but abusive language detection is used as the general terminology to explore the phenomenon of abusive language.

1.2.1 Abusive Language Detection

A large number of machine learning methods have been used for hate speech detection and abusive language detection, ranging from traditional machine learning to deep learning techniques. In the early days, the strategies were mainly based on traditional machine learning approaches. Models like Naive Bayes (NB) (Kiilu et al., 2018), Logistic Regression (LR) (Ginting et al., 2019) and Support Vector Machines (SVM) (MacAvaney et al., 2019; Sevani et al., 2021; Das et al., 2023), were employed for classification. Furthermore, some approaches relied on features extracted from text data, such as n-grams, sentiment analysis, and part-of-speech tagging. Waseem and Hovy (2016) analyzed the impact of various linguistic features in conjunction with character n-grams on hate speech detection and concluded that this approach provides a solid foundation. However, most methods based solely on these kinds of approaches are not typically considered effective because they have several limitations, including issues related to context, generalization, and the dynamic nature of the abusive language.

The introduction of artificial neural networks, particularly deep learning, marked a significant shift. The methods involved the use of Recurrent Neural Networks (RNNs) (De la Pena Sarracén et al., 2018; Saksesi et al., 2018; Pitsilis et al., 2018b; Corazza et al., 2020a) and Convolutional Neural Networks (CNNs) (Gambäck and Sikdar, 2017), with Bidirectional Long Short-Term Memory (BiLSTM) gaining further popularity (Rajamanickam et al., 2020). These methods improved the ability to capture context and nuances in abusive language. Akhter et al. (2021) studied abusive language detection using four deep learning models (CNN, LSTM, BLSTM, and Convolutional Long Short-Term Memory (CLSTM)) and five machine learning models including NB, SVM, and LR. They compared the performance of the models and concluded that deep learning models perform significantly better than conventional machine learning models. Badjatiya et al. (2017) investigated the application of deep neural network architectures to detect hate speech and also showed that these models significantly outperform previous methods.

Recently, state-of-the-art methods for abusive language detection are primarily based on the Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019a). This is due to the ability of BERT to capture contextual and semantic information in text, making it more effective in recognizing abusive language. Moreover, BERT has been pre-trained on a large amount of text data in an unsupervised manner, which allows transfer learning. This has proven to be very effective for a wide range of natu-

ral language processing tasks, offering a significant increase in performance compared to training models from scratch. Mozafari et al. (2020a) investigate the ability of BERT to capture hateful context with transfer learning using fine-tuned methods. Their results indicate that this strategy can improve the results obtained by previous approaches. Similarly, Swamy et al. (2019) show that BERT has established itself as a state-of-the-art model through the comparison to other neural networks. Mnassri et al. (2022) focus on classifying hate speech using various models that integrate BERT and different neural network architectures. Additionally, they combine pairs of the first models to evaluate several ensemble techniques. The experiments show good results especially the ensemble models by stacking. Fortuna et al. (2021) also find that BERT tends to generalize the best with respect to a number of other models they experimented with, across several datasets. Beyond BERT, various transformer-based models, like RoBERTa (Liu et al., 2019a), XLNet (Yang et al., 2019) and Generative Pre-trained Transformer (GPT) (Radford et al., 2018), have been fine-tuned for abusive language detection. Like BERT, these models have demonstrated superior performance by capturing context and semantics (Mutanga et al., 2020; Shishah and Fajri, 2022). In our study, we focused on transformer-based models due to their strengths and the results reported in most recent works.

1.3 Offensive Language

Abusive language also includes cases of offensive language, but there are cases where offensive language is not abusive (see Figure 1.1). Offensive language can encompass a wide range of expressions that are considered socially inappropriate, impolite, or disrespectful. Still, they may not necessarily be intended to cause significant harm or be abusive (Pradhan et al., 2020; Risch et al., 2020). In fact, offensive language can sometimes be used unintentionally or can arise from various situations, including misunderstandings or cultural differences. In many cases, this can be a matter of individual perception, and what one person finds offensive, another might not.

Offensive language detection (OLD) and ALD share the goal of identifying harmful content, but ALD is often more complex because abusive language includes a wider range of harmful expressions, including threats, harassment, and personal attacks. However, similar techniques and models, such as traditional machine learning and deep learning models, have been used for both tasks.

Large-scale pre-trained language models like BERT (Alavi et al., 2021), ALBERT (Singh and Li, 2021) and GPT (Liu et al., 2020) have been fundamental in improving offensive language detection. Fine-tuning these models on labeled datasets specific to offensive content has shown remarkable results. Alavi et al. (2021) describe a study on using attention mask input in

BERT for detecting offensive content. The results indicate that models can be enhanced with this methodology. Singh and Li (2021) introduce a domain adaptation training procedure to ALBERT, such that it can use auxiliary data from source domains to improve the OLD performance in a target domain. The results on benchmark datasets show that this approach obtains state-of-the-art performances in most cases. Particularly, the approach benefits underrepresented classes.

1.4 Abusive Language Detection in Low-Resource Settings

Abusive language is highly context-dependent and varies across cultures and regions. Models trained on data from one cultural context may not generalize well to others, necessitating a deep understanding of cultural nuances to ensure accurate detection. Detecting abusive language in low-resource settings can be challenging due to the limited availability of labeled data and resources for training robust models capable of distinguishing between abusive and non-abusive content (Aluru et al., 2021; Bigoulaeva et al., 2021a). However, different strategies can be employed to address this issue, including multilingual model, active learning, and data augmentation (Nkemelu et al., 2022). Multilingual models are useful in transferring knowledge between languages with varying levels of prevalence (Ali et al., 2022; Ranasinghe and Zampieri, 2021). More generally, transfer learning allows to leveraging of pre-trained models on a related task, such as sentiment analysis or text classification. Thus, this strategy can be useful in low-resource settings. Bigoulaeva et al. (2021a) use cross-lingual transfer learning to leverage data from higher-resource languages. Their results indicate that it can be a useful method for achieving good performance on low-resource target languages. Active learning is a technique that focuses on selecting and labeling the most informative data to train a model. This allows a model to reach a certain level of performance with fewer labeled examples, which is especially useful when there are limited resources (Ahmed and Lin, 2022). Semi-supervised learning can also help capture semantic information with limited data. D'Sa et al. (2020) show that semi-supervised learning based on label propagation helps to improve hate speech classification in very low-resource scenarios.

Data augmentation is a technique where new data is created by applying various transformations or perturbations to existing data. Techniques such as paraphrasing, translating text to other languages and back, or introducing typos and variations, can help diversify the scarce data and improve the generalization of the models. Khullar et al. (2023) propose a data augmentation approach based on machine translation and contextual entity substitution to address the lack of data for hate speech detection. They generate synthetic data in a target language from hate speech examples in English as a high-

resource language. Their findings show that a model trained on the new synthetic data performs comparably to a model trained only on the samples available in the target domain. Azam et al. (2022) explore different data augmentation techniques for the improvement of hate speech detection. The techniques include synonym replacement, random insertion, random swapping, random deletion, text generation with a multilingual model, and word replacement via multilingual BERT. The results point out the ability of these strategies to improve performance. In this thesis, we analyze data augmentation strategies based on vicinal risk minimization. This is a principle that can be employed to ensure that the augmented data closely approximates the true risk distribution to enhance the quality and quantity of training data for machine learning models. We use two well-known strategies and propose a new one to evaluate how they affect abusive language detection in low-resource settings.

1.5 Motivation and Objectives

Recent studies in abusive language detection have paid much attention to the issue of bias (Dixon et al., 2018; Manerba and Tonelli, 2021; Nascimento et al., 2022; Cheng et al., 2022; Rizzi et al., 2023). Dixon et al. (2018) show the existence of bias between texts containing identity terms and a specific toxicity category. They attribute this issue to the disproportionate representation of texts containing certain terms in training data. Therefore, they use a technique of data augmentation to mitigate this bias. Mozafari et al. (2020b) also confirm the existence of bias in trained classifiers and introduce a bias mitigation mechanism to reduce the effect of bias during the fine-tuning of a model for hate speech detection. More recently, Nascimento et al. (2022) evaluate the distribution of bias-sensitive words in hateful tweets and the entire dataset to investigate disproportionate representations. The results show that the high disproportional distribution of the term *women* usually occurs in datasets composed of tweets related to sexism or misogyny categories. Rizzi et al. (2023) investigate a bias estimation technique to identify specific elements of misogynous memes that could lead to unfair models. They found that the list of terms with the highest bias score for the misogynous class is composed of tokens that are typically associated with some specific misogyny categories like *dishwasher* and *chick* for stereotype and *whore* for objectification. Furthermore, Manerba and Tonelli (2021) show how BERT-based classifiers can perform very poorly as regards fairness and bias, in particular on samples involving implicit stereotypes, expressions of hate towards minorities, and protected attributes such as race or sexual orientation. These works reveal the presence of bias and its possible impact on abusive language detection. We believe that this issue can increase when data is scarce, due to the limited availability of diverse and representative

labeled data to train robust models. Consequently, we aim to study model bias as a major problem for the detection of abusive language in low-resource settings.

Secondly, we have found evidence that graph-based models can have significant potential in low-resource settings. This kind of model facilitates the incorporation of small amounts of labeled data and a larger amount of unlabeled data. Thus, they offer various advantages for making the most of limited data by propagating information through a graph to make predictions. Yao et al. (2019a) propose a model based on a graph neural network for text classification. The result suggests that the improvement of graph-based models over state-of-the-art models becomes prominent as the percentage of training data becomes smaller.

Another observation in low-resource settings is that cross-lingual techniques can be highly effective (Adams et al., 2017; Yarmohammadi et al., 2019). These techniques allow for the transfer of knowledge and models from resource-rich languages to low-resource languages. Moreover, they enable zero-shot and few-shot learning, where models can make predictions in low-resource languages with minimal training examples. However, there are some considerations to take into account. Casula and Tonelli (2020) present an evaluation of hate speech detection in Italian using machine-translated data from English and investigate how the source data should follow the same annotation scheme and possibly class balance as the smaller data set in the target language. Another consideration is that within low-resource languages, there can be a significant linguistic diversity and cross-lingual models may struggle to accommodate all language variations. Additionally, low-resource languages may contain words that are not present in the source language. To overcome these limitations, it is essential to carefully evaluate the specific characteristics of low-resource languages and explore strategies that take advantage of existing linguistic resources. As mentioned in the previous section, data augmentation is a valuable technique that can be designed to create new data that are specifically tailored to low-resource languages. This can simulate language diversity and variations within a low-resource language and introduce synthetic out-of-vocabulary terms, helping models handle words not present in the source language.

Considering what we mentioned above, our research has the following objectives:

- To study the effects of the bias toward potential abusive keywords in models for abusive language detection in low-resource languages.
- To develop a graph-based strategy for text representation to address the lack of data faced by many languages in abusive language detection.
- To position the developed strategy and other graph-based models w.r.t. state-of-the-art strategies for abusive language detection in low-resource

languages.

- To evaluate the performance of abusive language detection in low-resource settings with data augmentation.

1.6 Research Questions

The above objectives can be divided into three groups according to the perspective from which the abusive language detection in low-resource settings is analyzed: 1) bias toward potential abusive keywords, 2) graph-based models, and 3) data augmentation. Considering these groups, the research questions we aim to answer in this thesis are:

Research question about bias in the models

- RQ1: *Could bias toward potential abusive keywords in the models affect the performance of abusive language detection in low-resource settings?* We are interested in studying the bias that models tend to learn toward abusive keywords in low-resource settings. In particular, we evaluate how this bias affects the performance of abusive language detection and whether bias mitigation is worthwhile. For this purpose, we first propose methods to extract potentially abusive keywords and then evaluate the bias toward these keywords in models.

Research question about graph-based models

- RQ2: *What is the contribution of models based on graph neural networks for abusive language detection in low-resource settings?* Aiming to study the potential of this kind of model, we evaluate the use of graph neural networks in two directions: 1) to detect abusive language and 2) to generate appropriate textual representations capable of distinguishing abusive language. Then, we compare their performance with state-of-the-art models in low-resource settings.

Research question about data augmentation

- RQ3: *What is the contribution of data augmentation for abusive language detection in low-resource settings?* In this thesis, we also aim to investigate the potential of data augmentation to improve abusive language detection in the particular case of low-resource settings. To do so, we employ strategies based on transformations and combinations of the original data.

1.7 Contributions of this Thesis

In this section, we summarize the main contributions of this thesis.

We study the performance of **abusive language detection in low-resource settings** from three perspectives. We first propose and evaluate two methods to extract potential abusive keywords from data collections. Next, we analyze the salient words to which pre-trained transformer-based models pay more attention for abusive detection. The results show that over 50% of these salient words are not abusive and that there is a higher number of abusive words among our extracted keywords. However, we examine **the bias toward the keywords in models** and show that the models do indeed appear to be biased. Additionally, we prove that some mitigation strategies can reduce this bias and improve the performance of the models.

Another contribution of this thesis is made in the field of **models based on graph neural networks**. We propose an approach using graph auto-encoders to generate embeddings from text. To do so, we represent the texts as nodes of a graph with the hypothesis that the nodes with similar characteristics will be close in the generated embedding space. The results illustrate an evident separation between the embeddings of abusive and non-abusive texts. We employ these embeddings for abusive language detection and show that our method produces competitive results on small datasets. On the other hand, we use a model based on convolutional graph neural networks and show that our model is more stable than other state-of-the-art deep learning models with limited data.

Moreover, we use **data augmentation to improve cross-lingual abusive language detection** in low-resource settings. We employ a dataset that contains extensive data in English and involves limited data in seven languages typologically distinct from English. This allows us to study cross-lingual abusive language detection for different languages in settings with little data. We analyze two existing data augmentation methods based on the principle of vicinal risk minimization. Furthermore, we propose a variant of one of these methods that generates data from a novel interpolation of pairs of instances. The results reveal that the three methods can enhance cross-lingual abusive language detection. Specifically, we observe that our method improves significantly in multidomain and multilingual environments.

Finally, we help build a **dataset for profiling haters** in Twitter, both in English and Spanish. This dataset was used in the shared task organized at PAN in 2021. We will briefly describe it at the end of this manuscript.

1.8 Structure of the Thesis

This PhD thesis is presented as a compendium of research articles (from Chapter 2 to Chapter 6) that were published during the study phase of the

PhD candidate. Since our study is conducted from various perspectives, we split this thesis into 3 main parts, followed by a fourth part with our findings that includes a general discussion of the results together with some new experiments and our conclusions. Next, we briefly overview the content of the parts:

Part I: Keyword Extraction and Bias Analysis

Chapter 2: Offensive Keyword Extraction based on The Attention Mechanism of BERT and The Eigenvector Centrality using a Graph Representation.

Chapter 3: Systematic Keyword and Bias Analyses in Hate Speech Detection.

In this part, we present two research papers published in the Personal and Ubiquitous Computing and the Information Processing & Management (IPM) journals, respectively. The main focus of this part is to extract keywords from datasets with abusive texts and investigate how the bias toward these potentially abusive keywords can affect abusive language detection. The first work proposes a method for keyword extraction that leverages the abilities of the multi-head self-attention mechanism of BERT to assign attention values among pairs of words in a context. The attention values are used to represent the edges of a graph where the nodes are the words. Then, we use the eigenvector centrality algorithm on the generated graph to select the high-scoring nodes as the keywords. We evaluate this method in offensive language detection. The results show that our method can detect keywords from datasets that can influence the performance of offense detection. In the second work, we study the relationship between a set of keywords extracted from three datasets and the salient words of two transformer-based models pre-trained in hate speech detection. For keyword extraction, we propose a statistics-based method to identify very frequent keywords in hateful texts that are less frequent in other texts. We show that this method manages to extract a large number of hateful words, unlike other leading keyword extraction methods. We also observe that there is little similarity between the extracted keywords and the words that the transformer-based models pay the most attention to. However, we show that there is a bias toward the extracted keywords in the models and that this bias can be reduced to improve the performance of the models.

Part II: Graph-Based Exploration

Chapter 4: Unsupervised Embeddings with Graph Auto-Encoders for Multi-Domain and Multilingual Hate Speech Detection.

Chapter 5: Convolutional Graph Neural Networks for Hate Speech Detection in Data-Poor Settings.

This part introduces two works published in the Language Resources and Evaluation Conference (LREC) and the International Conference on Applications of Natural Language to Information (NLDB), respectively. These research works consider graph neural networks for hate speech detection. In the first one, we propose a graph auto-encoder framework to obtain a latent representation from an initial vector representation of texts. Then, embeddings are extracted from this space. We used this framework for hate speech detection by using the embeddings as input of a classifier. The analysis shows that our strategy outperforms state-of-the-art models in hate speech detection when the availability of data is notably scarce. In the second work, we study a model based on convolutional graph neural networks to address hate speech detection in scenarios with little data. Similar to the first work, the results of this work show that this model is robust in small datasets, outperforming state-of-the-art models in low-resource settings.

Part III: Data Augmentation

Chapter 6: Vicinal Risk Minimization for Few-Shot Cross-lingual Transfer in Abusive Language Detection.

This part is composed of our work published in the conference on Empirical Methods in Natural Language Processing (EMNLP). This publication presents a study of three techniques to improve few-shot cross-lingual transfer learning in abusive language detection. These techniques are focused on the data-level approach to deal with the problem of data scarcity that can lead to a high estimation error in few-shot learning. Specifically, we use vicinal risk minimization techniques to increase the data in the vicinity of the few-shot samples. We explore two existing techniques and propose a variant of one of them. The results show the effectiveness of these techniques to improve cross-lingual abusive language detection in different domains and languages. Particularly, we observe that our variant outperforms the other strategies in a multidomain scenario for all target languages.

Part IV: Summary

Chapter 7: Discussion of the Results.

Chapter 8: Conclusions and Future Work.

In this part, we discuss the results obtained in the previous parts, we answer our research questions, and we draw the main conclusions of

this thesis. In Chapter 7, we complement our study with further experiments to gain additional insights. First, we conduct further experiments to compare our two methods for keyword extraction. We also extend the analysis of the bias toward the keywords extracted with both methods for particular low-resource settings (Part I). Later, we extend our experiments with an ablation analysis of different types of graphs neural networks for abusive language detection in low-resource settings (Part II). Finally, we evaluate the EDA techniques (Wei and Zou, 2019) and compare their influence on abusive language detection in low-resource settings w.r.t. a method based on vicinal risk minimization that we previously studied (Part III). At the end, we add extra analyses for the study of potential spreaders of hate speech. We present our work published in the Conference and Labs of the Evaluation Forum (CLEF), where we help build a dataset for profiling haters in Twitter, both in English and Spanish (Rangel et al., 2021). This is an overview of the shared task we helped organize at PAN 2021. Additionally, we present preliminary results of a study of the network features of potential spreaders of hatred. In Chapter 8, we list our contributions that were disseminated as publications and comment on the open research lines for possible future works.

Part I

Keyword Extraction and Bias Analysis

This first part presents the study about the bias toward potential abusive keywords in the models and how it can affect the performance of abusive language detection. In Chapter 2, we propose a method for potentially abusive keywords, that leverages the ability of the multi-head self-attention mechanism of BERT to capture contextual and semantic information in texts. In Chapter 3, we present a simpler method for potentially abusive keywords. This is a statistics-based method to identify prevalent keywords in hateful texts that are less common in other texts. We use this method to evaluate the bias toward abusive keywords and how it may affect the performance of abusive language detection.

Chapter 2

Offensive Keyword Extraction based on The Attention Mechanism of BERT and The Eigenvector Centrality using a Graph Representation

Published in:

- **De la Peña Sarracén, G.L.** and Rosso, P. (2023). Offensive Keyword Extraction Based on the Attention Mechanism of BERT and the Eigenvector Centrality using a Graph Representation. Personal and Ubiquitous Computing, 27(1), (pp. 45-57).
  (Impact Factor: 3.406, Q2)

Abstract. The proliferation of harmful content on social media affects a large part of the user community. Therefore, several approaches have emerged to control this phenomenon automatically. However, this is still a quite challenging task. In this paper, we explore offensive language as a particular case of harmful content and focus our study on the analysis of keywords in available datasets composed of offensive tweets. Thus, we aim to identify relevant words in those datasets and analyze how they can affect model learning. For keyword extraction, we propose an unsupervised hybrid approach that combines the multi-head self-attention of BERT and reasoning on a word graph. The attention mechanism allows us to capture relationships among words in a context, while a language model is learned. Then, the relationships are used to generate a graph from which we identify the most relevant words by using the eigenvector centrality. Experiments were performed by means of two mechanisms. On the one hand, we used an information retrieval system to evaluate the impact of the keywords in recovering offensive tweets from a dataset. On the other hand, we evaluated a keyword-based model for offensive language detection. Results highlight some points to consider when training models with available datasets.

2.1 Introduction

Automatic Keyword Extraction (AKE) is a technique of text analysis that consists of automatically extracting the most relevant words in a text. In general, it can be used to identify topics in a text, summarize its content, index data, or generate tag clouds with the most representative words. In this paper, we aim to apply the idea of AKE to obtain words that best describe offensive language as a particular case of harmful content. Therefore, in the scope of this work, we define keywords as words that are relevant to identify offensive content. There are different approaches and available tools for keyword extraction, but they have been designed with a general purpose. Those methods extract keywords from texts with certain criteria, such as frequency. In this sense, we have identified some limitations to extracting keywords of our interest, since we cannot make a distinction between offensive and non−offensive texts. This is a problem because a relevant word in non−offensive texts should not be selected as a keyword. A shallow solution would be to analyze only offensive texts, but relevant words in offensive texts that are also relevant in non−offensive texts should not be selected as keywords. Therefore, we need to solve the keyword extraction problem for our particular case, i.e. how to select words that are relevant in offensive tweets and at the same time very little relevant in non−offensive tweets.

Our methodology consists of three stages: i) weighting pairs of words by their relationship in the tweets, ii) building a graph where the vertices are words and the edges are weighted with the values obtained in the previous

stage, and iii) reasoning on the graph to identify the most relevant words. In the first stage, we consider the class of the tweet from which each word pair is taken. If the tweet is non−offensive, the corresponding weight is penalized. In that way, we address the aforementioned limitation. In order to obtain the weights for each word pair, the method we propose is based on the multi-head self-attention mechanism of BERT Devlin et al. (2019b). Although other strategies can be adapted, we are motivated by the state-of-the-art results that BERT has obtained in several tasks, including offensive language detection and related tasks. The attention mechanism is precisely one of the strengths of BERT. This mechanism allows capturing relationships among words in a context, while a language model is learned.

The main contributions of our work are the following:

i. We propose a method for extracting keywords from datasets composed of offensive and non−offensive tweets. The method distinguishes between tweets from different classes.

ii. We use an unsupervised method that does not need annotated datasets for automatic keyword extraction, that is, datasets with gold keywords of the texts. Instead, our method only uses a set of tweets tagged as offensive or non−offensive.

iii. We present a way to exploit the multi-head self-attention mechanism of BERT to weight word pairs from tweets.

iv. We use a method to represent the words and their relationships in a graph for extracting relevant words by using the eigenvector centrality.

v. We analyze the extracted keywords and evaluate their impact on offensive language identification. As results, we give insights about points to consider when training models with available datasets.

An important advantage of our proposal lies in the facility to be adapted to related phenomena, such as hate speech, misogyny, and sexism Basile et al. (2019a); Fersini et al. (2018). These are similar phenomena with common characteristics and similar datasets. There are no clear boundaries among them, although each one has particular characteristics Poletto et al. (2021). For instance, offensiveness includes rude or vulgar language that does not represent hatred. However, our method can be applied to hate speech or other related tasks by varying the type of datasets and fixing the hyper-parameters of the model.

The rest of this paper is organized as follows. Section 2.2 summarizes the related work and Section 2.3 presents the problem formally. Our keyword extraction method is proposed in Section 3.5.1. Section 2.5 describes the experiments and Section 2.6 presents a discussion of the results. Finally, Section 2.7 concludes the paper.

2.2 Related Work

This section presents a summary of some widely used keyword extraction techniques. Then, we provide a brief overview of offensive language detection, both in the sense of keyword-based strategies and in the sense of BERT-based approaches. Finally, this section introduces a synopsis of textual graph representation, with a focus on techniques used for keyword extraction.

2.2.1 Automatic Keyword Extraction.

AKE has been developed with different approaches Kaur and Gupta (2010); Hasan and Ng (2014); Nasar et al. (2019). Using statistics is one of the simplest mechanisms for selecting keywords within a text. This approach includes well-known techniques such as word frequency, term frequency-inverse document frequency (TF-IDF), word collocations, and co-occurrences. Roughly, they consist of listing the words according to some criterion and selecting the top ones. For instance, the word frequency technique looks for the most common words occurring within a collection of texts. The advantage of this kind of approach is that they do not need training data in order to extract keywords. However, they may ignore some relevant words that are mentioned only once but are indeed relevant. Linguistic approach is another type of mechanism which considers linguistic information about texts. Some strategies involve morphological, syntactic or semantic information about the words, such as part-of-speech or the relations between words in a dependency grammar. This kind of information provides an important tool for keyword extraction Hu and Wu (2006). Moreover, AKE is also addressed by employing machine learning techniques which are usually supervised approaches. A well-known method that transforms AKE into a binary classification task was presented in Witten et al. (2005). Other methods include models such as Support Vector Machines, Conditional Random Fields, and Deep Learning strategies Firoozeh et al. (2020). The authors of Sahrawat et al. (2020) proposed a keyphrase extraction as a sequence labeling task. They used BERT to obtain contextual embeddings, although they required manually annotated keyphrases.

Some of the previous techniques are combined in a hybrid mechanism in order to obtain better results. In this paper, we exploit this idea.

2.2.2 Keywords in Offensive Language Detection.

Regarding offensive language, keywords have been mainly used to build datasets. The data collection for the construction of the Offensive Language Identification Dataset (OLID) Zampieri et al. (2019a), used in OffensEval 2019 shared task Zampieri et al. (2019c), was based on searching for keywords and constructions that are often included in offensive messages.

Initially, a set of words was used to collect tweets, and then some keywords that were not frequent in offensive content were excluded during the trial annotation. Similarly, for the dataset of the HASOC track Mandl et al. (2019a) the data were acquired using hashtags and keywords with offensive content. Here, we aim to use a keyword-based technique to evaluate our keyword extraction method by analyzing how these keywords can influence the detection of offensive language. However, it is important to consider that keyword-based strategies have been found to be biased and problematic for offensive language detection and related tasks. They overlook cases in which no profane nor offensive words are used but the text actually conveys an offense. Moreover, these strategies can cause non−offensive texts, that contain some keywords, to be misclassified. That is why we only employ a keyword-based strategy to evaluate the extracted keywords, not to improve the offensive language detection.

2.2.3 BERT for Offensive Language Detection.

Most of the strategies used for offensive language detection are based on traditional machine learning and deep learning Pitsilis et al. (2018a); Uglow et al. (2019); Wani et al. (2019); Vashistha and Zubiaga (2021). Among them, BERT and other transformers-based models are state-of-the-art in the latest results, especially in shared tasks such as OffensEval 2020 Zampieri et al. (2020a). The best team used RoBERTa-large, which was fine-tuned on the dataset by using the masked language modeling objective Wiedemann et al. (2020). The second team used an ensemble that combined XLM-RoBERTa-base and XLM-RoBERTa-large Wang et al. (2020a). In general, the top teams used BERT, RoBERTa or XLM-RoBERTa Dai et al. (2020); Casula et al. (2020); De la Peña Sarracén et al. (2020).

2.2.4 Text Representation based on Graph.

A graph-based text representation allows the exploration of the relationships and structural information in a text very effectively. Then, AKE is often performed by selecting vertices or groups of vertices with a search on a graph. TextRank Mihalcea and Tarau (2004) is a model widely used in this type of approach. It is derived from PageRank Brin and Page (1998) which scores each vertex taking into account the importance of its neighborhood. Ao et al. (2020) recently proposed a new keyword extraction algorithm based on TextRank. However, Boudin (2013) compares various centrality measures for graph-based AKE, and the experiments on datasets in English and French show that the simple degree centrality achieves results comparable to TextRank. We test our proposal with different types of degree centralities to select vertices from a word graph. Finally, we use the eigenvector centrality as it is explained later.

2.3 The Problem

The problem we address in this work can be formally described as follows: Let O and N be two sets of offensive and non−offensive tweets respectively, for which the following holds: $\{O \cap N = \emptyset\}$. Let W be the set of words from $\{O \cup N\}$. The problem is to identify a set of words $K \subset W$ such that each $k \in K$ is in the top ranking of the words highly relevant in O and little relevant in N. Modeling this problem we represent W in a graph with weighted edges from which we rank the words. In the graph, each vertex is a word of W and each edge (w_1, w_2), $w_1, w_2 \in W$, indicates the weight between the words w_1 and w_2. We aim to calculate the weights considering the context of the words in each tweet, as well as whether the tweet is offensive or not. For that, BERT can be suitable since its self-attention mechanism analyzes each word looking at other words in the context. Thus, the research questions we address in this work are:

> **RQ1:** How can we leverage the attention mechanism of BERT to weight pairs of words in the context of a text?
>
> **RQ2:** How can we effectively extract words that are relevant in offensive tweets and little relevant in non−offensive tweets from a dataset?

2.4 Keyword Extraction based on BERT

In this section, we first introduce our methodology for keyword extraction from a dataset. Then, we explain in detail each of the stages of the methodology and comment on how our method can be extended to deal with longer texts.

Figure 2.1 illustrates the elements on which our proposal is based on. The methodology is composed of three stages. The first stage consists of obtaining a relationship between words in the dataset. In this sense, we obtain a weight for each pair of words by relying on BERT and specifically on the multi-head self-attention mechanism. In the second stage, we generate a graph where the vertices are the words from the datasets, and the edges are weighted according to the relationship between words. The weight of each edge is updated every time the corresponding word pair appears in a text. For each text, the weight calculated for each pair is added to the weight of the corresponding edge in the graph if the text is offensive, and is subtracted otherwise. In this way, we penalize the non−offensive texts. Finally, we obtain a keyword list in the third stage by identifying the most relevant vertices in the graph with the eigenvector centrality.

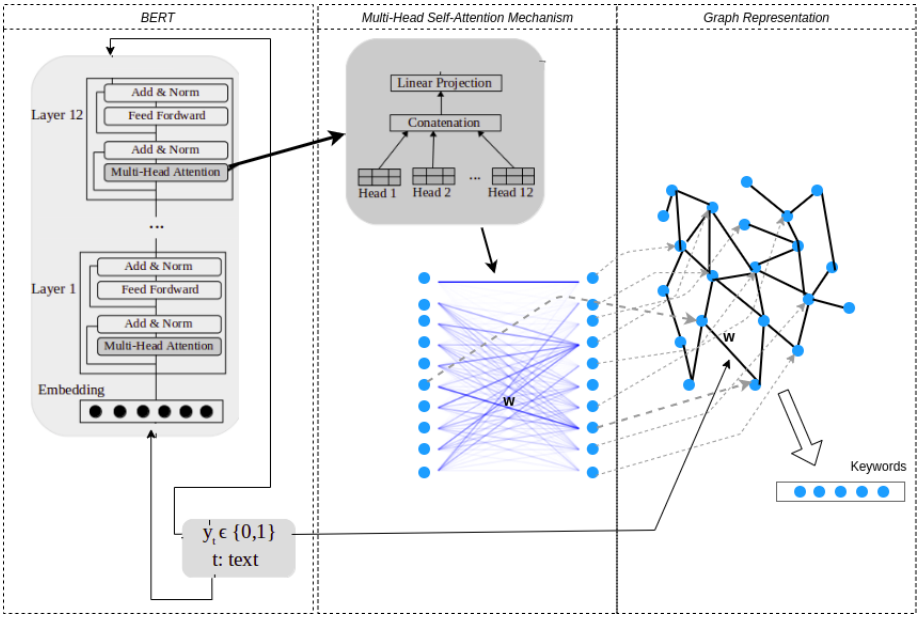


Figure 2.1: Illustration of our keyword extraction model.

2.4.1 Attention from BERT

In text classification, some parts of the input can often be more relevant compared to others. Attention mechanisms incorporate the notion of relevance by allowing a model to dynamically pay attention to only certain parts of the input. The assumption is that the higher attention weights correlate with how relevant a specific region of input is Chaudhari et al. (2019). The following example shows three texts from the OffensEval 2019 dataset Zampieri et al. (2019b). An attention mechanism can help to classify the second example as offensive and identify *fu**ing* in that text as more meaningful to determine offensive.

> He is such a good ad for conservatives. |||
> She is fu**ing delusional. ||| I quite enjoy these tweets you are liking.

We use this idea to obtain the relationship between words in the texts of a dataset. Concretely, we leverage the multi-head self-attention mechanism from BERT.

The first step is fine-tuning a pre-trained BERT model for the offensive language detection task as the first part of Figure 2.1 shows. For this, we use a dataset with offensive and non−offensive texts (1 or 0 respectively). Thus, while the model is trained, the parameters of the attention mechanism in each layer of the BERT model are updated according to the data.

In this step of fine-tuning, the input is text and a softmax is added on the top of the last layer of BERT. We only retain non−padding tokens to feed the softmax by multiplying the output with a mask. Cross entropy is used as the loss function (2.1), where y_i is the true classification of a text i, and $\hat{y_i}$ is its predicted value.

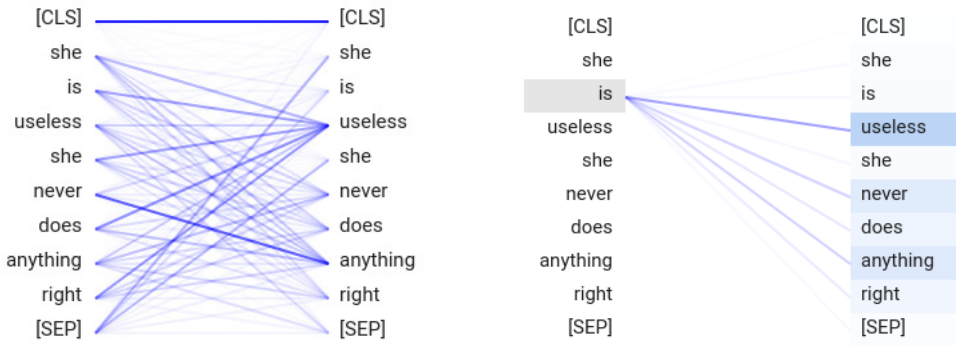
$$L = -\Sigma_i \; y_i * log(\hat{y_i}) \tag{2.1}$$

Now, it is important to understand what happens inside BERT. With the self-attention mechanism, each position t in the input (token) is processed by looking at other positions to obtain a good encoding for t. Thus, this mechanism is used to capture related and important words. Basically, self-attention creates three vectors for each word (token) by multiplying the input embedding by three matrices of parameters which are fitted in the training process. The vectors are known as Query (q), Key (k) and Value (v), and the matrices of parameters are W^q, W^k, W^v respectively. Then, a score is calculated for each word against each of the other words. The calculation is done by a normalized dot product of the q vector of the current token t and the k vector of the other tokens. As a result, a vector ($Attention_t$) is obtained for each token t where each component i determines how much focus to put on the position i of the input. Next, this vector is multiplied by the v vector to keep the values of the original token t. Equation 2.2 recaps this process for the matrix calculation for all words at once Vaswani et al. (2017a)). Where d_k is the dimension of q, k, v and Q, K and V are the matrix representations respectively for the text.

$$\begin{aligned} Attention(Q, K) &= softmax(\frac{QK^T}{\sqrt{d_k}}) \\ Head(Q, K, V) &= Attention(Q, K)V \end{aligned} \tag{2.2}$$

Figure 2.2 shows the visualization of the weights (Attention) between pairs of words for the text *"she is useless she never does anything right"* using the pre-trained BERT_base model. This is an offensive text taken from the dataset of the OffensEval 2019 shared task. On the left (Figure 2.2a), we can see the higher weights with a more intense color, which indicates relevant parts of the text for each term. For example, the word *"anything"* seems to be quite relevant when the word *"never"* is analyzed. On the right, Figure 2.2b shows the particular case of the attention values (weights) between the word *"is"* and the other words in the text.

Furthermore, BERT incorporates multi-head attention which expands the ability to focus on different positions by giving the attention layer multiple representation subspaces from multiple sets of $Q/K/V$ weight matrices. Thus, there are $L \cdot H$ self-attention patterns in the model, where L is the number of layers in the model and H is the number of heads in each layer.



(a) All attention values

(b) Attention from the word 'is'

Figure 2.2: Attention visualization for a sample text.

Figure 2.3 shows 12 patterns for the text analyzed previously. They correspond to the three last layers of the model, which are usually the most used layers for obtaining the output. The first row corresponds to layer #12 (the last one of the model), the second row corresponds to layer #11 and the third row corresponds to layer #10. For each layer, we show the first four heads from left to right. That is, the first column corresponds to head #1. We can see interesting patterns in these layers. For example, the fourth head in layer #12 represents a pattern where the attention for each word is focused on the previous word in the text. This makes sense because adjacent words are often relevant for predicting the next word. On the other hand, the second head of the same layer matches the one shown in Figure 2.2. Other heads, like the fourth one in layer #11, represent a null pattern where almost all the attention is focused on the token CLS. This probably indicates that those heads did not find a linguistic phenomenon. However, with the multi-head mechanism different strategies are combined to analyze the relationship between words.

Once the BERT parameters are learned in the fine-tuning step, the texts feed the model again to obtain the new vectors for each token. These vectors are obtained by the condensation of the pattern of each head in each BERT layer as Equation 2.3, where h is the number of heads in the layers. i.e. the output of all heads is first concatenated and then projected to a new space by multiplying by a matrix W, which is also fitted in the training step.

$$\text{MHA} = Concat(Head_1, \ldots, Head_h)W, \quad i = \overline{1,h} \tag{2.3}$$

For our method, we use only the attention values Attention^i of each head i in a layer. Specifically, we use the sum of the values of all the heads as Equation 2.4.

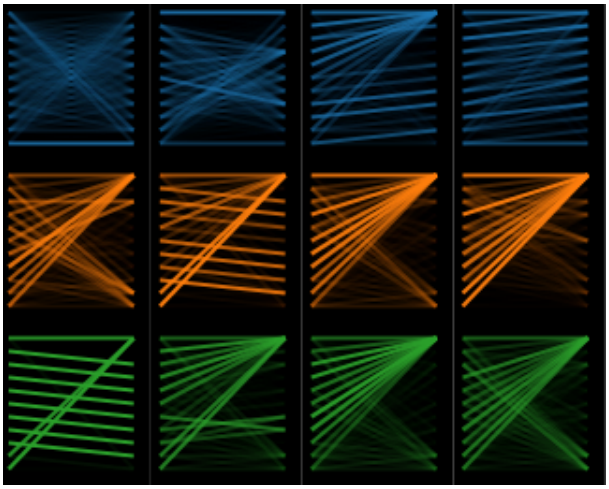


Figure 2.3: Visualization of 12 heads of the attention mechanism.

$$\mathrm{A} = \sum_{1}^{h} Attention^{i} \tag{2.4}$$

As a result, we obtain a matrix $A \in \mathcal{M}_{|\mathcal{T}|}(\mathbb{R})$ with the relationship of relevance (attention) between pairs of words. Where $\mathcal{T}$ is the set of the words, $|\cdot|$ denotes the size of a set and $\mathcal{M}_n(\mathbb{R})$ represents the set of square matrices of size n with inputs in the field $\mathbb{R}$.

We consider a word as terms that are not stopwords and that represent English words. The special tokens CLS and SEP are not selected into $\mathcal{T}$. They do not have a meaning in the human language, therefore they cannot be selected as keywords. Moreover, these tokens tend to get high scores due to the null patterns in the attention mechanisms. Furthermore, the tokens starting with the characters $'\#\#'$ are not considered as words because they only represent parts of words.

Earlier, we have seen how to get the matrix A for a set of text, now we explain how to update this matrix with new texts. Let $\mathcal{T}_t = \{w : w \in t\}$ be a set of words for a sample text t from the dataset, then $\mathcal{T}$ is modified as $\mathcal{T} = \mathcal{T} \cup \mathcal{T}_t$. Moreover, let A_t be the attention matrix obtained given t. The attention matrix A is updated with t by a function $\theta : \{0,1\} \times \mathcal{M}_{|\mathcal{T}_t|}(\mathbb{R}) \times \mathcal{M}_{|\mathcal{T}|}(\mathbb{R}) \to \mathcal{M}_{|\mathcal{T}_t \cup \mathcal{T}|}(\mathbb{R})$ as Equation 2.5 indicates, where y_t is the label of t and ϵ is a parameter to control the change in A given t.

$$\begin{aligned} &A'_t = (2 \cdot y_t - 1) \cdot \epsilon \cdot A_t \\ &\theta(y_t, A_t, A) = \begin{cases} (A)_{i,j} + (A'_t)_{i,j} & \textbf{if } \exists (A)_{i,j} \\ add((A'_t)_{i,j}) & \textbf{if } \nexists (A)_{i,j} \end{cases} \\ &i, j = \overline{1, |\mathcal{T}_t|} \end{aligned} \tag{2.5}$$

The function $add(\cdot)$ incorporates a raw and a column in A for each word in $\mathcal{W}_t = \{w : w \in \mathcal{T}_t \wedge w \notin \mathcal{T}\}$. For each pair of words that is not in A_t, (a pair (w_1, w_2) such that $w_1 \in \mathcal{W}_t$ and $w_2 \in \{w : w \in A \wedge w \notin \mathcal{W}_t\}$ or vice versa) the function puts 0 in the corresponding cell of A, otherwise the value into A'_t for the pair is taken.

Notice that for an offensive text (label 1) the new attention value of each word pair is added according to the magnitude of ϵ. In contrast, the new attention values for a non−offensive text (label 0) are subtracted. In this way we control the score of each word pair, penalizing those extracted from non−offensive texts.

2.4.2 Graph Representation

We use a graph as a mathematical model to represent the relation among pairs of words from the matrix A of attention values. Formally, we build a directed graph $G = (V, E_A)$ as a set of vertices V that matches with the set $\mathcal{T}$ and a set of edges $E_A \subseteq \{(w_1, w_2) : (w_1, w_2) \in \mathcal{T} \times \mathcal{T}\}$. Also, we define a function $val : E_A \to \mathbb{R}$ such as $val(w_1, w_2) = (A)_{w_1,w_2}$ to assign a weight to each edge. Thus, the vertices represent words and the edges represent the relation between pairs of words (attention values).

In the construction of the graph we define the function $\Phi : \mathcal{M}_{|\mathcal{T}|}(\mathbb{R}) \to E_A$ such that $\Phi(A) = \{(w_1, w_2) : (w_1, w_2) \in \mathcal{T} \times \mathcal{T}, val(w_1, w_2) > 0\}$. Hence, we only represent the relationships between words with a positive attention value. Furthermore, we use this function Φ to update the graph G once new texts are incorporated into the analysis.

2.4.3 Keyword Extraction from Graph

The candidate keyword list $\Gamma(G)$ is obtained by selecting the words associated with the most relevant vertices in the graph G. To carry it out, we rank the vertices using a measure that assigns a score to each vertex considering the weights of the edges in G.

The eigenvector centrality (EC) is the measure we use for the ranking Newman (2008). EC measures the influence of a vertex in a graph scoring each vertex as a function of the centralities of its neighbors as Equation 2.6. Where λ is a constant called eigenvalue and $\mathcal{N}(w) = \{w_j : (w, w_j) \in E_A\}$ is the neighborhood of the vertex w.

$$EC(w_i) = \frac{1}{\lambda} \sum_{w_j \in \mathcal{N}(w_i)} val(w_i, w_j) \cdot EC(w_j) \tag{2.6}$$

Alternatively, we use some centrality measures based on the degree of the vertices. Unlike EC, they do not take into account the weight of the edges. Instead, they use the information of the neighborhood of each vertex. These measures make sense since vertices with a high degree indicate relevant

words. However, the comparison among all the strategies allows us to analyze the importance of considering the weight of edges estimated from the attention mechanism. We used the following alternative centrality measures:

- Degree (DC): $DC(v)$ is calculated by dividing the amount of vertices that v is connected to, by the maximum possible degree in G.
- In degree (IC): $IC(v)$ is calculated by dividing the amount of incoming edges at v, by the maximum possible degree in G.
- Out degree (OC): $OC(v)$ is calculated by dividing the amount of outgoing edges from v, by the maximum possible degree in G.

Selection criterion based on Part-of-Speech: Once the ranking of vertices is obtained, we select as keywords those candidates that are one of the parts of speech: noun, adjective, or verb. This idea has been developed in some approaches where usually only nouns and adjectives are taken into account Kathait et al. (2017). We consider that verbs also can express offenses, like for instance *'fu**'*.

2.4.4 Analyzing longer texts

Notice that our method is designed to work with tweets, that are small texts. However, our proposal can be extended based on the way that BERT can be generalized to deal with longer texts. BERT can handle input sequences up to 512 tokens long. However, some strategies can be adopted. Among them, the strategy presented in Pappagari et al. (2019) is a good one. The idea is to divide each large text into segments and feed BERT with each of them. The pooled output and the logits are used as representations for each segment. Then, they are passed along to either an LSTM recurrent neural network model (RoBERT variant) or a lightweight transformer (ToBERT variant). Thus, our method can be used in the same way.

2.5 Experiments

This section presents our experiments and results. We first describe the datasets and the evaluation methods we used. Then, we detail the experimental setup and other models used to compare our proposal. Finally, we present the results and analysis.

Datasets. We used the datasets released for the OffensEval shared task in its two editions: OffensEval 2019 Zampieri et al. (2019c) and OffensEval 2020 Zampieri et al. (2020a). Both tasks focus on the identification of offensive language in tweets.

OffensEval 2019 (OFF19): This is a dataset which contains English tweets. The labels are organized in a hierarchical tag set Zampieri et al. (2019a). We used the tags at the level of the binary classification, such that we worked with the two labels offensive (4640 tweets) and non−offensive (9460 tweets).

OffensEval 2020 (OFF20): This is a multilingual dataset with tweets in five different languages Rosenthal et al. (2021). We randomly selected 30000 tweets from the nine million of English tweets, keeping the proportion between offensive (4782 selected) and non−offensive (25218 selected) tweets in the original set. In this dataset, we had access to the average of the confidence in the offensive class of several supervised models. In order to work with binary labels, we transformed the average values by considering any tweet with a confidence average higher than 0.5 as offensive, otherwise, we considered it as non−offensive.

Evaluation Methods. An AKE model is usually evaluated with a set of 'gold' keywords that constitute the references. The idea is to compare these references with the extracted keyword list. In this case, we do not have this kind of labeled dataset. Therefore, we use an extrinsic evaluation method with two relevant applications. Thus, we evaluate the keywords through their impact on the other two tasks: Information Retrieval and Offensive Language Detection.

Information Retrieval (IR): We evaluated the keywords by searching tweets in a collection of index tweets. The idea is to analyze how suitable the keywords are to extract offensive tweets. In this regard, we defined an IR task as finding offensive tweets from a large set of tweets given a keyword list as a query. We created a model which indexes the documents (tweets) by concepts. Then, we use the BM25 algorithm Robertson et al. (1995) to retrieve tweets given a set of keywords. As the evaluation measure, we use the Precision@K (P@K) and F@K which compute the precision and F1-score respectively over the top-K retrieved tweets Büttcher et al. (2016). Here, a true positive is a retrieved tweet that is offensive.

Offensive Language Detection (OLD): In order to study how the keywords can impact offensive language detection we evaluated a keyword-based model. We have to point out that keyword-based approaches can be very misleading for detecting offensive language since they overlook many cases in which no offensive words are used but the text still conveys extremely offensive content Wiegand et al. (2019). In this sense, our aim is not to improve the state-of-the-art in this task but rather to analyze how relevant words in the datasets used to train models can influence the task. We used an LSTM recurrent neural network and its output is concatenated with a vector of similarity values between the input (tweet) and each of the keywords. Finally,

the classification is obtained with a softmax[1]. As the evaluation measure, we used the macro-averaged F1-score (F1).

Experimental Setting. In order to obtain the keywords, we used the pre-trained BERT_base[2] model which has 12 layers and 12 heads per layer. Moreover, we used *NLTK*[3] to obtain the Part-of-Speech tags of each word, since our method only considers nouns, adjectives, and verbs, as we explained before. We trained BERT for 4 epochs with minibatches of size 16. The optimizer we used was Adam Kingma and Ba (2015a) and we set the learning to 5e-5. The parameter ϵ of our proposal was fixed to 0.1 after a parameter setting. For OLD we relied on the stratified 5-fold cross-validation technique and the paired permutation test with p-value < 0.05 for the analysis of statistical significance. For IR we used the t-test with p-value < 0.05 For the reproducibility of the experiments, we set the random seed to 5.

Furthermore, in our experiments, we only used the datasets introduced before. Both OFF19 and OFF20 are composed of tweets, but they are different in the sense of data collection and annotation. OFF19 was collected by retrieving tweets with keywords that are common in offensive texts. In contrast, OFF20 was collected by searching the 20 most common English stopwords to ensure a variety of random tweets. Moreover, semi-supervised labeling was used for OFF20. Thus, OFF19 can be more biased towards the keywords used in the collection, and OFF20 towards the way it was annotated. In this sense we followed the two possible cross-evaluations: i) obtain the keywords from OFF19 and then, use OFF20 to evaluate the OLD and IR tasks considering the keywords from OFF19 (OFF19-OFF20) and ii) vice versa (OFF20-OFF19).

In addition, we carried out two in-domain evaluations OFF19-OFF19 and OFF20-OFF20 to study how the bias in the datasets can influence the extraction of keywords. For example, we suppose that the keywords from OFF19 should reflect the bias in this dataset. Therefore, the use of these keywords should affect the detection of offenses.

Benchmarks. As part of our experiments, we used some well-known models to extract keywords. The objective is to analyze the relevance of our proposal by comparing it with methods that are not tailored to our concern. We adapted the term frequency – Inverse document frequency (TFIDF) by calculating TF in the offensive tweets and IDF in the non−offensive tweets. Moreover, we used TF, TEXTRANK (TRANK), RAKE Berry and Kogan (2010), and YAKE[4]. This last method reports state-of-the-art results. For

[1]We also use this model without concatenating the vectors with the keyword information. We refer to this model as the base model in this paper.

[2]https://tfhub.dev/google/bert_uncased_L-12_H-768_A-12/1

[3]https://www.nltk.org/

[4]https://github.com/LIAAD/yake

each of these last five methods we only used the set of offensive tweets, since they are not thought to discriminate between classes.

2.5.1 Results

Table 2.1 shows the performance of BERT in offense detection after fine-tuning with each of the datasets. We evaluated different layers as output to feed the classifier on the top of BERT, including the concatenation of some layers. Specifically, we evaluated the last layers and their concatenation. That is why Table 2.1 shows the results for the two last layers, as well as for the concatenation of the four last layers. In general, the results are similar among them. Hence, in the rest of the experiments we used the output of the last layer (layer #12) to feed the classifier for fine-tuning. Furthermore, in the experiments, we used lists of keywords of different sizes. However, we only report the results taking into account 20 keywords[5].

Table 2.1: F1-score with BERT. Each column corresponds to the layer(s) used to feed the classifier in the fine-tuning.

	Layers		
Dataset	[12]	[11]	[12-9]
OFF19	0.789	0.770	0.779
OFF20	0.895	0.894	0.885

A. Implication of the parameter ϵ. In this section, we present the results obtained in the setting of the parameter ϵ. This is the parameter that our method uses to update the weights of the word pairs in the graph. Since one of our objectives is to discriminate between offensive and non−offensive tweets to select the keywords, this parameter is relevant in the model. Table 2.2 illustrates the results for different values of ϵ in IR. Although there are no relevant differences among the results, 0.1 seemed to be an appropriate value considering F@50. That is, we obtained more suitable weighted graphs with this value of ϵ. From these graphs, we obtained keywords that allowed us to retrieve offensive tweets from datasets with 0.550 of F@50 in OFF19 and 0.732 in OFF20.

B. Implication of the parameters related to the attention mechanism. We also evaluated different parameters regarding the multi-head self-attention mechanism of BERT that we used in our method. In each case, we fixed ϵ to 0.1.

First, we compared the results by varying the layer of BERT from which we leverage the attention mechanism. Table 2.3 shows the results for different

[5]Similar results are obtained with other numbers of keywords.

Table 2.2: IR results for different values of ϵ. Each column corresponds to an evaluation A-B: A is the dataset used to obtain the keywords and B is the dataset used in the evaluation.

	OFF20-OFF19		OFF19-OFF20	
ϵ	P@50	F@50	P@50	F@50
0.01	0.640	0.464	0.518	0.675
0.1	0.620	0.550	0.527	0.732
0.5	0.633	0.458	0.514	0.672
1.0	0.627	0.456	0.513	0.670

layers, specifically for the two last layers and the two first ones. We could see that using a specific BERT layer to obtain the attention values (weights of word pairs) does not seem to be significant. However, the last layer seems to be better. That is why we used the layer #12 to study other parameters.

Table 2.3: IR results when varying the attention layer. Each column corresponds to an evaluation A-B: A is the dataset used to obtain the keywords and B is the dataset used in the evaluation.

	Off20-OFF19		OFF20-Off19	
Layer	P@50	F@50	P@50	F@50
12	0.620	0.550	0.527	0.732
11	0.637	0.468	0.527	0.684
2	0.630	0.460	0.521	0.675
1	0.637	0.468	0.516	0.676

Then, we evaluated different heads in the attention mechanism of the last layer. That is, we obtained the weights of the word pairs for a specific head in that layer and compared the results with our variant of taking the combination of all the heads. Table 2.4 shows the comparison for the second and penultimate layers. We obtained worse results with the first and last layers. They seemed to be null patterns. In general, we noticed that one of the heads obtained better results than the others when they were used individually. It can be for the type of pattern represented in each particular head. The 11th head was the best in the experiments for both datasets. However, our proposal uses the combination of all the heads which obtained slightly better results according to F@50.

Moreover, we analyzed the centrality measure (CM) used to look for relevant vertices in the graph of words. Table 2.5 shows the comparison among the use of EC and other alternatives based on the degree of vertices. As we expected, the best results are obtained with the keywords obtained by using EC which takes into account the weight of edges in the graph.

Table 2.4: IR results for different heads in the layer #12. Each column corresponds to an evaluation A-B: A is the dataset used to obtain the keywords and B is the dataset used in the evaluation.

	Off20-OFF19		Off19-OFF20	
Head	P@50	F@50	P@50	F@50
All	0.620	0.550	0.527	0.732
2	0.593	0.437	0.505	0.614
11	0.415	0.537	0.527	0.684

Table 2.5: IR results for different centrality measures (CM). Each column corresponds to an evaluation A-B: A is the dataset used to obtain the keywords and B is the dataset used in the evaluation.

	Off20-OFF19		Off19-OFF20	
CM	P@50	F@50	P@50	F@50
EC	0.620	0.550	0.527	0.732
DC	0.607	0.442	0.514	0.662
IC	0.600	0.438	0.512	0.672
OC	0.597	0.436	0.516	0.677

C. Comparison with other keyword extraction methods. We fixed the parameters of our method according to the previous results. That is, we used the combination of all the heads in the attention mechanism of layer #12 of BERT. Moreover, we set ϵ to 0.1 and used the EC for extracting the keywords from the graph. With this configuration, we compare our results with other keyword extraction methods.

On the one hand, Table 2.6 illustrates the results in IR. As we expected, the results with our method are higher than those obtained with other methods that have shown good performance in general-purpose keyword extraction.

Table 2.6: F@50 in IR. Each column corresponds to an evaluation A-B: A is the dataset used to obtain the keywords and B is the dataset used in the evaluation.

Approach	Off20-OFF19	Off19-OFF20
Our	**0.550**	**0.732**
TF	0.399	0.581
TFIDF	0.490	0.605
RAKE	0.541	0.440
YAKE	0.325	0.483
TRANK	0.382	0.495

On the other hand, Table2.7 shows the results in OLD. Once again, our proposal outperformed other general-purpose keyword extraction methods. However, it is important to consider that the keyword-based models might lead to skewed results according to the bias in the datasets. Therefore, we analyze this problem later, where we compare the results with those obtained with a method that is not based on keywords.

Table 2.7: F1-scores in OLD. Each column corresponds to an evaluation A-B: A is the dataset used to obtain the keywords and B is the dataset used in the evaluation.

Approach	Off20-OFF19	Off19-OFF20
Our	**0.5687**	**0.5798**
TF	0.3747	0.4108
TFIDF	0.5047	0.5588
RAKE	0.4707	0.4588
YAKE	0.5327	0.5548
TRANK	0.4287	0.4718

2.5.2 Bias analysis

Along with the cross-validation, we included both evaluations OFF19-OFF19 and OFF20-OFF20 (in-validation). The idea is to illustrate how the extracted keywords can reveal the bias in the datasets. The three last rows in Table 2.8 show the results when we used OFF20 in the evaluation. Among these three rows, the first one corresponds to the results obtained with a model based on the keywords extracted from OFF19. The second row corresponds to the results obtained with a model based on the keywords extracted from OFF20, while the third row shows the results obtained without considering keywords.

First, we can see that the use of keywords can improve slightly the results. However, it depends on the characteristics of the keyword list. The base method (LSTM) obtained 0.7958 and this result increased to 0.8071 when the keywords from OFF20 were added. On the other hand, the results were considerably on the decline when the keywords from OFF19 were added instead. This makes sense since the first keyword list is from the same domain of the dataset used for the evaluation. Nevertheless, let's analyze the results of the evaluation with OFF19.

The three rows corresponding to the evaluation with OFF19 show a different performance. In this case, the base method obtained 0.5864 and the inclusion of keywords did not improve it, neither the keywords from OFF20 nor the keywords from OFF19. It can be explained by the bias in OFF19, which affects not only the performance when a keyword-based method is used but also the list of keywords extracted from this dataset. i.e. the keywords

are biased according to the characteristics of the dataset. Thus, conforming our results, the more skewed the dataset (from which the keywords are extracted), the worse the generalization of the keyword-based models (based on the extracted keywords).

Table 2.8: Cross-validation and in-validation in OLD. Each row corresponds to an evaluation A-B: A is the dataset used to obtain the keywords and B is the dataset used in the evaluation. A is * in the method that does not use keywords.

Evaluation with		F1-score
OFF19	OFF19-OFF19	0.5651
	OFF20-OFF19	0.5687
	*-OFF19	**0.5864**
OFF20	OFF19-OFF20	0.5798
	OFF20-OFF20	**0.8071**
	*-OFF20	0.7958

2.6 Discussion

Regarding the use of the attention mechanism of BERT to weight the word pairs (RQ1), we first fine-tuned BERT with a set of texts for the offensive language detection task. Then, with the learned weights of BERT, we captured the attention each word assigns to other words in its context to estimate the weights of the corresponding pairs. In the experimentation, we varied the parameters of the attention mechanism, i.e. the layer and heads. The variation does not seem to be significant. However, the experimental results suggest the use of the last layer of BERT and the combination of all the heads in the selected layer.

Concerning the distinction between the offensive and non−offensive texts (RQ2), we designed the method to update the weights between words according to the class of each tweet. That is, for each offensive text the weight of each word pair updates in a positive sense the corresponding edge in the word graph, while for each non−offensive text, the update is in a negative sense. Thus, we penalize the words that can be relevant to the non−offensive texts. Moreover, we used a parameter ϵ to control the update. The higher the value of ϵ, the greater the increase or decrease of the weight of the word pairs. In the experiments, we varied the value of ϵ and saw that small variations in this parameter are not relevant. However, we verified the suitability of our proposal for keyword extraction by the IR and OLD tasks. Besides, we point out that the proposed method is unsupervised, in the sense that it does not require a dataset with a set of keywords as a reference, instead it effectively only uses a set of offensive and non−offensive texts.

2.6.1 Error Analysis

In order to gain deeper insight into our method performance, we conducted an error analysis. First, we manually analyzed some keywords extracted for each of the datasets. Then, we analyzed the presence of the keywords in the instances misclassified.

Table 2.9: List of keywords

Dataset	Keywords
OFF19	'trump', 'hate', 'gun', 'anti', 'mag', 'liberals', 'stupid', 'sick', 'black', 'government', 'violence', political', 'wrong', 'bad', 'election', 'violent', 'woman', 'conservatives', 'control', 'vote', 'stop', 'country', 'people', 'president', 'law', 'white'
OFF20	'bitch', 'hate', 'bad', 'ass', 'trump', 'girl', 'stop', 'black', 'last', 'someone', 'real', 'season', 'game', 'world', 'little', 'guess', 'school', 'hard', 'person', 'god', 'old', 'twitter', 'sad', 'fun', 'white', 'work'

Table 2.9 illustrates examples of extracted keywords per dataset[6]. Some of them can be easily recognized as offensive words, like for example *'stupid'* in OFF19 and *'bitch'* in OFF20. However, others are non−offensive in a general sense. For instance, the word *'liberals'* was selected as an offensive keyword from OFF19, but we do not consider it as an offensive word. We checked on the original paper where the dataset was proposed and realized that *'liberals'* was one of the terms used to filter tweets. Nevertheless, other terms that were also used in the filtering of tweets as *'antifa'*, were not extracted as keywords by our method. Therefore, we calculated the percentage of occurrence of the words used to collect the dataset, discriminating between offensive and non−offensive tweets. As we expected, these words are very frequent in the dataset. In the case of *'liberals'*, the percentage of occurrence in offensive tweets is higher. Thus, errors can arise from those non−offensive words that are relevant only in the offensive tweets. On the other hand, errors can appear due to those non−offensive tweets that contain offensive words.

Furthermore, we conducted an error analysis on the experimental results in OLD. We observed that most of the errors were in tweets that did not contain keywords. i.e. tweets that do not contain at least one of the keywords from the list we extracted. Table 2.10 illustrates a statistic related to the tweets misclassified with both the keyword-based models and the models that do not consider the keywords (base model). In each case, it is shown the percentage of misclassified tweets without keywords (last column)

[6]Some examples can represent offensive content. They are not the views of the authors.

and the percentage of misclassified tweets with at least one keyword. With the keyword-based method, 74.24% of the errors came from tweets without keywords in OFF19, and 78.7% in OFF20. These percentages increased to 75.76% and 80.28% respectively, with the base models. Thus, the probability of error is higher in tweets that do not contain keywords.

Table 2.10: Percentage of misclassified tweets. Each row corresponds to an evaluation A-B: A is the dataset used to obtain the keywords and B is the dataset used in the evaluation. A is * in the method that does not use keywords.

Evaluation	% of tweets with keywords	% of tweets without keywords
OFF20-OFF19	28.8	74.2
*-OFF19	24.2	75.8
OFF19-OFF20	21.3	78.7
*-OFF20	19.7	80.3

Regarding OFF20-OFF19, 91.3% of the misclassified tweets that do not contain keywords (74.2% of the total of misclassified tweets), correspond to offensive tweets. This amount represents 74.7% of the total of misclassified offensive tweets. This data suggests that a large part of the errors come from offensive tweets that do not contain offensive keywords. The rest 8.7% of misclassified tweets without keywords are non−offensive, which represents 58.5% of all the misclassified non−offensive tweets. Therefore, a large percentage (41.5%) of errors in non−offensive tweets is due to the presence of keywords. This suggests a possible bias in the dataset concerning some keywords. In the case of OFF19-OFF20, all the misclassified tweets without keywords correspond to offensive tweets, which represent 80% of the total of the misclassified offensive texts.

2.6.2 Limitations of Our Work

One limitation of our keyword extraction method is that it does not consider the tokens starting with ##. BERT uses this symbol to identify parts of unknown words in the tweets. We intend to include this information in the coming works. Moreover, we attempt to extend the method for phrase extraction. The idea is to define some patterns to identify phrases in the tweets and extract those that contain closed keywords. One way to measure closeness among keywords is the sum of the weights of all the edges on the path between the words in the word graph.

Another limitation is the characteristics of the phenomenon that we aim to address. Since offensive language can be expressed subtly, many offensive tweets may simply not have words that are considered offensive. Thus, our

method can extract lists of words that do not generalize offensive content. Moreover, our method depends on the distribution of words between the classes in the dataset. The words that are relevant in non−offensive tweets will not be extracted as keywords, even when they are offensive. Therefore, the quality of the extracted keywords is data-dependent. Anyway, this can be useful, because it helps us characterize the dataset used by our method.

2.7 Conclusion and Future Work

In this paper, we proposed an unsupervised method for extracting keywords from datasets with offensive content. The aim is to study offensive language as a particular case of online harmful content. Our approach provides a way to extract keywords from datasets without the need of a tagged dataset with reference keywords. Instead, the method only uses a set of tweets tagged as offensive or non−offensive. In this sense, the extracted keywords can be used to explain the offensive language within a dataset, since they are relevant words in the offensive tweets. An important contribution lies in the exploitation of BERT. We designed the method by leveraging the abilities of the multi-head self-attention mechanism of BERT to assign attention values among word pairs in a context. In the proposal, we calculate a weight for each word pair from the tweets as the attention value obtained with BERT for this pair. Then, the weight is updated when processing each tweet containing the pair. The proportion of the update of the weight is controlled by a parameter ϵ, and the weight increases if the processed tweet is offensive and decreases otherwise. Thus, we penalized the word pairs from non−offensive tweets for distinguishing between offensive and non−offensive tweets. Then, the weights are used to represent the edges of a graph where the vertices are the words from all the tweets. This representation finally allows us to select the keywords by using the eigenvector centrality. We extrinsically evaluated the quality of the generated keyword list in two ways. On the one hand, we tested an Information Retrieval system to extract offensive tweets taking the keywords as queries. On the other hand, we evaluated the performance of a model for offense detection as a classification task. Firstly, we made experiments to find a good configuration for the parameters of our method. Then, we evaluated the suitability of our method to extract keywords for our particular purpose over other general-purpose AKE techniques. Moreover, we evaluated how our method can detect some characteristics in the datasets that can influence the performance of offense detection. In future work, we aim to expand our method for dealing with multilingual datasets.

Acknowledgements

This research work was partially funded by the Spanish Ministry of Science and Innovation under the research project MISMIS-FAKEnHATE on Misinformation and Miscommunication in social media: FAKE news and HATE speech (PGC2018-096212-B-C31). The authors thank also the EU-FEDER Comunitat Valenciana 2014-2020 grant IDIFEDER/2018/025.

Chapter 3

Systematic Keyword and Bias Analyses in Hate Speech Detection

Published in:

- **De la Peña Sarracén, G.L.** and Rosso, P. (2023). Systematic Keyword and Bias Analyses in Hate Speech Detection. Information Processing & Management, 60(5), (pp. 103433).
  (Impact Factor: 7.466, Q1)

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 analyzing two transformer-based models trained to detect hate speech. Our study focuses on analyzing 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 favor 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.

3.1 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 humor is used (Fortuna and Nunes, 2018a). 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 labeled 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 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 analyze 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

[1]NOTE: This paper contains examples of potentially explicit offensive content. They do not represent the views of the authors.

inspired by the concept of fairness and one metric introduced by Borkan et al. (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 et al., 2022; Sánchez-Junquera et al., 2021b).

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. In particular, we use this method for the HSD task in order to extract hateful keywords. Second, we analyze the effect of bias mitigation for critical cases of hate speech detection.

The rest of this paper is organized as follows. Section 3.2 summarizes the related work and Section 3.3 introduces the text collections that we use in this work. Sections 3.4, 3.5 and 3.6 present our studies and findings for the research questions respectively. In particular, the description of our proposal to extract keywords is in Section 3.5. Finally, Section 3.7 presents limitations and ethical concerns of our research work, and Section 3.8 concludes the paper.

3.2 Related Work

Different strategies have been proposed for hate speech detection, which ranges from methods based on linguistic characteristics to machine learning techniques. Poletto et al. (2021), Velankar et al. (2022) and Alkomah and Ma (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 et al. (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 hate speech detectors based in transformer 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 analyze 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 et al., 2020) used BERT, ALBERTo, and UmBERTo language models to reach the first position in HaSpeeDe 2.

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 et al. (2019) analyzed how high-ranking scores in biased datasets that contain mostly implicit abuse, are due to bias modeling in those datasets. In our work, we analyze this report by studying the relationship between hateful keywords and transformers-based models. Balkir et al. (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 et al. (2019) analyzed 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 et al.

(2020) stated the bias in annotated training data causes text to often be mislabeled 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 analyze how the bias varies with strategies aimed at reducing bias. Mozafari et al. (2020b) 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 used 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.

3.3 Datasets

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

- **HatEval:** It is the dataset used for Task 5 of SemEval 2019.[2] 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 9,000 tweets for training and 1,000 tweets for development.

- **Waseem & Hovy (W&H):** A popular dataset referenced in several studies (Gröndahl et al., 2018; Arango et al., 2019; Schmidt and Wiegand, 2017; Fortuna and Nunes, 2018a; Poletto et al., 2021). 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 3,378 are labeled as sexist and 1,970 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.

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

- **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 3,635 tweets tagged as hateful, 10,122 tagged as abusive, 13,404 tagged as spam, and 52,835 tagged as normal.

Table 3.1 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.

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%

Table 3.1: Essential information of the text collections.

3.4 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 behavior 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[3], 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

[3] https://captum.ai/

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 3.2. They are trained to detect hate speech and are accessible in HuggingFace for English.

Model	Architecture	URL
BERT (Aluru et al., 2020)	bert	https://huggingface.co/Hate-speech-CNERG/dehatebert-mono-english
ROBERTA (Vidgen et al., 2021)	roberta-base	https://huggingface.co/facebook/roberta-hate-speech-dynabench-r4-target

Table 3.2: Links to the pre-trained models used in this work.

Findings. Figure 3.1 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 in identifying hatred in subtle cases, it may fail in the critical cases exemplified by Examples 5 and 6. In section 3.6 we analyze this issue in more detail by examining the bias of the model towards a set of hateful words.

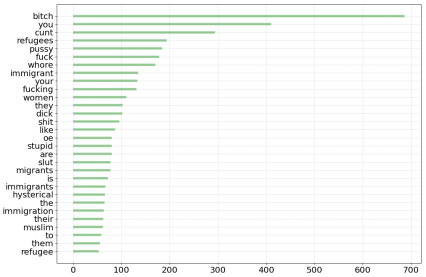
Once we obtain a model attribution for the words in each text, we calculate the general word score in a text collection. The score is the sum of the attributions that the words receive in the texts of the collection in which they appear. Note that the higher the attribution that the model gives to a word in each instance, the higher its score is. Similarly, the score of the words to

Legend: Negative Neutral Positive

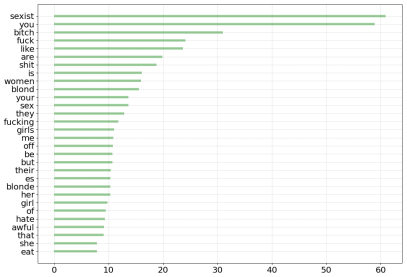
	True Label	Predicted Label	Word Importance
1	hate	hate	#s Kat ie and Nikki have definitely proved they re not just dumb blond es with pretty faces . They re just dumb blond es
2	hate	hate	#s # mk r O mg there restaurant looks like a a little girls room oh my ! G osh #/s
3	hate	hate	#s These girls are the equivalent of the irritating Asian girls a couple years ago . Well done
4	not-hate	not-hate	#s . . but you ask a guy yo go hiking with you just so you can have him carry your bags and pay all the expenses . #/s
5	hate	not-hate	#s Kim Kardashian has had marriages that didnt stretch as long as # M KR #/s
6	not-hate	hate	#s @ P inea q ples @ Den ial Es ports b tw I watched where you watched my p ov on stream and bo ii do I feel retarded #/s

Figure 3.1: Attributions with Transformers-Interpret for randomly selected texts from the text collections. The positive class (green) corresponds to the class of hateful texts.

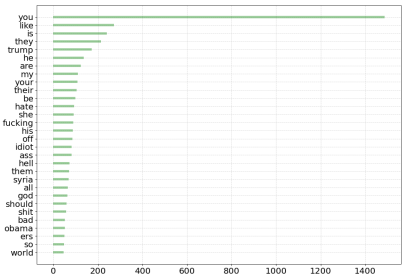
which the model gives negative attribution is small. Figure 3.2 shows the top 30 words of the ranking generated with ROBERTA in each text collection. The way to access the analogous result for BERT is in appendix 3.8.



(a) HatEval



(b) W&H



(c) Founta

Figure 3.2: Ranking of the words to which ROBERTA pays the most attention in each text collection

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 Figure 3.1, where this behavior 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 3.3 shows the results in terms of F1-score (see Powers (2015) to understand the F1 metric).

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

Table 3.3: Hate speech detection results in terms of F1-score. Numbers in bold indicate the best significant result with a significance level of .05.

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 us 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 3.3 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 3.6. 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 to and their performance.

3.5 Analysis of Hateful Keyword

Once we have analyzed the salient words of the models, we may address **RQ2** from Section 3.1 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 keyword 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 favor 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.

3.5.1 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 analyze 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 (Equation 3.1), and 2) the frequency of w in S with respect to its frequency in the entire collection of texts C (Equation 3.2). The variable k identifies all possible words in the text set and the indicator $\mathbf{1}_{(w)}(t)$ defines the number of times that the word w appears in a text t.

$$f_1^S(w) = \frac{\sum_{t \in \mathrm{S}} \mathbf{1}_{(w)}(t)}{\sum_k \sum_{t \in \mathrm{S}} \mathbf{1}_{(k)}(t)} \tag{3.1}$$

$$f_2^S(w) = \frac{\sum_{t \in \mathrm{S}} \mathbf{1}_{(w)}(t)}{\sum_{t \in \mathrm{C}} \mathbf{1}_{(w)}(t)} \tag{3.2}$$

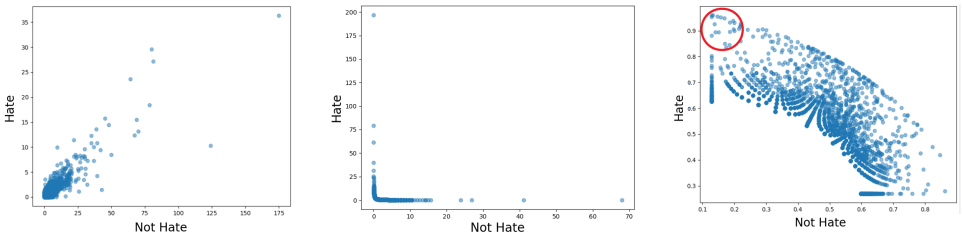
Then, we use the cumulative distribution function (CDF)[4] on the relative frequencies. This is a distribution function of a random variable X: $F_X(x) = P(X \leq x)$. So, $CDF(f_1^S)$ indicates the ratio of words that will take a value of f_1^S less than or equal to $f_1^S(w)$. Similarly, $CDF(f_2^S(w))$ 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 et al., 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. Equation 3.3 specifies how the harmonic mean is used to obtain the final score for w.

$$HMRF_S(w) = \frac{2 * CDF(f_1^S(w)) * CDF(f_2^S(w))}{CDF(f_1^S(w)) + CDF(f_2^S(w))} \tag{3.3}$$

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 $\{\mathrm{H} \cup \mathrm{N}\}$. In this way, w is represented by the tuple ($\mathrm{HMRF}_N(w)$,$\mathrm{HMRF}_H(w)$).

Figure 3.3 shows word-tuple representations extracted from the HatEval dataset (Basile et al., 2019a) as points on a plane. Figures 3.3a and 3.3b show the points according to TFIDF and PWI respectively. While Figure 3.3c 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.



(a) TFIDF (b) PWI (c) HMRF

Figure 3.3: Representation of terms.

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

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

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.

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.[5] 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. To obtain the phrases we only use hateful texts and a strategy to analyze 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 (Equation 3.4). Where n is the size of the sequence and P is its probability in the text set.

$$PMI(W) = log_2 \frac{P(w_1, \ldots, w_n)}{\prod_{i=1}^{n} P(w_i)} \tag{3.4}$$

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.

3.5.2 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.

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

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).

3.5.3 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 analyzed whether the extracted words reflect the focus of the source dataset. Next, we analyzed if the extracted keywords are really potentially hateful terms (words and phrases). Table 3.4 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 is not fixed.

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 3.4: Sets of keywords extracted with HMRF and YAKE.

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 3.5 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.

	HateEval		**W&H**		**Founta**	
Class	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

Table 3.5: 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.

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 3.4 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 analyze the overlap between each pair of word sets.

Overlap among keywords sets. Figure 3.4 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.

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

Figure 3.4: Heatmap of the overlap between each pair of word sets. Each cell represents the percentage of overlap taken as the total size of the group of words on the X-axis.

3.6 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 3.5, in this section we investigate **RQ3** of Section 3.1: our goal is to evaluate the role of hateful keywords, extracted with HMRF, in hate speech detection with HSD models. We first analyze 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.

3.6.1 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 et al. (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 toward 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 Equation 3.5.

$$bias = |P(\hat{Y} = 1|K = 1) - P(\hat{Y} = 1|K = 0)| \tag{3.5}$$

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 modeling. We follow the Equations 3.6 and 3.7 to calculate the probabilities.

$$P(\hat{Y} = 1|K = 1) = \frac{\#\text{ Texts containing K and classified as hateful}}{\#\text{ Texts containing K}} \tag{3.6}$$

$$P(\hat{Y} = 1|K = 0) = \frac{\#\text{ Texts not containing K and classified as hateful}}{\#\text{ Texts not containing K}} \tag{3.7}$$

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'.[6] This metric considers both texts classified as hateful and texts classified as non-hateful, by using three sub-metrics based on the ROC-AUC[7] 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 of distinguishing between hateful and non-hateful texts that contain the keyword.

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

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

- **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:

$$M_p(n) = \left(\frac{1}{K} \sum_{k=1}^{K} m_{k,n}^p \right)^{\frac{1}{p}} \tag{3.8}$$

$m_{k,n}$ 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.

$$score = w_0 AUC_{overall} + \sum_{n=1}^{N} w_n M_p(n) \tag{3.9}$$

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 on 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 3.1. 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 analyze the behavior 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.
- $V2_{HMRF}$: 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 3.4 (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 $1x10^{-7}$. 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 3.4. 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).

3.6.2 Results and Discussion

A summary of the estimated biases, per model and dataset, is provided in Table 3.6.

Following is a list of the most important findings when analyzing the table:

	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

Table 3.6: 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 toward the keywords specified in the columns (HMRF and YAKE), based on ROC-AUC metrics (**b1**) and fairness (**b2**) .

- 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 behavior 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 behavior in Founta with BERT.
- The bias **b2** seems to have the opposite behavior: V1 tends to increase **b2** with respect to the Original model, while V2 tends to reduce it.

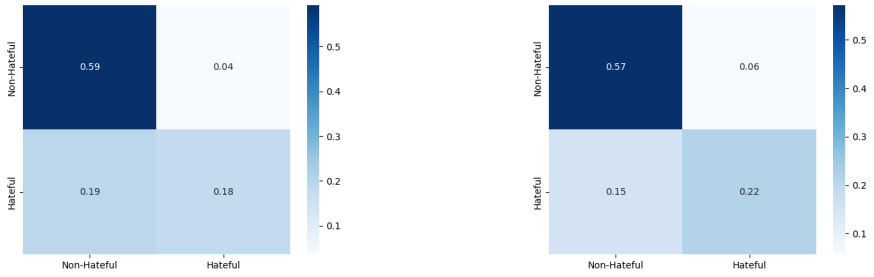
Once we have analyzed the variation of the bias towards the keywords extracted with HMRF and YAKE, we need to evaluate how this influences the variation of the performance of the HSD models. Table 3.7 summarizes the results for all models in terms of F1-score and P value. We do the statistical analysis 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 analyze the variation of F1 in relation to the variation of bias. Taking into account the analyzed 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 of F1. However, note that there are unexpected cases, where the behavior is different. Therefore, we cannot assert that there is a correlation.

Finally, let's remember that **b2** only takes into account instances classified as hateful. With V1 this class is favored, so the number of texts classified as hateful will tend to increase (see an example in Figure 3.5). The opposite happens with V2, whereby disfavoring 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 favor the values of F1 by improving the positive class (hateful texts).

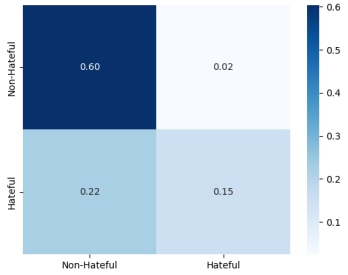
	HateEval		**W&H**		**Founta**	
	F1	P value	F1	P value	F1	P value
BERT						
Original	**0.8004**	-	0.6034	-	0.6188	-
$V1_{HMRF}$	0.7927	P=.13	0.6161	P<.001	**0.6378**	P<.001
$V1_{YAKE}$	0.7890	P<.001	**0.6199**	P<.001	0.6375	P<.001
$V2_{HMRF}$	0.7994	P<.001	0.5572	P<.001	0.6124	P<.001
$V2_{YAKE}$	0.8003	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

Table 3.7: 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 $\alpha = .05$.

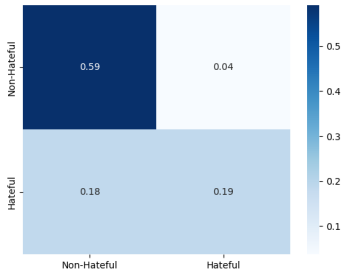


(a) Original

(b) V1$_{HMRF}$



(c) V2$_{HMRF}$



(d) V3

Figure 3.5: Confusion matrices for classification with ROBERTA in W&H. Rows represent the actual labels and columns the predicted labels.

3.7 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. Note that our goal is limited to assisting in this effort to help online platforms identify and remove hateful content.

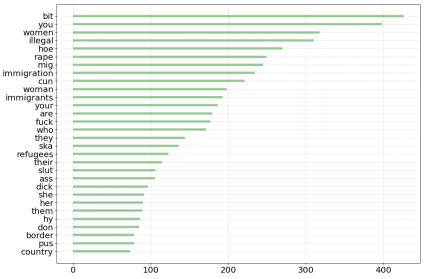
3.8 Conclusions

Transformer-based models have arrived to mark an important leap in different natural language processing tasks. Among them, hate speech detection has been favored 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 pretrained 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 analyzed 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 analyze the bias towards hateful keywords taken from external resources such as HurtLex (Bassignana et al., 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.

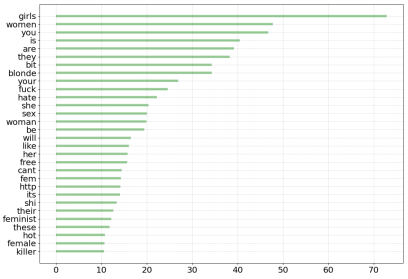
Acknowledgments

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.

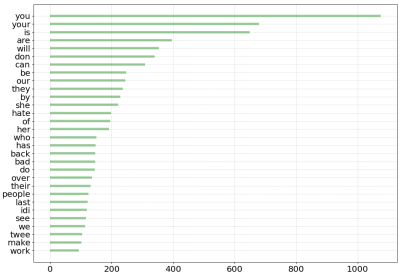
Appendix



(a) HatEval



(b) W&H



(c) Founta

Figure 3.6: Ranking of the words to which BERT pays the most attention in each text collection

Part II

Graph-Based Exploration

In this second part, we investigate the potential of models based on graph neural networks for hate speech detection. In Chapter 4 we propose a graph auto-encoder framework to obtain a latent representation from an initial text representation. We used this framework for hate speech detection by using the embeddings as input of a classifier. In Chapter 5 we study a model based on convolutional graph neural networks to address hate speech detection in scenarios with little data.

Chapter 4

Unsupervised Embeddings with Graph Auto-Encoders for Multi-Domain and Multilingual Hate Speech Detection

Published in:

- **De la Peña Sarracén, G.L.** and Rosso, P. (2022). Unsupervised Embeddings with Graph Auto-Encoders for Multi-Domain and Multilingual Hate Speech Detection. Proceedings of the Thirteenth Language Resources and Evaluation Conference, (pp. 2196-2204).
  (Core B Conference)

Abstract. Hate speech detection is a prominent and challenging task since hate messages are often expressed in subtle ways and with characteristics that may vary depending on the author. Hence, many models suffer from the generalization problem. However, retrieving and monitoring hateful content on social media is a current necessity. In this paper, we propose an unsupervised approach using Graph Auto-Encoders (GAE), which allows us to avoid using labeled data when training the representation of the texts. Specifically, we represent texts as nodes of a graph and use a transformer layer together with a convolutional layer to encode these nodes in a low-dimensional space. As a result, we obtain embeddings that can be decoded into a reconstruction of the original network. Our main idea is to learn a model with a set of texts without supervision, in order to generate embeddings for the nodes: nodes with the same label should be close in the embedding space, which, in turn, should allow us to distinguish among classes. We employ this strategy to detect hate speech in multi-domain and multilingual sets of texts, where our method shows competitive results on small datasets.

4.1 Introduction

In this paper, we investigate an unsupervised, graph-based approach to learning embeddings for hate speech detection. According to Fortuna and Nunes (2018b), hate speech can be defined as a language that attacks, diminishes or incites violence against groups based on specific characteristics. Accordingly, the aim of hate speech detection is to discriminate texts that contain hate from those that do not. This is a widely studied task that involves a number of challenges (Poletto et al., 2021). Specifically, we study the problem of data-poor settings that appear in low-resource domains and languages – i.e., in those settings where supervised approaches are not able to generalize well.

For this, we make use of Graph neural networks (GNNs). They are a framework based on deep learning to operate on graphs. They follow a recursive neighborhood aggregation scheme, called message passing, where each node aggregates feature vectors of its neighbors to compute its new feature vector (Xu et al., 2018). After a number of iterations, a node is represented by its transformed feature vector, which captures the structural information within its neighborhood. Therefore, GNNs have been effective at tasks thought to have rich relational structure since they can preserve global structure information of a graph in embeddings. Wu et al. (2021) provide an overview of recent studies on deep learning approaches for graphs and discuss the applications of GNNs in several areas. Zhou et al. (2020) present some GNN-based methods that have been applied to text classification, and point out that representing texts as graphs can effectively capture semantics among words. Yao et al. (2019b), for instance, use GNNs for text classi-

fication. The authors propose a strategy to represent texts as graphs and use a convolutional graph neural network to learn embeddings of words and documents. As a result, they show that the improvement of their graph-based model over state-of-the-art models becomes more prominent as the percentage of training data is lower.

Cross-lingual transfer learning is one of the strategies used to leverage already existing data from higher-resource languages (Ranasinghe and Zampieri, 2020a; Stappen et al., 2020a; Bigoulaeva et al., 2021b). However, this approach can introduce other problems related to the variability between different languages. Moreover, the resulting datasets are heterogeneous not only in terms of languages but also in terms of domains, which can affect the learning of the models.

We aim to study the performance of hate speech detection with GNNs with the motivation of improving this task in data-poor settings without the need of external data. For this, we propose a graph auto-encoder framework that allows us to learn a latent representation from a set of texts in an unsupervised way. In this representation the texts from the same class are close, hence we can use embeddings from this space to efficiently distinguish instances of different classes. Our framework builds a graph with the texts, encodes the nodes in a low-dimensional space (the latent space)„ and then reconstructs the original graph with a decoder. The encoder is composed of a transformer layer, which introduces an attention mechanism, and a convolutional layer for generating the embeddings.

We evaluate the embeddings for hate speech detection as a classification task in small datasets. Moreover, we study the multi-domain and multilingual settings, to experimentally analyze whether graphs can jointly represent different types of information. Our contributions are the following ones[1]:

- We propose a graph auto-encoder for unsupervised representation learning on graph-structured data by reconstructing the initial graph. In this framework, we incorporate a self-attention mechanism that allows us to adapt the strengths of the Transformer model (Vaswani et al., 2017b) in the generation of embeddings.

- We use the embeddings generated with this framework for hate speech detection and show results that outperform state-of-the-art models in data-poor settings, without using pre-trained word embeddings.

- We investigate the performance of our approach in multi-domain settings, where the small amount of data once more highlights the potentiality of our proposal.

- We extend the analysis for multilingual hate speech detection and use

[1]We will make our codes freely available by the publication date of this work.

a strategy to aggregate prior knowledge about the language to obtain outstanding results.

4.2 Related Work

Hate Speech Detection. Most of the works done to detect hate speech are based on the analysis of textual instances. Other works have studied the phenomenon at the author level (Rangel et al., 2021), where the idea is to analyze a set of texts published by the same author to detect possible propagators of hate on the web. In general, the techniques used for hate speech detection range from traditional machine learning models to methods based on deep learning, such as convolutional neural networks and recurrent neural networks, including attention mechanisms (Badjatiya et al., 2017; Gröndahl et al., 2018; Magalhaes, 2019). Due to the nature of the task, it is worth noting the models that take into account certain keywords that may indicate hateful content. De la Peña Sarracén and Rosso (2021) proposed an approach for keyword extraction based on the attention mechanism of BERT and reasoning on a word graph. Experimental results highlighted some points to consider when training models based on keywords. Moreover, De la Peña Sarracén and Rosso (2023) studied how models learn bias towards relevant words in the training data. To extract the relevant words, the authors proposed a keyword extraction method based on the harmonic mean of relative frequencies and the discrimination between hateful and non-hateful texts. In recent years, the bidirectional encoder representations from Transformers (BERT) (Devlin et al., 2019c), as well as other transformer-based models such as RoBERTa (Liu et al., 2019b) have been widely used due to their ability to capture language phenomena (Mozafari et al., 2020b; Samghabadi et al., 2020). In fact, they have been used in most systems with outstanding results in shared tasks (Basile et al., 2019b; Zampieri et al., 2020b; Mandl et al., 2019b). Moreover, Mozafari et al. (2019) investigated the ability of BERT to detect hateful content on social media and the results showed a considerable performance in comparison to other existing approaches. That is why we use this model to compare the results obtained with our framework.

Graph Neural Network for Abusive Language Detection. Regarding GNN-based models, the literature points out a number of strategies (Koncel-Kedziorski et al., 2019; Shi et al., 2021a). Peng et al. (2018) proposed a graph-based deep learning model to convert texts to graphs of words, and then used graph convolution operations over the graph. Yao et al. (2019b) represented documents and words as nodes to construct a graph and used a convolutional graph neural network to learn embeddings of words and documents. As a result, the authors showed improvements over state-of-the-art models for text classification. However, very little has been studied to employ

strategies based on GNNs to address the problem of hate speech detection. Mishra et al. (2019) proposed a convolutional graph neural network for capturing the structure of online communities and the linguistic behavior of the users. They showed that the resulting heterogeneous graph significantly advanced the state of the art in abusive language detection. Thus, to the best of our knowledge, our method is the first proposal to learn embeddings in an unsupervised way for the specific problem of hate speech detection.

4.3 Graph Auto-Encoders for Hate Speech Detection

In this section, we describe the preliminaries of the framework, followed by details of our proposal.

4.3.1 Formalization

In this work, we consider hate speech detection as a classification problem that involves the classes hate and not hate. The data comprises N samples, where each sample is given by $\{t_i, y_i\}$. The set $\{t_i\}_{i=1}^{N}$ is composed of texts that are represented with numeric feature vectors $\{x_i\}_{i=1}^{N}$. In order to generate these feature vectors we use Term Frequency - Inverse Document Frequency (TFIDF) representation of each text in $\{t_i\}_{i=1}^{N}$.

TFIDF generates vectors from texts in such a way that often produces lower scores for high frequency function words and increases scores for terms that are more relevant in each text, it is well suited for tasks involving textual similarity. Thus, we represent the texts as vectors with TFIDF scores, for which we do not need external sources to train the initial vectors.

The set $\{y_i\}_{i=1}^{N}$ is composed of the labels 0 and 1, which indicate the presence or not of hate in each of the texts in $\{t_i\}_{i=1}^{N}$. Then, hate speech detection aims to assign one of the labels to each t_i by using x_i.

Our aim in the present work is to learn embeddings from x_i in an unsupervised way to improve the performance in hate speech detection when N is small. Besides, we attempt to use this approach for multi-domain and multilingual hate speech detection, due to the suitability of graphs to jointly represent different types of information. In these cases $\{t_i\}_{i=1}^{N} = \cup_m \{t_i^m\}_{i=1}^{S_m}$, where m represents each of the M domains or languages and S_m its size, such that $N = \sum_{m=1}^{M} S_m$.

We address the problem by using a graph auto-encoder framework. Following, we describe our framework in detail.

4.3.2 Background: Graph Auto-Encoders

Graph neural networks (GNNs) are models based on deep learning to operate on the graph domain. In particular, Graph Auto-Encoders (GAEs)

are unsupervised learning frameworks which encode nodes or graphs into a latent vector space. Therefore, they can be used to learn embeddings. In general, they are trained with the aim of reconstructing their original graph input. First, an encoder takes a graph as its input and compresses it into a low-dimensional vector. Then, a decoder takes this vector representation and attempts to generate a reconstruction of the original input. Encoder-decoder pair is designed to minimize the loss of information between the input graph and the output graph (Wu et al., 2021).

Formally, let $G = (V, E)$ be a graph, where V and E represent the set of nodes and edges respectively. Let $X \in \mathbb{R}^{|V| \times d}$ be a matrix containing the features of the nodes, such that the $i-th$ row is a d-dimensional feature vector of the $i-th$ node. Moreover, let $A \in \mathbb{R}^{|V| \times |V|}$ be a matrix representation with a representative description of the graph structure, such as the adjacency matrix. A GAE takes as input the matrices X and A to learn a function $Z = enc(X,A)$ and produces a latent representation $Z \in \mathbb{R}^{|V| \times d'}$ (embeddings), where $d' < d$ is the number of features of the nodes in the latent representation. Then, Z is used to produce an approximate reconstruction output $\hat{A} = dec(Z)$ such that the error between A and $\hat{A}$ is minimized for preserving the global graph structure. Both functions $enc(\cdot, \cdot)$ and $dec(\cdot)$ are often defined through stacked layers.

Graph convolutional layer (GCL) re-defines the notion of convolution for graph data and is widely used as propagation operators for GNNs in general. The main idea is to operate directly on a graph and induce the embedding vectors of nodes based on the properties of their neighbors. A GCL takes as input the matrices X and A and generates a representation $H = f(X, A)$, where $f(\cdot, \cdot)$ is a propagation rule. Kipf and Welling (2017a) introduced the propagation rule (4.1), where W is a weight matrix and $\sigma(\cdot)$ is an activation function. The matrix $\tilde{A} = A + I$ (I is the identity matrix) contains self-connections to aggregate, for each node, not only the information from its neighbors but also the node itself. Moreover, the matrix D is the diagonal node degree matrix of $\tilde{A}$, which is used for a symmetric normalization to deal with the problem of changing the scale of the feature vectors.

$$f(X, A) = \sigma(D^{-\frac{1}{2}} \tilde{A} D^{-\frac{1}{2}} X W) \tag{4.1}$$

Graph transformer layer (GTL) adapts the multi-head attention of Transformer (Vaswani et al., 2017b) for graph learning. This was introduced for Shi et al. (2021c) considering the case of edge features. Given the features vectors $X = \{x_i\}_{i=1}^{N}$, they generate new features vectors $\hat{X} = \{\hat{x_i}\}_{i=1}^{N}$ by calculating multi-head attention for each node i with its neighbors $\mathcal{N}(i)$:

$$q_{c,i} = W_{c,q} x_i \,,\; k_{c,j} = W_{c,k} x_j \,,\; v_{c,j} = W_{c,v} x_j$$
$$e_{c,ij} = W_{c,e} e_{ij}$$
$$\alpha_{c,ij} = softmax(\frac{q_{c,i}(k_{c,j} + e_{c,ij})}{\sqrt{d_c}})$$

$$r_i = W_r x_i$$
$$\hat{x_i} = r_i + \frac{1}{C}\sum_{c=1}^{C}[\sum_{j \in \mathcal{N}(i)} \alpha_{c,ij}(v_{c,j} + e_{c,ij})]$$

where c represents each head, d_c its hidden size and C the total number of heads. The vectors $q_{c,i}$, $k_{c,j}$ and $v_{c,j}$ correspond to the 'query', 'key', and 'value' vectors respectively, and $e_{c,ij}$ is a representation for the edge between i and j. $W_{c,q}$, $W_{c,k}$, $W_{c,v}$ and $W_{c,e}$ are the parameters in the head c. Notice that a term r_i is calculated to add a gated residual connection between layers.

4.3.3 Auto-Encoder Architecture

Figure 4.1 illustrates our auto-encoder. In order to generate the input for the model, we build the matrix X with the set of numeric feature vectors $\{x_i\}_{i=1}^{N}$, such that each vector is a row in X. On the other hand, we build the edges among nodes, to generate the matrix A, based on the inner product of the feature vectors. Then, the weight of each edge is defined by the inner product between the original vectors. We only consider edges between node pairs with values higher than a threshold (positive edges). The rest of the node pairs are considered as non-existent edges (negative edges).

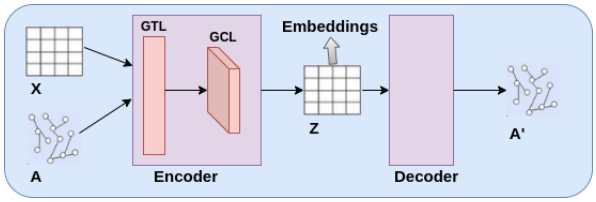


Figure 4.1: Auto-encoder architecture.

The encoder in our model stacks two layers. The first one is a GTL and the second one is a GCL. In particular, we use a GTL to enrich the node embeddings with attentive information propagation between nodes. Thus, the encoder uses a GTL as the first layer to determine the relevance between nodes and their neighbors by leveraging the advantages of the attention mechanism among nodes. In this sense, we adopt the proposal of Shi et al. (2021c) by considering only nodes and using a unique head. Hence, we transform the input X matrix as (4.2).

$$\begin{aligned} \alpha_{ij} &= softmax(\frac{(W_q x_i)^T W_k x_j}{\sqrt{d}}) \\ \hat{x_i} &= W_r x_i + \sum_{j \in \mathcal{N}(i)} \alpha_{ij}(W_v x_j) \end{aligned} \tag{4.2}$$

The second and final layer is based on the propagation rule (4.1) and the ReLU as the activation function. The input of this layer is composed of the new matrix $\hat{X}$ and the matrix A. Thus, we obtain the output of the encoder as (4.3), where W_c is a parameter matrix.

$$Z = enc(X, A) = ReLU(D^{-\frac{1}{2}}\tilde{A}D^{-\frac{1}{2}}\hat{X}W_c) \qquad Encoder \tag{4.3}$$

The decoder implements the idea of the GAE of Kipf and Welling (2016). Thus, we base on the inner product of the embeddings to generate $\hat{A}$. The aim is to decode node relational information from the embeddings by reconstructing A as (4.4) defines. Then, the auto-encoder (4.5) is trained by minimizing the negative cross entropy given the real matrix A and the reconstructed matrix $\hat{A}$.

$$\hat{A} = dec(Z) = sigmoid(ZZ^T) \qquad Decoder \tag{4.4}$$

$$\hat{A} = GAE(X, A) = dec(enc(X, A)) \qquad Auto - Encoder \tag{4.5}$$

4.4 Experimental Design

In this section, we present our methodology for the empirical evaluation of the capability of our framework for unsupervised learning of embeddings. We also present the used dataset and details for the reproduction of the experiments.

4.4.1 Dataset

We evaluate our proposed auto-encoder framework on the XHate-999 dataset (Glavaš et al., 2020), which was built for abusive language detection. This dataset is composed of large training and validation sets of English texts, and a small multi-domain and multilingual test set. The test set contains text of six typologically diverse languages: English (EN), German (DE), Russian (RU), Turkish (TR), Croatian (HR), and Albanian (SQ). For each language, there are three distinct domains: Fox News (GAO) with 99 samples, Twitter/Facebook (TRAC) with 300 samples, and Wikipedia (WUL) with 600 samples, for a total of 999 samples per language. We only rely on the test set since our purpose is to study the data-poor settings as well as the multi-domain and multilingual perspective.

4.4.2 Experimental Setup

For each experiment, we set the size of the vectors generated with the GTL in the encoder to 32 and the size of the output of the GCL to 16. The auto-encoder was trained using batches of 32 instances and the Adam optimizer with a learning rate of 0.01, in 200 epochs with the strategy of early stopping with patience of 10. For the threshold used in the generation of the matrix A, we searched in $\{0.01, 0.1, 0.5\}$, but realized that a value close to the average of the weights calculated for the pairs of vectors fitted in a better way, hence we set this value to 0.07.

In the evaluation, we first visualize the embeddings in the latent representation, generated with the encoder of our graph auto-encoder. We also visualize the initial vectors (TFIDF) to visually compare both representations.

Secondly, we evaluate the capacity of the embeddings on the task of node classification to study the performance of hate speech detection. In this sense, we use a classifier of two fully connected layers of 32 neurons with the ReLU activation function and the softmax function in the last layer to generate the predictions. This classifier was used to obtain prediction for the texts with the initial representation and on the other hand, with the embeddings obtained with our encoder. Hence, we can compare the results of classification between them. In both cases, the classifier was trained with a size of batch 32, using the Adam optimizer with a learning rate of 0.01, in 10 epochs. For the test we separated the 30% of the data and the rest was used to train. That is, we used 30% from the test set of the XHate-999 dataset for testing and the other 70% for training. We ran all the experiments five times and report the average scores.

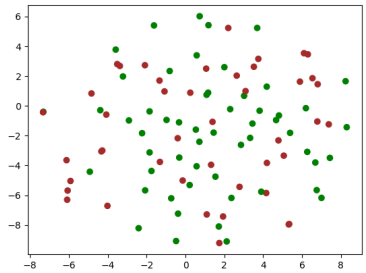
4.5 Embeddings Evaluation

In this section, we analyze the mono-domain and mono-lingual evaluation. Therefore, we focus on each domain and each language separately.

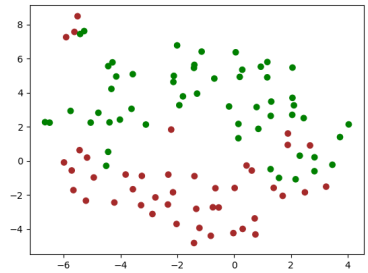
4.5.1 Analysis of Latent Representation

In order to better analyze the generated embeddings by our encoder, we use t-SNE (Pezzotti et al., 2017a) to visualize the initial and the latent representation of the vectors, corresponding to the hateful and non-hateful texts. As an illustration, Figure 4.2 shows the results for the texts in English in each of the domains GAO, TRAC, and WUL. In each case, the representation for the initial vectors (with TFIDF) is visualized on the left, and on the right there is the latent representation (embeddings). We can see that our model can be used to distinguish both classes since the separation between hateful texts (red points) and non-hateful texts (green points) is more

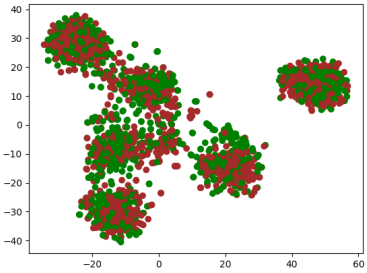
evident. Then, with this representation, a simple algorithm can be used to separate both types of texts. Similar behavior was observed for the rest of the languages.



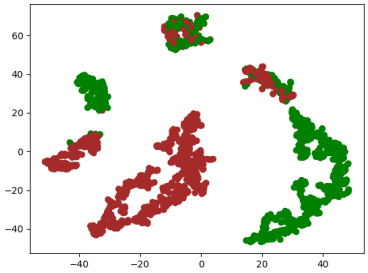
(a) GAO - Initial



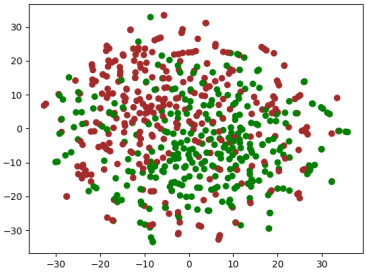
(b) GAO - Embeddings



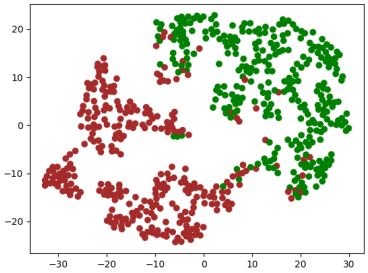
(c) TRAC - Initial



(d) TRAC - Embeddings



(e) WUL - Initial

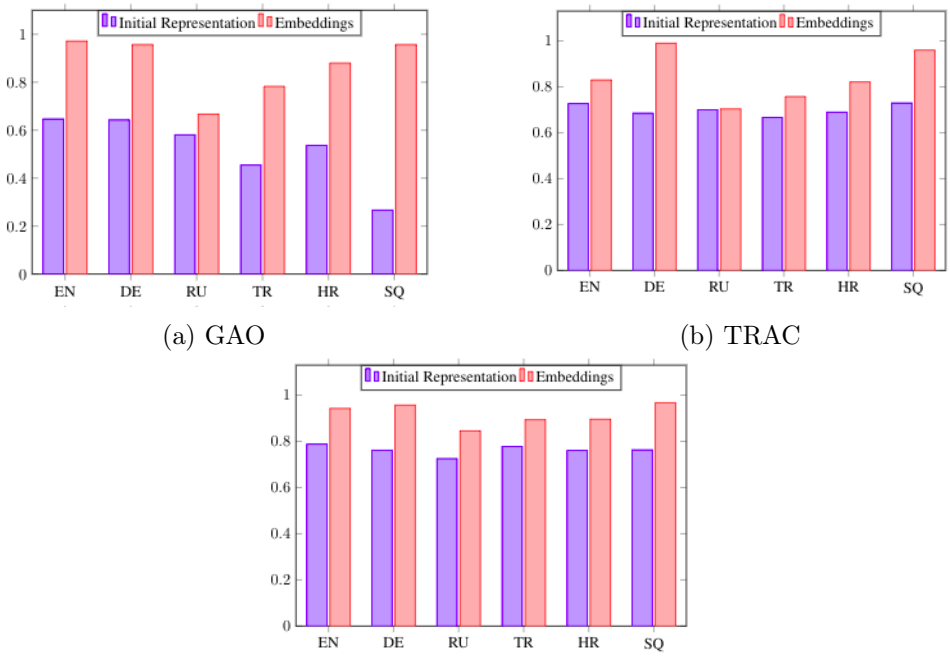


(f) WUL - Embeddings

Figure 4.2: Representation for English texts with t-SNE.

4.5.2 Evaluation for Hate Speech Detection

The results for hate speech detection, using both the initial vectors and the embeddings, are summarized in Figure 4.3. In general, we observe an improvement by using the embeddings as the input in the classifier. Thus, we can verify the suitability of the embeddings to discriminate among classes.



(a) GAO

(b) TRAC

(c) WUL

Figure 4.3: F1 in Hate Speech Detection.

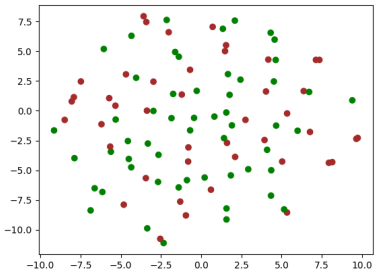
Notice that the results between both variants were similar only for the case of Russian texts in the TRAC domain. Figure 4.4 illustrates the initial and latent representation for Russian, where we can see that in the TRAC domain, it is more difficult to learn embeddings that allow discriminating between classes. This suggests that in this domain and language, the hateful and non-hateful texts are more similar. In future work, we will try to increase the number of convolutional layers in the encoder to make a deeper propagation and analyze if this case improves. Anyway, for GAO and WUL in this same language, we observe better performance for the embeddings.

4.6 Multi-domain Evaluation

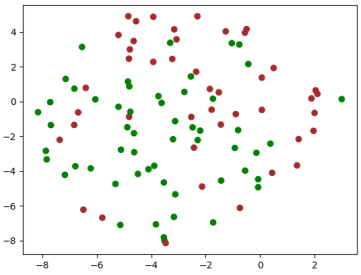
Besides the mono-domain evaluation, we focus on the multi-domain setting. In this case, the set of texts for the input of the auto-encoder is composed of three different types of data per language i.e. $\{t_i\}_{i=1}^{N} = \{t_i^{GAO}\}_{i=1}^{99} \cup \{t_i^{TRAC}\}_{i=1}^{300} \cup \{t_i^{WUL}\}_{i=1}^{600}$.

In order to compare our classification results with a state-of-the-art model we use the multilingual BERT (mBERT) (Devlin et al., 2019c) and XLM-R (Conneau et al., 2020a). We used the HuggingFace Transformers framework[2]

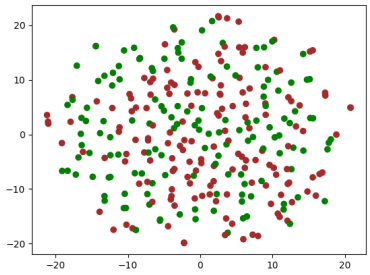
[2]https://github.com/huggingface/transformers



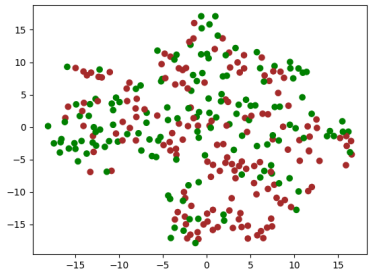
(a) GAO - Initial



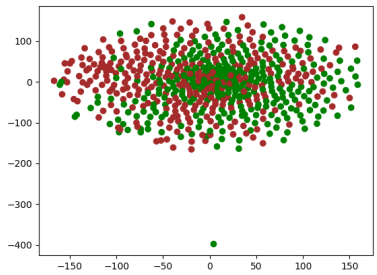
(b) GAO - Embeddings



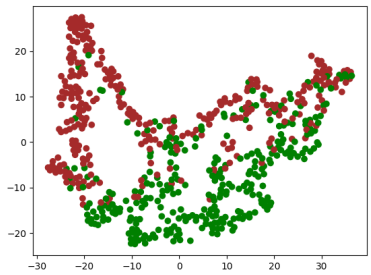
(c) TRAC - Initial



(d) TRAC - Embeddings



(e) WUL - Initial



(f) WUL - Embeddings

Figure 4.4: Representation for Russian texts with t-SNE.

with the pre-trained models bert-base-multilingual-cased and xlm-roberta-base. For these models, the input is composed of the texts $\{t_i\}_{i=1}^{N}$ instead of the vectors $\{x_i\}_{i=1}^{N}$. For fine-tuning we used the same setup that we presented above for the classifier that we use to evaluate our embeddings. The only difference is that we set the learning rate to 2e-5.

Note that Glavaš et al. (2020) reported the results on the entire test set. In our experiments, we used a part of the test set 3 times. That is why we reproduced the experiments of that paper with mBERT and XLM-R using the same data that we used to evaluate our framework. Anyway, we observed

	GAE		
	GAO	TRAC	WUL
EN	$0.9714_{.018}$	$0.8301_{.025}$	$0.9424_{.007}$
DE	$0.9565_{.022}$	$0.9900_{.017}$	$0.9569_{.011}$
RU	$0.6667_{.014}$	$0.7238_{.010}$	$0.8453_{.014}$
TR	$0.7826_{.008}$	$0.7567_{.030}$	$0.8936_{.021}$
HR	$0.8799_{.030}$	$0.8214_{.023}$	$0.8958_{.017}$
SQ	$0.9565_{.011}$	$0.9600_{.011}$	$0.9673_{.026}$
	XLM-R		
	GAO	TRAC	WUL
EN	$0.5909_{.041}$	$0.7245_{.022}$	$0.8538_{.034}$
DE	$0.6857_{.031}$	$0.7272_{.014}$	$0.8635_{.049}$
RU	$0.5754_{.009}$	$0.7070_{.005}$	$0.8384_{.041}$
TR	$0.5171_{.011}$	$0.7371_{.017}$	$0.8199_{.036}$
HR	$0.5050_{.047}$	$0.6377_{.041}$	$0.8384_{.047}$
SQ	$0.5642_{.041}$	$0.7148_{.016}$	$0.8231_{.041}$

Table 4.1: F1 in Multi-domain Hate Speech Detection.

	All		
	GAE	XLM-R	mBERT
EN	$\mathbf{0.8333}_{.010}$	$0.5642_{.047}$	$0.5111_{.053}$
DE	$\mathbf{0.9565}_{.014}$	$0.4545_{.038}$	$0.4850_{.045}$
RU	$\mathbf{0.8333}_{.001}$	$0.4923_{.061}$	$0.4527_{.031}$
TR	$\mathbf{0.8799}_{.004}$	$0.6192_{.043}$	$0.3864_{.057}$
HR	$\mathbf{0.8461}_{.012}$	$0.6459_{.039}$	$0.4545_{.044}$
SQ	$\mathbf{0.9565}_{.002}$	$0.4978_{.021}$	$0.4457_{.046}$

Table 4.2: F1 in Multi-domain Hate Speech Detection for all domains.

that the reported results in that paper do not exceed 0.90 of F1.

Results and Discussion. Tables 4.1 and 4.2 summarizes the results of these experiments. The results obtained by using our embeddings are identified with the acronym GAE. In particular, Table 4.2 illustrates the results in the multi-domain setting per language. While Table 4.1 corresponds to the results obtained for each domain separately.

We observe two interesting points. First, GAE seems to be more stable

in data-poor settings. We verify outstanding results even in GAO, which is the domain with the least amount of data. In contrast, the smaller the dataset, the worse the results obtained with XLM-R. Notice that for all the languages, the best results of XLM-R were in the WUL domain, which is the larger one. This confirms the findings of Yao et al. (2019b), where the authors use a model based on a convolutional graph neural network for text classification, and point out that the improvement over state-of-the-art models becomes more prominent as the percentage of training data is lower.

On the other hand, we note that XLM-R and mBERT obtain the worst results in the multi-domain setting. This suggests that heterogeneous data can affect the performance of these models. The behavior is different for GAE. Although we do not see any gains moving from the mono-domain, we do not observe a considerable decrease. This suggests the suitability of our framework to deal not only with data-poor settings but also with heterogeneous data.

4.7 Multilingual Evaluation

In order to evaluate the multilingual perspective, we consider the combination of all the languages. Therefore, the set of texts for the input of the auto-encoder is composed of six different languages per domain i.e $\{t_i\}_{i=1}^{N} = \cup_{m \in \mathrm{L}} \{t_i^m\}_{i=1}^{6 \times S}$, where S is the number of samples in the specific domain and $\mathrm{L} = \{EN, DE, RU, TR, HR, SQ\}$. Then, we use three datasets, one per domain. The size of the dataset in GAO is 594 (6*99), in TRAC is 1800 (6*300), and in WUL is 3600 (6*600).

For comparison, we use mBERT and XLM-R as in the multi-domain evaluation.

Results and Discussion Table 4.3 illustrates the results of the multilingual evaluation. We observe that mBERT and XLM-R outperform the classifier that uses our embeddings. In fact, the results obtained with our framework decrease considerably. This makes sense since no knowledge about the difference between the languages has been added to the learning of our embeddings. Whereas mBERT and XLM-R have been trained with large data collections that include the languages of the datasets.

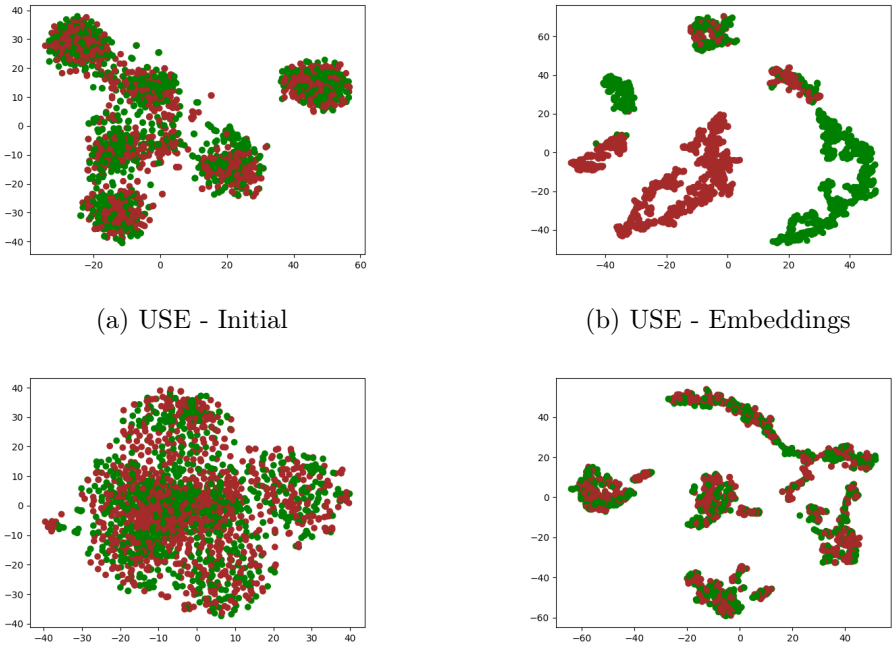
For this reason, we use a strategy to incorporate language knowledge in the input of our graph auto-encoder with the embeddings from the Universal Sentences Encoder (USE)[3] (Cer et al., 2018a). The results of this new variant (GAE-USE) are also shown in Table 4.3. In this way, the use of the embeddings once again improves the results obtained with mBERT and XLM-R.

[3]https://tfhub.dev/google/universal-sentence-encoder/4

Domain	GAO	TRAC	WUL
GAE	$0.3972_{.090}$	$0.6858_{.062}$	$0.6255_{.058}$
GAE-USE	$\mathbf{0.9308}_{.011}$	$\mathbf{0.9598}_{.005}$	$\mathbf{0.9491}_{.021}$
mBERT	$0.7047_{.054}$	$0.7952_{.080}$	$0.8939_{.072}$
XLM-R	$0.7349_{.015}$	$0.8585_{.012}$	$0.9303_{.008}$

Table 4.3: F1 in Multilingual Hate Speech Detection.

Adding Language Knowledge. In order to add information about the languages, we change the strategy to represent $\{t_i\}_{i=1}^{N}$ as $\{x_i\}_{i=1}^{N}$. In this case, instead of using TFIDF, we use the multilingual model of the universal sentences encoder (USE)[4]. This model was trained on several data sources and tasks to dynamically adapt a wide variety of natural language understanding tasks. The input is a text of variable length and the output is a 512 dimensional vector.



(a) USE - Initial

(b) USE - Embeddings

(c) TFIDF - Initial

(d) TFIDF - Embeddings

Figure 4.5: Multilingual Representation with t-SNE for the TRAC domain.

Figure 4.5 illustrates the representation of the initial vectors on the left and the latent representation (embeddings) on the right. This only corresponds to the case of TRAC, but we observed similar behavior in the other

[4]https://tfhub.dev/google/universal-sentence-encoder-multilingual-large/3

two domains. The two first Figures 4.5a and 4.5b show the representation when we add knowledge about languages with USE. In the initial representation, we observe six groups (marked with red circles) that suggest the six different languages that the input contains. In the latent representation, we observe a division among classes, similar to the ones obtained in the monolingual evaluation. Therefore, it seems that our framework can be useful to deal with multilingual datasets by adding prior knowledge of the languages.

On the other hand, Figures 4.5c and 4.5d show the representations obtained by using TFIDF. That is the case where no knowledge about languages is added. An interesting phenomenon in the embeddings generated with our encoder is that separated groups of points with both types of points (red and green) can be identified. It seems that the encoder has learned the difference among languages instead of the difference between the classes hate and not hate. In this sense, we note that this representation is somewhat similar to the initial representation obtained with USE. Therefore, the embeddings learned with our framework could improve by adding more layers to the encoder. Therefore, we attempt to adapt the proposal of Li et al. (2019) as future work. In that paper, the authors present a way to successfully train a very deep convolutional graph neural network.

4.8 Conclusion and Future Work

In this work, we proposed a graph auto-encoder framework to learn embeddings of a set of texts in an unsupervised way. The auto-encoder receives an initial vector representation of the texts and the relation among them in the form of a graph, to generate a low-dimensional representation. Then, the embeddings are extracted from this latent representation. In this sense, we built the encoder with a sequence of a transformer layer and a convolutional layer to consider the information of the graph structure in the learning of the embeddings. We used this framework for hate speech detection by using the embeddings as input of a classifier. In the evaluation, we considered multi-domain and multilingual settings with small datasets. We observed promising results by outperforming mBERT and XLM-R, one of the state-of-the-art models in hate speech detection. We noticed that the improvement by using the embeddings generated with our auto-encoder became more notable in small data, suggesting the suitability of our proposal to deal with data-poor settings. Moreover, we observed that the use of our embeddings was more stable in multi-domain settings. In the case of multilingual settings, we had to add prior knowledge about languages to avoid a decrease in performance.

As future work, we will extend our analysis by building a deeper encoder. The idea is to investigate if it is possible to learn the embeddings for a multilingual setting without the need of prior knowledge about the languages.

Moreover, we will adapt our graph auto-encoder to encode not only text but also visual information. Thus, we will be able to deal with multimodal hate speech detection.

Acknowledgements

This research work was partially funded by the Spanish Ministry of Science and Innovation under the research project MISMIS-FAKEnHATE on Misinformation and Miscommunication in social media: FAKE news and HATE speech (PGC2018-096212-B-C31). The first author gratefully acknowledges the support of the Pro^{2}Haters - Proactive Profiling of Hate Speech Spreaders (CDTi IDI-20210776) and XAI-DisInfodemics: eXplainable AI for disinformation and conspiracy detection during infodemics (MICIN PLEC2021-007681) R&D grants. The work of the first author was also partially funded by the Centre for the Development of Industrial Technology (CDTI) of the Spanish Ministry of Science and Innovation under the research project IDI-20210776 on Proactive Profiling of Hate Speech Spreaders - PROHATER (Perfilador Proactivo de Difusores de Mensajes de Odio). Moreover, the work of the second author was partially funded by the Generalitat Valenciana under DeepPattern (PROMETEO/2019/121).

Chapter 5

Convolutional Graph Neural Networks for Hate Speech Detection in Data-Poor Settings

Published in:

- **De la Peña Sarracén, G.L.** and Rosso, P. (2022). Convolutional Graph Neural Networks for Hate Speech Detection in Data-Poor Settings. International Conference on Applications of Natural Language to Information Systems. Springer International Publishing, (pp. 16–24). **(Core C Conference)**

Abstract. Hate speech detection has received a lot of attention in recent years. However, there are still a number of challenges to monitoring hateful content on social media, especially in scenarios with little data. In this paper, we propose HaGNN, a convolutional graph neural network that is capable of performing an accurate text classification in a supervised way with a small amount of labeled data. Moreover, we propose Similarity Penalty, a novel loss function that considers the similarity among nodes in the graph to improve the final classification. Particularly, our goal is to overcome hate speech detection in data-poor settings. As a result, we found that our model is more stable than other state-of-the-art deep learning models with little data in the considered datasets.

5.1 Introduction

Hate speech detection is a prominent task in Natural Language Processing and other disciplines. According to Fortuna and Nunes (2018b), which is a reference survey in the area, hate speech can be defined as a language that attacks or diminishes, that incites violence or hate against groups, based on specific characteristics such as physical appearance, religion, gender identity or other, and it can occur with different linguistic styles, even in subtle forms or when humor is used. Due to its negative real-life implications, a number of proposals to face the problem have emerged in the last few years. Among them, deep learning has gained significant traction, highlighting the state-of-the-art performance of the transformer-based models (Isaksen and Gambäck, 2020). However, hate speech is a complex phenomenon and human annotation is not straightforward, since there is no uniformity across all demographics. Then, expert-based datasets are usually small, especially in low-resource languages.

In order to deal with this limitation, we use a strategy based on graph neural networks (GNNs) which have been effective at tasks thought to have a rich relational structure, since they can preserve global structure information of a graph in embeddings (Zhou et al., 2020). In this sense, our idea is to represent the texts from a dataset as nodes in a graph and learn embeddings in terms of neighborhood aggregation. Thus, we do not need a large amount of data, such that we make use of limited labeled data by allowing information propagation through our automatically constructed graph.

The motivation derives from the strong representation learning capability of GNNs, which have gained practical significance in several applications. In general, GNNs generalize the deep neural network models to graph structured data, providing a way to effectively learn representations for graph-structured data either from the node level or the graph level. Wu et al. (2021) provide a practical overview of the different types of GNNs by presenting a taxonomy that divides them into four categories: recurrent graph

neural networks, convolutional graph neural networks, graph auto-encoders, and spatial-temporal graph neural networks. We focus on convolutional graph neural networks (CGNNs) which redefine the notion of convolution for graph data (Kipf and Welling, 2017a). The main idea is to generate a representation for each node by aggregating its features and the features of its neighbors. Then, high-level node representations are extracted by stacking multiple graph convolutional layers. The use of this type of GNN is inspired by Yao et al. (2019b) that proposed a graph representation of documents and words together, and showed an improvement of GNNs over other methods with small training sets.

Our Contributions: The novelty of this work is three-fold. First, we propose a model based on CGNNs for text classification in a data-poor setting. Particularly, we study the case of hate speech detection where it is often difficult to obtain an expert-based large dataset due to the complexity of the task. Secondly, we propose a loss function to improve the final embeddings of the nodes in the graph by penalizing the closeness among nodes of different classes. Finally, we provide a comparison of HaGNN and other models. We show that our model is robust with a small amount of data, outperforming state-of-the-art models in these few data scenarios[1].

5.2 HaGNN Model

In this section, we formalize CGNNs and describe the way we use them in our system, followed by other details of our proposed loss function.

5.2.1 Hate Speech Detection

In this work, we formalize hate speech detection as a binary classification, such that the task involves the classes hate and not hate. The data comprises N samples, where each sample is given by $\{t_i, y_i\}$. The set $\{t_i\}_{i=1}^{N}$ is composed of texts that are represented with numeric feature vectors $\{x_i\}_{i=1}^{N}$. In order to generate these feature vectors we use the universal sentences encoder (USE)[2] (Cer et al., 2018b), which encodes text into high-dimensional vectors. The model was optimized for greater-than-word length texts, such as sentences or short paragraphs. It was trained on several data sources and tasks to dynamically adapt a wide variety of natural language understanding tasks. The input is an English text of variable length and the output is a 512 dimensional vector. The set $\{y_i\}_{i=1}^{N}$ is composed of the labels 0 and 1, which indicate the presence or not of hate in each of the texts in $\{t_i\}_{i=1}^{N}$. Then, the aim of the task is to detect hateful content by assigning one of the labels to each t_i by using x_i.

[1] We will make our codes freely available by the publication date of this work

[2] https://tfhub.dev/google/universal-sentence-encoder/4

Our goal is to obtain an accurate performance in hate speech detection when N is small. We address the issue by adapting CGNNs from a node level classification. Following, we describe the CGNN model and the loss function used.

5.2.2 Background: Convolutional Graph Neural Networks

Graph neural networks are models based on deep learning for graph-related tasks in an end-to-end manner. In particular, a CGNN redefines the notion of convolution for graph data. This is a multi-layer neural network that operates directly on a graph and induces the embedding vectors of nodes based on the properties of their neighbors. Formally, let $G = (V, E)$ be a graph, where V and E represent the set of nodes and edges respectively. Let $X \in \mathbb{R}^{|V| \times d}$ be a matrix containing the features of the nodes, such that the $i-th$ row is a d-dimensional feature vector of the $i-th$ node. Moreover, let $A \in \mathbb{R}^{|V| \times |V|}$ be a matrix representation with a representative description of the graph structure, such as the adjacency matrix.

Then, CGNN takes as input the matrices X and A to learn a function of features on G and produces a node-level output ϕ. That is a $|V| \times d'$ feature matrix, where d' is the number of output features per node. A hidden layer in a CGNN can be defined as a function $H_i = \mathrm{f}(H_{i-1}, \mathrm{A}))$, where $H_0 = X$, $H_L = \phi$, L is the number of layers, and $f(\cdot, \cdot)$ is a propagation rule. Thus, the feature vectors become more abstract at each consecutive layer.

Kipf and Welling (2017a) introduced the propagation rule (5.1). Where W_i is the weight matrix for the $i-th$ layer and $\sigma(\cdot)$ is an activation function. The matrix $\hat{A}$ contains self-connections to aggregate, for each node, not only the information from its neighbors but also the node itself. It is done by adding the identity matrix I, that is $\hat{A} = A + I$. Furthermore, the matrix D is the diagonal node degree matrix of $\hat{A}$, which is used for a symmetric normalization to deal with the problem of changing the scale of the feature vectors.

$$f(H_{i-1}, A) = \sigma(D^{-\frac{1}{2}} \hat{A} D^{-\frac{1}{2}} H_i W_i) \tag{5.1}$$

5.2.3 Our Model

In order to generate the input for the model, we build the matrix X with the set of numeric feature vectors $\{x_i\}_{i=1}^{N}$, such that each vector is a row in X. On the other hand, we build the edges among nodes, to generate the matrix A, based on the inner product of the feature vectors. Then, the weight of each edge is defined by the inner product between the original vectors. We only add edges between node pairs with values higher than a threshold.

Once the matrices are generated, we feed our model. This consists of 2 convolutional layers with the propagation rule (5.1) and the ReLU as the

activation function. Moreover, we add a normalization layer after each convolutional layer. Finally, we add two linear transformation layers and a softmax to obtain the nodes classification, as Equations (5.2), (5.3) and (5.4), where $A^* = D^{-\frac{1}{2}}\hat{A}D^{-\frac{1}{2}}$.

$$H_1 = ReLU(A^*XW_0) \tag{5.2}$$

$$H_2 = ReLU(A^*H_1W_1)W_1^L \tag{5.3}$$

$$Z = softmax(H_2W_2^L) \tag{5.4}$$

In each case W_i corresponds to the parameters of the convolutional layers and W_i^L are the parameters of the linear layers.

5.2.4 Proposed Loss: Similarity Penalty

The loss function is defined using the binary cross-entropy $\mathcal{CE}$ over the labeled samples. In addition, we introduce a novel loss function which is a combination of the $\mathcal{CE}$ and a function $\mathcal{DP}$ that considers the closeness among nodes in the graph. Equation (5.5) presents this combination, where θ represents the set of all the parameters of the model. The idea with $\mathcal{DP}$ is to penalize each pair of nodes from different classes with a high similarity between their generated embeddings. We use as the generated embedding ϕ the output of the last convolutional layer, and the cosine function to calculate the similarity between vectors. As Equation (5.8) illustrates, we rely on the function $g(x) = 1 - log(x + 1)$ which is positive and decreasing in the interval of the similarity values. The term $|y_i - y_j|$ ensures only penalization for the pair of nodes from different classes by multiplying by zero the cases of pairs of vectors from the same class.

$$\mathcal{L}(\theta) = m\_\mathcal{CE} + m\_\mathcal{DP} \tag{5.5}$$

$$m\_\mathcal{CE} = \frac{1}{N}\sum_n \mathcal{CE}(\theta, x_n, y_n) \tag{5.6}$$

$$m\_\mathcal{DP} = \frac{2}{N(N-1)}\sum_i \sum_{j>i} |y_i - y_j| log\_dist(x_i, x_j) \tag{5.7}$$

$$log\_dist(x_i, x_j) = 1 - log(dist(\phi(x_i), \phi(x_j)) + 1) \tag{5.8}$$

5.2.5 Training the Model

The training is also based on Kipf and Welling (2017a) which describes a semi-supervised classification. In this sense, we divide the data into labeled (90%) and unlabeled (10%) texts. The aim is to make use of both labeled and unlabeled examples. That is, the training knows all the nodes, but not all the labels. Then, CGNN produces a latent feature representation of each node by aggregating both the labeled and unlabeled neighbors of each node during convolution, and the weights shared across all nodes are updated by propagating backward the loss calculated from the labeled examples.

5.3 Experiments

We illustrate the performance of our HaGNN model with two datasets built for hate speech detection: HatEval (Basile et al., 2019b) and CONAN (Chung et al., 2019). The second one has the characteristic that non-hateful texts are counter-narrative to hate speech, which makes it interesting to discover how CGNNs can separate both types of texts.

HatEval was Task 5 in SemEval 2019 about the detection of hate speech against immigrants and women in Spanish and English tweets. The corpus is composed of 10,000 tweets in English. The tweets were collected by three strategies: monitoring potential victims of hate accounts, downloading the history of identified haters, and filtering tweets with three groups of terms: neutral keywords, derogatory words against the targets, and highly polarized hashtags. The first task was to detect hate speech and then to identify further features in hateful content such as whether each text was aggressive or not.

CONAN is a large-scale and multilingual corpus of hate speech/ counter-speech pairs. This corpus contains texts in English, French, and Italian, and the pairs were collected through nichesourcing to three different non-governmental organizations. Both the hate speech and the responses are expert-based. We only use the 3864 pairs in English that we downloaded from the web.[3]

For the hyperparameters setting we searched in the set $\{16, 32, 64\}$ for the size of both convolutional and linear layers, in $\{0.3, 0.5, 0.7\}$ for the threshold used in the generation of the matrix A, and in $\{0.0001, 0.001, 0.01, 0.1\}$ for the learning rate. Finally, we set the threshold to 0.5, the size of the hidden layers to 32, and we use the Adam optimizer with a learning rate of 0.01. We trained the model with 200 epochs and the strategy of early stopping with patience 10. We report the results obtained with this last configuration of hyperparameters, using the cross-validation strategy with 3 partitions. Moreover, we evaluated different number of convolutional layers and observed an improvement by using two layers instead of only one. How-

[3] https://github.com/marcoguerini/CONAN

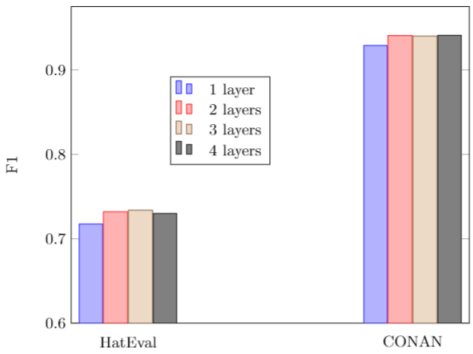


Figure 5.1: Varying number of layers

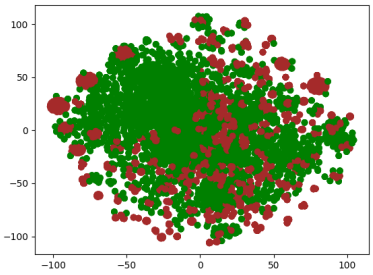
ever, we observed that the results remained similar for a number of layers greater than two as Figure 5.1 shows.

In order to compare with other models we evaluated other classifiers. The first one is based on BERT (Devlin et al., 2019c) and the other one is based on ALBERT (Lan et al., 2020). These are transformer-based models with state-of-the-art results, not only in text classification but also in many other tasks. Furthermore, we evaluated a feedforward neural network (FFNN) of 2 layers with the same input that we use for HaGNN. We aim to analyze if the performance improvement is attributed to the proposal beyond the sentence embedding.

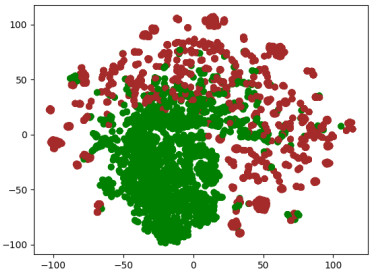
5.4 Results

In order to analyze the embeddings generated with the CGNN, Figures 5.2a and 5.2b illustrate a visualization of CONAN, and 5.2c and 5.2d for HatEval with t-SNE (Pezzotti et al., 2017a,b). We observe the effectiveness of the convolutions since in the last layer (2nd) the embeddings are more distinguishable between classes than between the original vectors. Similar variations in embedding representations are obtained for HatEval. On the other hand, Table 5.1 shows the average of F1 and standard deviation obtained with our model. The results of classification are slightly higher, but not significant for CONAN by using our loss function. However, for HatEval, we can see an important improvement. Moreover, we observe an improvement in comparison to FFNN, where we use the same sentence embedding but change the model. This shows the suitability of our proposal.

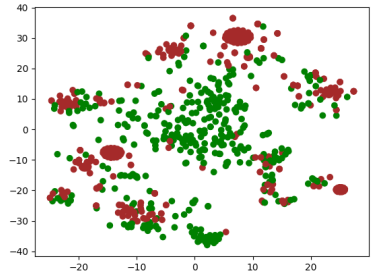
Furthermore, Figures 5.3a and 5.3b show a comparison among HaGNN, BERT, and ALBERT for HatEval and CONAN respectively. We note that HaGNN obtains a better F1 with few data. Such that, with only 100 samples, it achieves 0.8148 in CONAN, while the other models obtain less than 0.62. In HatEval, the results obtained by HaGNN with 100 samples are not so high,



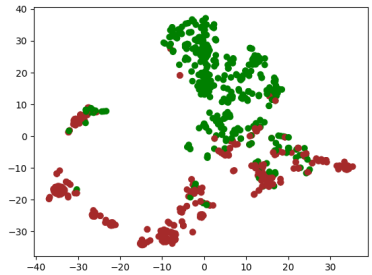
(a) Original vectors of CONAN



(b) Last layer of CONAN



(c) Original vectors of HatEval



(d) Last layer of HatEval

Figure 5.2: Embeddings

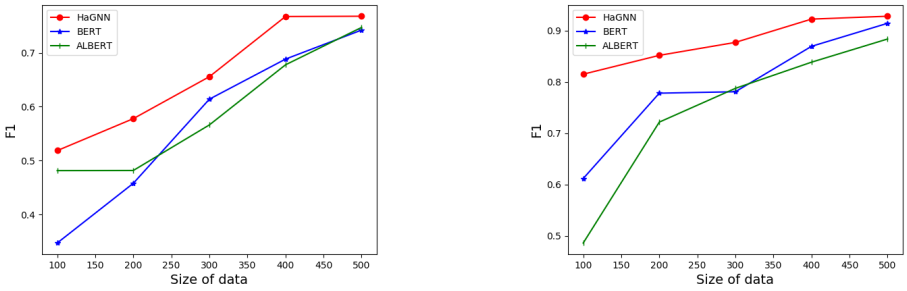
model	HatEval	CONAN
HaGNN	$0.7320_{0.0165}$	$0.9407_{0.0302}$
HaGNN + DP	$0.7500_{0.0170}$	$0.9499_{0.0231}$
FFNN	$0.7094_{0.0204}$	$0.8925_{0.0319}$
BERT	$0.7200_{0.0189}$	$0.9354_{0.0204}$
ALBERT	$0.7208_{0.0248}$	$0.931_{0.02505}$

Table 5.1: F1 and standard deviation of HaGNN.

although are higher than the results of the other models. Moreover, we can see that as the data size increases, Bert and Albert have better performance. Such that, around the size 500 the approaches are closer.

5.5 Conclusions and Future Work

In this work, we propose the HaGNN model to address hate speech detection in scenarios with few data. The model is based on convolutional graph neural networks and we proposed a new loss function to penalize nodes from different classes with close generated embeddings. We show that HaGNN



(a) HatEval (b) CONAN

Figure 5.3: F1 score for different sizes of data

is robust in small datasets, outperforming state-of-the-art models in these scenarios. In future work, we attempt to extend this model for handling multimodal datasets.

Acknowledgements

This research work was partially funded by the Spanish Ministry of Science and Innovation under the research project MISMIS-FAKEnHATE on Misinformation and Miscommunication in social media: FAKE news and HATE speech (PGC2018-096212-B-C31). The first author gratefully acknowledges the support of the Pro^{2}Haters - Proactive Profiling of Hate Speech Spreaders (CDTi IDI-20210776) and XAI-DisInfodemics: eXplainable AI for disinformation and conspiracy detection during infodemics (MICIN PLEC2021-007681) R&D grants. The work of the first author was also partially funded by the Centre for the Development of Industrial Technology (CDTI) of the Spanish Ministry of Science and Innovation under the research project IDI-20210776 on Proactive Profiling of Hate Speech Spreaders - PROHATER (Perfilador Proactivo de Difusores de Mensajes de Odio). Moreover, the work of the second author was partially funded by the Generalitat Valenciana under DeepPattern (PROMETEO/2019/121).

Part III

Data Augmentation

In this third part, we address the problem few-shot cross-lingual transfer learning in abusive language detection. We explore data augmentation techniques to deal with the problem of data scarcity that can lead to a high estimation error in few-shot learning. These techniques are based on the principle of vicinal risk minimization that aims to increase the data in the vicinity of the few-shot samples. We explore two existing techniques and propose a variant of one of them.

Chapter 6

Vicinal Risk Minimization for Few-Shot Cross-lingual Transfer in Abusive Language Detection

Published in:

- **De la Peña Sarracén, G.L.**, Rosso, P., Litschko, R., Glavas, G., and Ponzetto, S. P. (2023). Vicinal Risk Minimization for Few-Shot Cross-lingual Transfer in Abusive Language Detection. Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP), (pp. 4069–4085) **(Core A* Conference)**

Abstract. Cross-lingual transfer learning from high-resource to medium and low-resource languages has shown encouraging results. However, the scarcity of resources in target languages remains a challenge. In this work, we resort to data augmentation and continual pre-training for domain adaptation to improve cross-lingual abusive language detection. For data augmentation, we analyze two existing techniques based on vicinal risk minimization and propose MIXAG, a novel data augmentation method that interpolates pairs of instances based on the angle of their representations. Our experiments involve seven languages typologically distinct from English and three different domains. The results reveal that the data augmentation strategies can enhance few-shot cross-lingual abusive language detection. Specifically, we observe that consistently in all target languages, MIXAG improves significantly in multidomain and multilingual environments. Finally, we show through an error analysis how the domain adaptation can favor the class of abusive texts (reducing false negatives), but at the same time, declines the precision of the abusive language detection model.

6.1 Introduction

Few-shot learning (FSL) is a machine learning paradigm that allows models to generalize from a small set of examples (Wang et al., 2020b, 2023). Unlike traditional methods, FSL does not require training a model from scratch. Instead, pre-trained models are extended with just a little information, which is useful when training examples are scarce or data annotation is expensive.

Transfer learning is popularly used in few-shot learning, where the prior knowledge from a source task is transferred to the few-shot task (Pan and Yang, 2010; Pan et al., 2019). Usually, training data is abundant in the source task, while training data is low in the target task. In natural language processing, few-shot cross-lingual transfer learning (Glavaš et al., 2020; Schmidt et al., 2022; Winata et al., 2022) is the type of few-shot transfer learning in which the source/target tasks are the same but the source/target languages are different. A pre-trained multilingual model is first fine-tuned in a high-resource language and then fine-tuned on a few data in a target language (Zhao et al., 2021).

Due to the limited availability of examples in the target language, naive fine-tuning can lead to overfitting and thus poor generalization performance on the few-shot task (Parnami and Lee, 2022). A strategy usually used to alleviate this problem, not just in the few-shot cross-lingual transfer but in FSL in general, is to increase the number of samples of the few-shot task from prior knowledge. This is the data-level approach (Chen et al., 2023), which can be divided into two categories: 1) transforming samples from the few existing examples (Arthaud et al., 2021; Zhou et al., 2022; Zhang et al., 2022) and 2) transforming samples from external datasets (Antoniou and

Storkey, 2019; Rosenbaum et al., 2022; Pana et al., 2023).

Contributions. In this work, we explore abusive language detection in seven topologically diverse languages via few-shot cross-lingual transfer learning at the data-level. Although a number of studies have examined abusive language, we aim to take advantage of resources available for English in other less explored and low-resource languages. We focus on two aspects: 1) considering languages that are typologically distinct from English and 2) with little effort. Previous works focus on languages that are similar to English, such as European languages (Stappen et al., 2020b; Nozza, 2021; Rodríguez et al., 2021; Firmino et al., 2021; Zia et al., 2022; Castillo-López et al., 2023). In contrast, we analyze languages that are more different from English. 'Little effort' refers to a consistent strategy across all languages, without requiring external resources or ad hoc processing for each particular language. The main contributions of this paper can be summarized as follows:

- *Dataset extension:* We rely on a multidomain and multilingual dataset for abusive language detection (Glavaš et al., 2020). This dataset contains texts in 5 languages which have been obtained by translating original English texts. To facilitate a more comprehensive evaluation, we extend the dataset by manually translating it into Spanish.

- *Few-shot cross-lingual transfer learning improvement at data-level:* We rely on Vicinal Risk Minimization (VRM) (Chapelle et al., 2000) to generate synthetic samples in the vicinity of the examples to increase the amount of information to fine-tune the model in the target language. In this work we use three VRM-based techniques: 1) SSMBA (Ng et al., 2020), which uses two functions to move randomly through a variety of data, 2) MIXUP (Zhang et al., 2018), which linearly combines pairs of examples to obtain new samples and 3) MIXAG, our variant of MIXUP, which controls the angle between an example and the synthetic data generated in its neighborhood.

- *Unsupervised language adaptation:* We also simulate a fully unsupervised setup, removing the label information from the target languages. In that setup, we examine a strategy to address the lack of information that zero-shot transfer (no example to fine-tune the model) faces. The general idea is to make a domain adaption for abusive terms via masked language modeling (MLM) in the target language before the zero-shot transfer.

We aim to answer the following research questions:

RQ1: What is the role of VRM-based techniques in few-shot cross-lingual abusive language detection?

RQ2: What is the impact of different languages on few-shot cross-lingual abusive language detection?

RQ3: How do VRM-based techniques fare against domain specialization for cross-lingual transfer of abusive language detection models?

6.2 Background and Related Work

In this section, we discuss the main issue of few-shot learning and how data-based approaches can alleviate it. We take the definitions from Wang et al. (2020b), where more details can be found. Then, we provide a brief overview of abusive language and align our work with recent studies focused on few-shot cross-lingual transfer approaches.

Few-Shot Learning. Few-shot learning deals with a small training set $D_{train} = \{(x_i, y_i)\}$ to approximate the optimal function f^* that maps input x to output y, given a joint probability distribution $p(x, y)$. Thus, an FSL algorithm is an optimization strategy that searches in a functions space F to find the set of parameters that determine the best $f^{'} \in F$. The performance is measured by a loss function $l(f(x), y)$ which defines the expected risk with respect to $p(x, y)$. However, $p(x, y)$ is unknown, hence the empirical risk is used instead (Fernandes de Mello et al., 2018). This is the average of sample losses over D_{train} and can be reduced with a larger number of examples. One major challenge for FSL is then the small size of D_{train}, which can lead to the empirical risk not being a good approximation of the expected risk. To alleviate this problem, an approach that exploits prior knowledge can be used (Wang et al., 2023). Data-level approach involves methods that augment D_{train} with prior knowledge (Feng et al., 2021; Bayer et al., 2022; Dai et al., 2023).

Vicinal Risk Minimization formalizes the data augmentation as an extension of D_{train} by drawing samples from a neighbourhood of the existing samples (Chapelle et al., 2000). The distribution $p(x, y)$ is approximated by a vicinity distribution $D_v = \{(\hat{x_i}, \hat{y_i})\}_{i=1}^{N_v}$, whose instances are a function of the instances of D_{train}. Vicinal risk (R_v) is then calculated on D_v as Equation 6.1.

$$R_v = \frac{1}{N_v} \sum_{i=1}^{N_v} l(f(\hat{x_i}), \hat{y_i}) \tag{6.1}$$

In this work, we study three VRM-based techniques that use different strategies to generate the vicinity distribution (see §6.4).

Abusive Language. Typically, abusive language refers to a wide range of concepts (Balayn et al., 2021; Poletto et al., 2021), including hate speech (Yin and Zubiaga, 2021; Alkomah and Ma, 2022; Jain and Sharma, 2022), profanity (Soykan et al., 2022), aggressive language (Muti et al., 2022; Kanclerz et al., 2021), offensive language (Pradhan et al., 2020; Kogilavani et al., 2023), cyberbullying (Rosa et al., 2019) and misogyny (Shushkevich and Cardiff, 2019). Pamungkas et al. (2023) overview recent research across domains and languages. They identify that English is still the most widely studied language, but abusive language datasets have been extended to other

languages, including Italian, Spanish and German (Corazza et al., 2020b; Mamani-Condori and Ochoa-Luna, 2021; Risch et al., 2021). In addition, we have found studies for other languages such as Arabic (Khairy et al., 2021), Danish (Sigurbergsson and Derczynski, 2020), Dutch (Caselli et al., 2021b), Hindi (Das et al., 2022), Polish (Ptaszynski et al., 2019) and Portuguese (Leite et al., 2020). Regardless, some works like (Stappen et al., 2020b) state that there is a need to extend the resources for diverse and low-resource languages. To cover this problem, Glavaš et al. (2020) propose a multidomain and multilingual evaluation dataset. They show that language-adaptive additional pre-training of general-purpose multilingual models can improve the performance in transfer experiments. These are promising results, and although there are works like (Pamungkas et al., 2023) that cite this dataset, we have not found works that exploit it. In this work, we extend the study of the original work (Glavaš et al., 2020) to assess strategies for enhancing the performance of abusive language detection in low-resource languages.

Cross-Lingual Abusive Language Detection. In recent years, cross-lingual abusive language detection has gained increasing attention in zero-shot (Eronen et al., 2022) and few-shot (Mozafari et al., 2022) transfer. Pamungkas and Patti (2019) propose a hybrid approach with deep learning and a multilingual lexicon for cross-lingual abusive content detection. Ranasinghe and Zampieri (2020b) use English data for cross-lingual contextual word embeddings and transfer learning to make predictions in languages with fewer resources. More recently, Mozafari et al. (2022) propose an approach based on meta-learning for few-shot hate speech and offensive language detection in low-resource languages. They show that meta-learning models can quickly generalize and adapt to new languages with only a few labeled data points to identify hateful or offensive content. Their meta-learning models are based on optimization-level and metric-level. These are two approaches to improve the problem of poor data availability in few-shot learning. In contrast, we focus on the data-level approach. Unlike other works that are also based on increasing data (Shi et al., 2022), we explore VRM-based strategies for abusive language detection.

6.3 Dataset and Experimental Setup

XHate-999 (Glavaš et al., 2020) is an available dataset intended to explore several variants of abusive language detection. This dataset includes three different domains: Fox News (GAO), Twitter/Facebook (TRAC), and Wikipedia (WUL). In our work, we define ALL as the set of instances resulting from the union of all three domains. Each domain comprises different amounts of annotated data (abusive/non-abusive) in English for training,

validation, and testing (see Appendix 6.7). English test instances are translated into five target languages: Albanian (SQ), Croatian (HR), German (DE), Russian (RU), and Turkish (TR).

We extended this dataset with texts in Spanish. To generate the texts, we rely on machine translation and post-editing, following the monitored translation-based approach described in the dataset paper. Thus, slight modifications were made in the Spanish translation to reflect and maintain the level of abuse in the original English instances.

Models. We rely on mBERT (Devlin et al., 2019a) base cased with $L = 12$ transformer layers, hidden state size of $H = 768$, and $A = 12$ self-attention heads (see Appendix 6.7 for more details). First, we retrain the model with the XHate-999 training and validation sets, to obtain the model (*model_ base*) that we use in all our experiments. We search the following hyper-parameter grid: training epochs in the set $\{2, 3, 4\}$ and learning rate in $\{10^{-4}, 10^{-5}, 10^{-6}\}$. We train and evaluate in batches of 2 texts, with a maximal length of 512 tokens, and optimize the models with Adam (Kingma and Ba, 2015b). We set the random seeds to 7 to facilitate the reproducibility of experiments.

Fine-tuning and Evaluation Details. For each language, we draw 90% of instances from the test set to evaluate *model_ base*. In few-shot cross-lingual transfer experiments, we use the remaining 10% of instances to fine-tune *model_ base* before the evaluation. i.e. we use 10 instances to fine-tune *model_ base* in GAO (and 89 to evaluate), while the respective numbers are 30 (270) for TRAC, 60 (540) for WUL, and 100 (899) for ALL (GAO+TRAC+WUL). Notice that for each language, the test set used by Glavaš et al. (2020) is different from the one we use. However, we do not observe a significant difference between the use of the full test set and the use of the subset we rely on (see Appendix 6.7 to examine the results).

Statistical Analysis. In our experiments, we used McNemar's test as Dietterich (1998) recommends. This is a paired non-parametric statistical hypothesis test where the rejection of the null hypothesis suggests that there is evidence to say that the models disagree in different ways. We set the significance level to 0.05 and use $\alpha_{altered}$, obtained with the Bonferroni correction (Napierala, 2012).

6.4 Few-Shot Cross-lingual Transfer

We first examine the ability of three VRM-based techniques in few-shot cross-lingual transfer learning for abusive language detection to address **RQ1**.

6.4.1 SSMBA

Ng et al. (2020) propose SSMBA, a data augmentation method for generating synthetic examples with a pair of corruption and reconstruction functions to move randomly on a data manifold. In the corruption function, we use two strategies: 1) masking a word in each text in a random way (default) or 2) masking the salient abusive words in each text. To identify abusive words, we use HurtLex (Bassignana et al., 2018), a multilingual lexicon with harmful words. For texts that do not contain words in the lexicon, we follow strategy 1. In the reconstruction functions, we use mBERT.

6.4.2 MIXUP

(Zhang et al., 2018; Sun et al., 2020) is a VRM-based technique that constructs a synthetic example $(\hat{x_i}, \hat{y_i})$ (in the vicinity distribution) from the linear combination of two pairs (x_i, y_i) and (x_j, y_j), drawn at random from the training set D_{train} as Equation 6.2, with $\lambda \sim \beta(\alpha, \alpha)$ where α is a hyper-parameter[1].

$$\begin{aligned} \hat{x_i} &= \lambda x_i + (1-\lambda) x_j \\ \hat{y_i} &= \lambda y_i + (1-\lambda) y_j \end{aligned} \tag{6.2}$$

We rely on a **multilingual GPT model** (Shliazhko et al., 2022) (see Appendix 6.7) **for the linear combination of the texts representations** (Equation 6.3): we obtain the embedding E_w of each word of a text x_i and concatenate them to generate the vector representation $E(x_i)$. Then, we combine two texts x_i and x_j as the linear combination of their representations $E(x_i)$ and $E(x_j)$. Note that E_w is a single step of an auto-regressive model. The obtained vector is split into vectors of the same size as the original word embeddings E_w. Finally, we decode those vectors to obtain a sequence T of words, that we use as the new syntectic text $\hat{x_i}$. The linear combination of the labels $y \in \{0, 1\}$, when y_i and y_j are different depends on the value of λ. We assign 1 to $\hat{y_i}$ when the combination is greater than or equal to 0.5. Otherwise, we assign 0.

$$\hat{x_i} = T(\lambda E(x_i) + (1-\lambda) E(x_j)) \tag{6.3}$$

Procedure. This VRM-based technique is an iterative process. In each iteration, the few-shot set D_{train} is divided into pairs of samples to combine. Thus, the number of instances generated in each iteration is equal to $\frac{N}{2}$, where N is the number of samples in D_{train}. We make sure not to take the same pairs of examples in different iterations.

[1]We tried some values different from 1 for α and MIXUP was not sensitive to variation, so we set it to 0.2.

6.4.3 MIXAG

Motivated by the idea of MIXUP, **we propose the variant MIXAG: mix vectors with a focus on the AnGle between them**. We hypothesize that the distance between an example and the new synthetic examples may be relevant to generating an effective vicinity. As this aspect cannot be easily controlled in the original MIXUP, we propose a particular case that interpolates pairs of instances based on the angle of their representation.

The idea is to define a linear combination (Equation 6.4) with the parameter λ as a function of the angle α between the original vectors x_i and x_j, as well as the angle θ between the new vector $\hat{x}$ and one of the original vectors (Figure 6.1).

$$\hat{x} = \lambda x_i + x_j \tag{6.4}$$

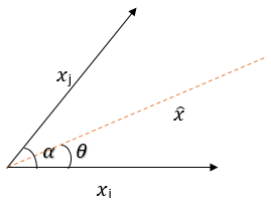


Figure 6.1: MIXAG description.

Using the Law of Sines we express λ as a function (Equation 6.5) of the cosine of α, which can be obtained with Equation 6.6, and the cosine of θ, which is the parameter of MIXAG. $||\cdot||$ denotes the norm of a vector. We refer readers to Appendix 6.7 for more details.

$$\lambda = \frac{||x_j||(cos(\theta)\sqrt{1-cos(\alpha)^2}-cos(\alpha)\sqrt{1-cos(\theta)^2})}{||x_i||\sqrt{1-cos(\theta)^2}} \tag{6.5}$$

$$cos(\alpha) = \frac{x_i x_j}{||x_i||||x_j||} \tag{6.6}$$

For MIXAG, we define the combination of texts by Equation 6.7, following the same representation and processing of texts as in MIXUP. The difference is basically in the parameter λ.

$$\hat{x_i} = T(\lambda E(x_i) + E(x_j)) \tag{6.7}$$

In this work, we set $\theta = \frac{\alpha}{2}$, thus the parameter of MIXAG is defined by Equation 6.8. We suggest extending this study to analyze how the parameter $cos(\theta)$ can influence the results.

$$cos(\theta) = \sqrt{\frac{1 + cos(\alpha)}{2}} \tag{6.8}$$

Procedure. This VRM-based technique is also an iterative process. In this case, we randomly select a sample x_i from D_{train} and create the pairs with x_i and each of the rest of the samples of D_{train}. Therefore, the number of instances generated in each iteration is $N - 1$, where N is the number of samples in D_{train}.

6.4.4 Multilingual MIXUP/MIXAG

By default, in MIXUP and MIXAG we use the few-shot set D_{train} of each language to generate new instances for that particular language. Alternatively, we use the union of the D_{train} of all languages. For each pair of original texts x_i and x_j, we make sure that x_i is from the language in the analysis, while x_j is a text from any language.

6.4.5 Multidomain MIXUP/MIXAG

We rely on training data for GAO, TRAC, and WUL, as well as ALL (WUL+ TRAC+GAO) in all monolingual and multilingual experiments. In short, we analyze performance when training and testing 1) only on a particular domain (for example, when testing on GAO we train only on GAO training data) and 2) on all available data from all three data sets (multidomain setup).

6.4.6 Results and Analysis

A summary of cross-lingual transfer results for the variants - few-shot and few-shot with SSMBA, MIXUP, and MIXAG - is provided in Figure 6.2.

As expected, we observed that VRM-based techniques improve the performance of few-shot cross-lingual transfer in most cases. There is no clear difference between the VRM-based techniques, but we can see interesting results that vary depending on the domain. In the GAO domain, all three techniques seem to have similar results across languages. In TRAC, MIXUP seems to be slightly better than MIXAG in most languages. However, the critical result in this domain is that SSMBA fails to improve the few-shot cross-lingual transfer. In contrast, SSMBA seems to be the best technique in WUL. We believe that these results are due to the nature of the texts in each domain. TRAC contains texts from Twitter and Facebook. We speculate that the reconstruction function of SSMBA affects the quality of the vicinity generated for each text by introducing terms that differ from common terms in this domain. On the other hand, WUL contains text from Wikipedia, which supports our assumption.

Multidomain. Table 6.1 shows the results for all the variants of the VRM-based techniques. We illustrate and analyze the results for the combination of all domains.

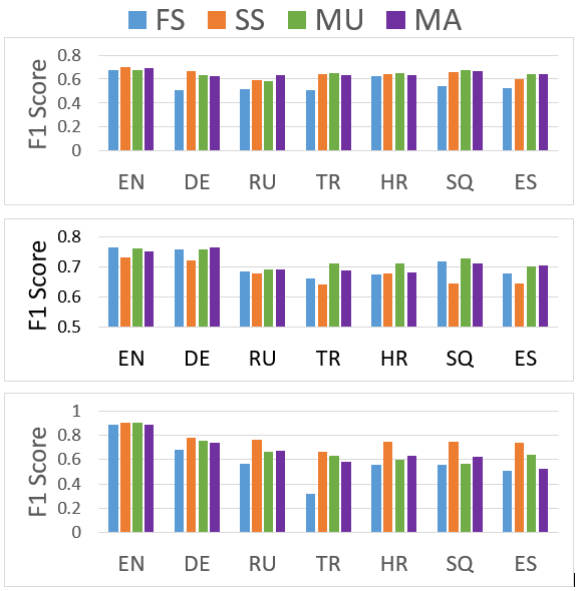


Figure 6.2: Performance with mBERT of few-shot (FS) cross-lingual transfer and the variants: SSMBA (SS), MIXUP (MU), and MIXAG (MA). **Upper Figure:** GAO domain, **Middle Figure:** TRAC domain and **Lower Figure:** WUL domian

All languages except German seem to benefit from few-shot cross-lingual transfer w.r.t. zero-shot cross-lingual transfer. Likewise, the few-shot cross-lingual transfer is improved with VRM-based techniques as in the results by domain.

SSMBA improves few-shot cross-lingual transfer in all languages except English. In this heterogeneous domain, we do not observe the problem that SSMBA has in TRAC. On the other hand, the use of HurtLex does not seem to be a relevant strategy, since the results are similar to those obtained with the default strategy (random selection). This is an encouraging result, which suggests that we can use SSMBA to improve few-shot cross-lingual transfer learning without relying on external resources.

MIXUP seems to be better than SSMBA and MIXAG for most languages. However, multilingual MIXAG is significantly the best strategy. This is a good indicator of the benefits of our variant for multidomain and multilingual environments. Note that the multilingual strategies outperform the rest of the variants and that particularly, multilingual MIXAG consistently performs better than multilingual MIXUP. This suggests that our hypothesis about the implication of controlling the angle between the original texts and the new synthetic texts seems to be relevant in multilingual data.

Finally, we combine MIXUP/MIXAG with SSMBA: First, we augment the data with SSMBA and then augment the vicinity with MIXUP/MIXAG.

ALL	EN	DE	RU	TR	HR	SQ	ES
ZS	0.8085	0.7156	0.6308	0.3627	0.6214	0.6127	0.6008
FS	0.8112	0.7141	0.6329	0.4063	0.6316	0.6238	0.6130
SS	0.8077	0.7253	0.7071	0.6568	0.6965	0.6990	0.6838
SS-HL	0.8097	0.7273	0.6987	0.6689	0.6725	0.6909	0.6973
MU	0.8102	0.7404	0.7013	0.6740	0.7116	0.7001	0.6878
MMU	**0.8284**	0.7500	0.7312	0.7113	0.7371	0.7128	0.7250
MU-SS	0.8176	0.7531	0.7233	0.6839	0.7186	0.6881	0.7087
MA	0.8083	0.7245	0.6757	0.5616	0.6710	0.6788	0.6508
MMA	0.8237	**0.7585**	**0.7392**	**0.7224**	**0.7523**	**0.7344**	**0.7476**
MA-SS	0.8096	0.7229	0.7193	0.6369	0.6759	0.6734	0.6713

Table 6.1: Zero-shot (ZS) and few-shot (FS) cross-lingual transfer performance with mBERT on the union of all domains. We also show 8 variants for FS: 1) SSMBA (SS) and 2) SSMBA with HurtLex (SS-HL), 3) MIXUP (MU), 4) multilingual MIXUP (MMU), 5) MIXUP with SSMBA (MU-SS), 6) MIXAG (MA), 7) multilingual MIXAG (MMA), 8) MIXAG with SSMBA (MA-SS). The results ($\alpha_{altered} = .005$) are reported in terms of F1 and significantly better results are underlined for each language and domain. Numbers in bold indicate the best results.

The results are also shown in Table 6.1. This strategy offers some improvement over MIXUP/MIXAG in most cases.

Correlation Analysis. Thus far, we have observed that the behavior of the strategies seems quite similar across languages. For instance, the few-shot cross-lingual transfer is outperformed with the VRM-based techniques. This motivates us to investigate **RQ2**, i.e. we examine if there is a high correlation between the performance of few-shot cross-lingual transfer (and its variants with VRM-based techniques) and the linguistic proximity scores of each language to English.

We analyze the correlation between the performance of the strategies that we use for cross-lingual transfer learning and the distance between each language and English. We rely on the tool LANG2VEC[2] which proves language vectors that encode linguistic features from the URIEL database (Littell et al., 2017). We obtain the vector representation of the languages with 4 features: 1) SYN: encodes syntactic properties, 2) FAM: encodes memberships in language families, 3) INV: denotes the presence of natural classes of sounds, and 4) PHO: encodes phonological properties.

Then, with the vectors from each linguistic feature, we calculate the cosine similarity between each language and English. Finally, we calculate the Pearson correlation coefficients (Sedgwick, 2012) between the cosine similar-

[2] https://github.com/antonisa/lang2vec

ity and the performance of each cross-lingual strategy across languages and domains.

	SYN	FAM	INV	PHO
FS	0.664	0.661	0.607	0.516
SS	0.527	0.627	0.608	0.486
MU	0.405	0.628	0.633	0.463
MA	0.571	0.721	0.686	0.529

Table 6.2: Pearson correlation coefficients between linguistic proximity scores (features SYN, FAM, INV, PHO) and few-shot (FS), few-shot with SSMBA (SS), few-shot with MIXUP (MU) and few-shot with MIXAG (MA) cross-lingual transfer performance with mBERT across all languages and domains.

Table 6.2 shows the correlation coefficients for the significant linguistic features with a significance level of 0.05 (Appendix 6.7 shows the correlation coefficients for all metrics and the similarity scores between each language and English). Coefficients whose magnitude is between 0.5 and 0.7 indicate a moderate correlation, while coefficients between 0.3 and 0.5 indicate a low correlation.

We only observe a moderate correlation between the performance of each strategy and the distance between the target languages and English. We consider these results encouraging because they suggest that the strategies are possibly consistent across languages.

6.4.7 Ablation Studies

MIXAG is a data augmentation method that randomly combines inputs and accordingly combines one-hot-label encodings. This is a variant of MIXUP where the new data is obtained by defining the angle between the inputs and the new instance.

In our strategy, we randomly select pairs of inputs and set the angle between the new instance and one of the inputs as $\theta = \frac{\alpha}{2}$, where α is the angle between the original inputs. However, there are other strategies that could be used. For example, selecting data pairs whose latent representations are close neighbors, as well as defining other values for θ. To compare MIXAG with these alternative possibilities, we run a set of ablation study experiments using not only mBERT but also the XLM-R model (Conneau et al., 2020b). We focus on multilingual and multimodal MIXAG (MMA in ALL) as it is the best data augmentation method that we observed in the first experiments.

On the one hand, we compare the combination of random pairs of inputs with the combination of nearest neighbors (NN). On the other hand, we set the angle $\theta = \frac{\alpha}{3}$ to evaluate the impact of varying this parameter on the

performance of the method. Finally, we use an alternative model for the text representation. Specifically, we used the multilingual generative model mT0 (Muennighoff et al., 2023), instead of mGPT.

From the results of the ablation study in Table 6.3, we have the following observations. First, there are no significant differences with $\alpha = .05$ between the variants studied, although experiments with XLM-R seem to have shown some improvement. Secondly, we note that the variation of the angle between the inputs and the generated instances does not seem to represent a relevant factor.

model	variant	EN	DE	RU	TR	HR	SQ	ES
mBERT	MMA	0.8237	0.7585	0.7392	0.7224	0.7523	0.7344	0.7476
	MMA-NN	0.8233	0.7585	0.7201	0.6473	0.7523	0.7273	0.7466
	MMA-ANG	0.8233	0.7475	0.7169	0.6774	0.7415	0.7273	0.7466
	MMA-MT0	0.8238	0.7585	0.7392	0.7224	0.7523	0.7344	0.7476
	MMA-MT0-NN	0.8254	0.7687	0.7314	0.6696	0.7477	0.7314	0.7528
XLM-R	MMA	0.8236	0.7927	0.7561	0.7258	0.7180	0.7780	0.7670
	MMA-NN	0.8251	0.7942	0.7328	0.7267	0.7180	0.7797	0.7650
	MMA-ANG	0.8251	0.7940	0.7521	0.7267	0.7216	0.7797	0.7650
	MMA-MT0	0.8245	0.7952	0.7503	0.7281	0.7243	0.7798	0.7658
	MMA-MT0-NN	0.8245	0.7940	0.7503	0.7297	0.7180	0.7803	0.7658

Table 6.3: Results of the ablation studies for 4 variants of multilingual MIXAG (MMA): 1) interpolation only between nearest neighbors (MMA-NN), 2) set $\theta = \frac{\alpha}{3}$ (MMA-ANG), 3) text representation with mT0 (MMA-MT0) and 4) text representation with mT0 and interpolation between nearest neighbors(MMA-MT0-NN). The results ($\alpha_{altered} = .005$) are reported in terms of F1 for each of the model mBERT and XLM-R.

All five variants obtain very similar results with mBERT. The variation of the factors that we analyze does not seem to influence the performance of the method. However, with XLM-R we observe some interesting findings. Spanish and Russian are the only languages where MMA method is not surpassed by the other variants. In the rest of the languages, we observe the opposite behavior, where text representation with the alternative model mT0 seems to be the best strategy. Notice that in Albanian the use of mT0 for text representation together with the strategy of selecting the nearest neighbor for interpolation seems to be the best variant.

6.5 Unsupervised Language Adaptation

In this section, we investigate the scenarios in which there is no information about the target language for the few-shot cross-lingual transfer. In § 6.4 we used a small amount of supervised data D_{train} in the target language to fine-tune the pre-trained model. This allowed us to adapt the model to the abusive language of each particular language. In contrast, now we assume

that the labels of D_{train} are not available. This is a simulated experiment where we only have an unlabelled set of texts and the set D_{test} in which we want to detect abusive language. Previous works have examined this scenario by adjusting a model with unlabelled external data. In this work, **we use only a few unlabelled instances from** D_{train}.

Basically, this strategy is a zero-shot cross-lingual transfer learning in which the model is adapted to the abusive terms of the target language. As mBERT is pre-trained on general-purpose and multilingual corpora, it is familiar with the target languages. However, it has not been adjusted to the particular case of abusive language. We follow then a two-step methodology: 1) continual pre-taining for domain adaptation via masked language modeling (MLM) to make it familiar to the particular abusive terms, and then 2) employ zero-shot learning to detect abusive language.

6.5.1 Results and Analysis

Table 6.4 illustrates the results obtained with the methodology across domains and languages. In most cases, the strategy of prior adaptation to the abusive terms seems to outperform zero-shot cross-lingual transfer learning. English is the only language in which the MLM adaptation worsens the results in all domains. Moreover, TRAC also shows no improvement, similar to the behavior observed with SSMBA in few-shot cross-lingual transfer.

GAO	EN	DE	RU	TR	HR	SQ	ES
ZS	**0.6747**	0.5067	0.5205	0.5116	0.6234	0.5405	0.5263
ZS_MLM	0.6050	**0.6364**	**0.6261**	**0.6341**	**0.6290**	**0.6016**	**0.6154**
TRAC							
ZS	**0.7642**	**0.7582**	**0.6815**	**0.6777**	**0.6892**	**0.7235**	**0.7000**
ZS_MLM	0.6821	0.6480	0.6718	**0.6785**	0.6785	0.6995	0.6118
WUL							
ZS	**0.8800**	0.6698	0.5561	0.2945	0.5469	0.5556	0.4960
ZS_MLM	0.6093	**0.6765**	**0.6708**	**0.6765**	**0.6732**	**0.6675**	**0.6765**
ALL							
ZS	**0.8085**	**0.7156**	0.6308	0.3627	0.6214	0.6127	0.6008
ZS_MLM	0.6662	0.6711	**0.6637**	**0.6716**	**0.6716**	**0.6721**	**0.6419**

Table 6.4: Zero-shot (ZS) and adapted zero-shot (ZS_MLM) cross-lingual transfer performance with mBERT on domains (GAO, TRAC, WUL) and the union of all domains (ALL). Results are reported in terms of F1 and numbers in bold indicate those that are significantly better for each language and domain ($\alpha = .05$).

These results allow us to answer **RQ3**: although domain adaptation can improve zero-shot cross-lingual transfer, VRM-based techniques seem to be more robust in few-shot cross-lingual transfer.

Error Analysis. In order to deepen the analysis of what happens in the model with the zero-shot cross-lingual transfer adaptation, we also analyze two metrics: Recall and Precision. Recall refers to the true positive rate and is the number of true positives divided by the total number of positive texts. Precision refers to the positive predictive value and is the number of true positives divided by the total number of positive predictions. In this work, positive refers to the class of abusive texts.

Results across domains and languages are in Appendix 6.7. In all cases we observe an increase in Recall, indicating that adapting the model could improve the proportion of the class of abusive texts that is correctly classified. At first glance, it seems to be a good result, since it is desirable to reduce the number of false negatives in abusive language detection. However, we observe that precision is reduced, suggesting that this strategy favors the positive class: while false negatives are reduced, false positives are increased.

Critical cases are negative texts that can be incorrectly detected as abusive. In order to study this phenomenon, we examine the percentage of texts that are non-abusive and are well-classified with zero-shot transfer learning and misclassified with the MLM adaptation. We investigate two statistics across languages and domains: 1) the percentage of non-abusive texts that are well- classified with zero-shot transfer and misclassified with the MLM adaptation and 2) the percentage of abusive texts that are misclassified with zero-shot transfer and well-classified with the MLM adaptation.

Table 6.5 illustrates the statistics across domains and languages. Consistent with the previous results we observe a detriment in the class of non-abusive texts. The number of negative texts well-classified with zero-shot transfer learning and misclassified with the MLM adaptation is large (reaching 100% in a case). However, that amount is surpassed in most cases by the gain in the class of abusive texts. We observe that the number of positive texts that are misclassified with zero-shot transfer learning and well-classified with adaptation via MLM is high (reaching 100% in four cases).

GAO	EN		DE		RU		TR		HR		SQ		ES	
	%P	%N	%P	%N	%P	%N	%P	%N	%P	%N	%P	%N	%P	%N
GAO	83.3	60	76.1	61.7	85.7	77.7	100	80.7	81.2	86.4	85.0	88.8	100	97.1
TRAC	81.2	74.6	82.3	83.8	91.8	88.8	88.1	92.9	87.5	80.7	100	90.5	69.2	89.3
WUL	85.7	83.8	62.2	78.0	56.8	63.8	74.5	89.3	88.3	96.1	99.3	98.8	96.7	98.8
ALL	87.2	83.9	76.6	84.1	100	100	100	88.4	88.9	94.5	99.5	99.4	83.9	93.3

Table 6.5: Percentage of non-abusive texts that are well-classified with zero-shot transfer and misclassified with the MLM adaptation (%N), and percentage of abusive texts that are misclassified with zero-shot transfer and well-classified with the MLM adaptation (%P).

6.6 Conclusions and Future Work

In this work, **we studied three techniques to improve few-shot cross-lingual transfer learning in abusive language detection**. These techniques are concentrated on data-level approach to deal with the problem of data scarcity that can lead to a high estimation error in few-shot learning. Specifically, **we focused on vicinal risk minimization techniques** to increase the data in the vicinity of the few-shot samples. First, we explored two existing techniques: 1) SSMBA, which is based on a pair of functions to corrupt and reconstruct texts, and 2) MIXUP, which generates new samples from a linear combination of original instances pairs. Then, **we proposed MIXAG, a variant of MIXUP, to parameterize the combination of instances with the angle between them**. Our experiments were based on the multidomain and multilingual dataset XHATE-999, which allowed us to explore low-resource languages as target languages and English as the base language. This dataset contains six different languages, and we extended it to Spanish, following the same methodology that was used to generate the texts of the other languages. The results showed the effectiveness of VRM-based techniques to improve few-shot cross-lingual transfer learning in most domains and languages. Particularly, we observed that multilingual MIXAG outperforms the other strategies in the heterogeneous set (multidomain) for all target languages. At the same time, we observed that structural language similarity does not seem to be highly correlated with cross-lingual transfer success in none of the strategies. These results are encouraging for abusive language detection in low-resource settings, as the strategies that we have examined appear to be consistent across languages.

Finally, we evaluated a scenario where it is not possible to perform a few-shot cross-lingual transfer due to the lack of supervised information. We used a strategy based on masked language modeling and saw a degradation in the class of non-abusive texts, but a gain in the class of abusive texts, reducing false negatives.

In future work, we aim to further examine our proposed VRM-based technique for data augmentation. MIXAG uses as a parameter the angle between the new instance and one of the original instances being combined. In our experiments, we fixed the angle as half the angle between the original instances, but we consider that the flexibility of varying that parameter must be exploited.

6.7 Limitations and Ethical Concerns

Our experiments relied on a dataset that only contains English texts in the training and development sets. Only the test set is multilingual. Therefore, we were forced to partition the test set in order to perform the few-shot

cross-lingual transfer and domain adaptation experiments. We compared the results obtained in zero-shot cross-lingual transfer with the original test set and with the subset used in our experiments. We did not observe statistical differences. However, this may be a limitation in comparing our results with the original results reported in the dataset paper. Moreover, we observed a limitation in the strategy of domain adaptation. As we discussed in the error analysis, although the class of abusive texts is favored with this strategy, we observed a detriment in the negative class.

This work aims to improve abusive language detection in low-resource languages. While this can be useful for many languages, there are certain ethical implications. Therefore, we strongly recommend not using the proposed strategies as the sole basis for decision-making in abusive language detection. Regarding the issue of privacy, all the data we use in our experiments, both the original dataset and the new texts in Spanish that we generated, are publicly available. It should be noted that the scope of this work is strictly limited to the evaluation of models that are also publicly available, and it is not used to promote abusive language with the information obtained.

Acknowledgements

FairTransNLP research project (PID2021-124361OB-C31) funded by MCIN/AEI/10.13039/501100011033 and by ERDF, EU A way of making Europe. Part of the work presented in this article was performed during the first author's research visit to the University of Mannheim, supported through a Contact Fellowship awarded by the DAAD scholarship program "STIBET Doktoranden".

Appendices

Reproducibility

Table 6.6 provides features and links to the pre-trained models that we use, and Table 6.7 illustrates details of the dataset.

Model:	mBERT
Vocab size:	120k
#Params:	177M
Link:	https://huggingface.co/bert-base-multilingual-cased
Use in this work:	SSMBA
	Experiments (zero-shot and few-shot cross-lingual transfer)
Model:	mGPT
Vocab size:	100k
#Params:	1417M
Link:	https://huggingface.co/ai-forever/mGPT
Use in this work:	MIXUP & MIXAG
Model:	XLM-R
Vocab size:	250k
#Params:	270M
Link:	https://huggingface.co/xlm-roberta-base
Use in this work:	Ablation studies
Model:	mT0
Vocab size:	250k
#Params:	550M
Link:	https://huggingface.co/bigscience/mt0-base
Use in this work:	Ablation studies

Table 6.6: Features of the models used in this work. We built our models directly on top of the HuggingFace Transformers library.

Domain	Train (EN)	Validation (EN)	Test (LANG)
GAO	919	218	99
TRAC	10,341	2,593	300
WUL	71,754	24,130	600
ALL	83,014	26,941	999

Table 6.7: Number of texts in each set of the XHate-999 dataset. LANG stands for each language in {EN, SQ, HR, DE, RU, TR, ES}.

MIXAG Details

MIXAG is a particular case of MIXUP where the parameter λ of the linear combination (Equation 6.9) is determined by the angle α between the original vectors x_i and x_j, as well as the angle θ between the new vector $\hat{x}$ and one of

the original vectors. We take x_i without loss of generality (see Figure 6.3). We rely on the cosine of α, calculated as Equation 6.10, where $||\cdot||$ denotes the norm of a vector. Notice that we only parameterize one of the original vectors, since α and θ are sufficient to determine $\hat{x}$.

$$\hat{x} = \lambda x_i + x_j \tag{6.9}$$

$$cos(\alpha) = \frac{x_i x_j}{||x_i||||x_j||} \tag{6.10}$$

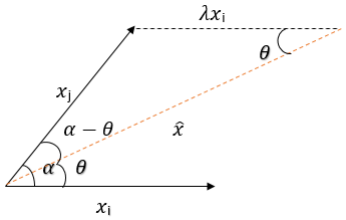


Figure 6.3: MIXAG explanation.

The objective is to express the parameter λ as a function of θ, hence we take advantage of the Law of Sines (Equation 6.11) that allows relating vectors and angles. Then, λ can be expressed in function of θ as Equation 6.12. Finally, using the known identities in Equations 6.13, we can define λ from the cosine of α, which can be obtained with Equation 6.10, and the cosine of θ, which is the parameter of MIXAG (Equation 6.14).

$$\frac{\lambda||x_i||}{sin(\alpha-\theta)} = \frac{||x_j||}{sin(\theta)} \tag{6.11}$$

$$\lambda = \frac{||x_j||sin(\alpha-\theta)}{||x_i||sin(\theta)} \tag{6.12}$$

$$\begin{aligned} &sin(\alpha-\theta)=sin(\alpha)cos(\theta)-cos(\alpha)sin(\theta) \\ &sin(\theta)=\sqrt{1-cos(\theta)^2}, \quad sin(\alpha)=\sqrt{1-cos(\alpha)^2} \\ &sin(\alpha-\theta)=\sqrt{1-cos(\alpha)^2}cos(\theta)-cos(\alpha)\sqrt{1-cos(\theta)^2} \end{aligned} \tag{6.13}$$

$$\lambda=\frac{||x_j||(cos(\theta)\sqrt{1-cos(\alpha)^2}-cos(\alpha)\sqrt{1-cos(\theta)^2})}{||x_i||\sqrt{1-cos(\theta)^2}} \tag{6.14}$$

Results by Language and Domain

We show complete results in this section. Table 6.8 illustrates that there is no significant difference between using the full test set and using a subset of texts from the test set (the subset that we used in our experiment).

Table 6.9 illustrates the cosine similarity between each language and English for five linguistic features. We obtain these features as language vectors from LANG2VEC (Littell et al., 2017).

GAO	EN	DE	RU	TR	HR	SQ	ES
FZS	0.6742	0.5185	0.5063	0.5217	0.6098	0.5432	0.5060
ZS	0.6747	0.5067	0.5205	0.5116	0.6234	0.5405	0.5263
TRAC							
FZS	0.7594	0.7527	0.6859	0.6806	0.6925	0.7177	0.7045
ZS	0.7642	0.7582	0.6815	0.6777	0.6892	0.7235	0.7000
WUL							
FZS	0.8812	0.6739	0.5581	0.2969	0.5476	0.5675	0.5049
ZS	0.8800	0.6698	0.5561	0.2945	0.5469	0.5556	0.4960
ALL							
FZS	0.8053	0.7146	0.6322	0.3565	0.6231	0.6088	0.6028
ZS	0.8085	0.7156	0.6308	0.3627	0.6214	0.6127	0.6008

Table 6.8: Cross-lingual transfer performance with mBERT on each domain (GAO, TRAC, WUL) and the union of all domains (ALL). FZS refers to zero-shot cross-lingual transfer with the full test set, which corresponds to the results reported in XHATE-999: Analyzing and Detecting Abusive Language Across Domains and Languages. ZS refers to zero-shot cross-lingual transfer with 90% of the test set of each language, which corresponds to the results discussed in this paper. Results are reported in terms of F1.

Table 6.10 shows the correlation coefficient and p-value for these linguistic features.

- **SYN:** vectors encode syntactic properties, e.g., if a subject appears before or after a verb.
- **FAM:** vectors encode memberships in language families.
- **INV:** vectors denote the presence or absence of natural classes of sounds.
- **PHO:** vectors encode phonological properties such as the consonant-vowel ratio.
- **GEO:** vectors express orthodromic distances for languages w.r.t. fixed points on the Earth's surface.

Table 6.11 shows the Precision and Recall results across domains and languages for the error analysis of the unsupervised language adaptation.

	EN	DE	RU	TR	HR	SQ	ES
SYN	1.0	0.9025	0.8118	0.5067	0.8318	0.7959	0.8216
FAM	1.0	0.5443	0.1667	0.0	0.1260	0.3333	0.0962
INV	1.0	0.7628	0.6475	0.6658	0.6967	0.7249	0.6382
PHO	1.0	0.8058	0.8581	0.8181	0.8581	0.8704	0.8581
GEO	1.0	0.9976	0.9681	0.9825	0.9950	0.9919	0.9959

Table 6.9: Cosine similarity between each language vector and English vector for LANG2VEC-based language vectors (SYN, FAM, INV) considering all domains.

	SYN		FAM		INV		PHO		GEO	
	Pearson	P-value	Pearson	P-value	Pearson	P-value	Pearson	P-value	Pearson	P-value
ZS	0.672	<.001	0.647	<.001	0.599	<.001	0.529	.003	0.302	.119
FS	0.664	<.001	0.661	<.001	0.607	<.001	0.516	.004	0.289	.136
SS	0.527	.004	0.627	<.001	0.608	<.001	0.486	.008	0.261	.180
MU	0.405	.033	0.628	<.001	0.633	<.001	0.463	.013	0.315	.210
MA	0.571	.001	0.721	<.001	0.686	<.001	0.529	.004	0.245	.209

Table 6.10: Complete table of correlations between zero-shot (ZS), few-shot (FS), few-shot with SSMBA (SS), few-shot with MIXUP (MU) and few-shot with MIXAG (MA) cross-lingual transfer performance with mBERT across all languages and domains, with linguistic proximity scores (features SYN, FAM, INV, PHO, GEO). Correlations that are not statistically significant are underlined (α=.05).

GAO	EN		DE		RU		TR		HR		SQ		ES	
	R	P	R	P	R	P	R	P	R	P	R	P	R	P
ZS	0.70	0.65	0.48	0.54	0.48	0.58	0.55	0.48	0.60	0.65	0.50	0.59	0.50	0.55
ZS_MLM	0.80	0.47	0.88	0.50	0.78	0.50	0.98	0.47	0.98	0.46	0.93	0.45	0.98	0.44
TRAC														
ZS	0.89	0.67	0.88	0.66	0.74	0.63	0.71	0.65	0.78	0.62	0.85	0.63	0.73	0.67
ZS_MLM	0.92	0.54	0.81	0.54	0.92	0.53	0.93	0.53	0.93	0.53	0.98	0.54	0.72	0.53
WUL														
ZS	0.80	0.98	0.51	0.97	0.39	0.94	0.17	0.96	0.38	0.97	0.40	0.92	0.33	0.97
ZS_MLM	0.83	0.48	0.89	0.49	0.99	0.51	0.98	0.51	0.99	0.51	0.97	0.51	0.99	0.51
ALL														
ZS	0.79	0.82	0.71	0.72	0.57	0.71	0.24	0.75	0.56	0.69	0.54	0.71	0.51	0.74
ZS_MLM	0.99	0.50	0.99	0.51	0.98	0.50	0.96	0.50	0.99	0.51	0.99	0.51	0.89	0.50

Table 6.11: Precision (P) and Recall (R) in cross-lingual transfer with mBERT on each domain (GAO, TRAC, WUL) and the union of all domains (ALL). ZS_MLM refers to adapted zero-shot cross-lingual transfer, while ZS refers to zero-shot cross-lingual transfer.

Part IV

Summary

Chapter 7

Discussion of the Results

In this chapter, we discuss in detail the results obtained in this thesis. We first analyze the results of the publications presented in Parts I, II, and III concerning the objectives of our research. We also include some further results to complete the picture of abusive language detection in low-resource settings. This should allow us to better answer the three research questions. First, in Section 7.1, we analyze the characteristics of the keywords obtained with the two abusive keyword extraction methods that we presented in Part I. Then, we present new experiments to compare these sets of keywords. Moreover, we investigate the bias of Transformer-based models with respect to the extracted keywords and evaluate how the performance of these models can be affected by altering the bias. Our analysis focuses on cases of low-resource settings.

In Section 7.2, we summarize our findings on the use of strategies based on graph neural networks (GNN) proposed in Part II. Then, we present additional experiments to analyze the performance of this type of strategy in low-resource settings cases. In this study, we compare: i) the use of GNN-based models and Transformer-based models, both ii) in low-resource settings and in settings where there is no insufficiency of data. In addition, we conducted an ablation analysis to investigate how GNN-based strategies can be modified to further improve abuse language detection results.

In Section 7.3, we extend the experiments conducted in Part III. We focus on evaluating how the performance of other Transformer-based models varies with data augmentation techniques in low-resource settings. Moreover, we compare this variation in settings where data is not scarce.

Finally, in Section 7.4 we introduce an approach that goes beyond natural language processing for hate speech detection in social networks. This research focuses on analyzing hate from the perspective of Twitter users. The studies were conducted in parallel to the research presented in the previous chapters. The main objective is to present preliminary findings on the characteristics of users who tend to post hate messages. This can help to obtain

a more comprehensive view of the phenomenon of abusive language on social networks. Firstly, we provide an overview of the Author Profiling shared task that we organized as part of PAN 2021.[1] This task was about determining whether or not the author of a Twitter feed is likely to spread hatred. In this section, we present the main strategies and the results obtained. Then, we present further experiments that we conduct to analyze networks of users prone to publish hate messages. We study these networks as graphs in which the users represent the nodes and the connections between them represent the edges. In this way, we characterize the relationships between the users and provide a framework for future research.

7.1 Keyword Extraction and Bias Analysis

In Part I, we present two methods for abusive keyword extraction. Table 7.1 lists the main aspects of both methods.

	M_{BERT} (Chapter 2)	M_{HMRF} (Chapter 3)
Based on	Attention mechanism of BERT to assign a relevance value to each pair of words in a text. This value is used as the weight of the edges of a word graph from which the keywords are extracted using the eigenvector centrality.	Harmonic mean of the relative frequencies: i) frequency (F) of each word w in the set of abusive texts and ii) F with respect to the frequency of w in the entire collection of texts.
Paper focus	Offensive language	Hate speech
Input	Dataset of abusive (offensive/hateful) and non-abusive texts.	
Main contribution	Exploitation of the multi-head self-attention mechanism.	Simple strategy. *Note*: Results of Chapter 3 show little overlap between the keywords extracted with this method and the salient words of BERT (words to which BERT pays more attention to detecting hatred).
Limitation	Extraction of words that are not abusive as keywords. It is due to the frequent presence of these words in abusive texts and their absence in the rest of the texts.	

Table 7.1: Overview of keyword extraction methods.

A point in favor of both keyword extraction methods is their ability to characterize the abusive language present in the datasets. In Part I, we observed that the extracted keywords are closely related to the target against which the abuse is directed. However, our methods can discard words that may indicate abuse but are frequently used in non-abusive texts. In Chapter 3, we comment on the example of a thread of posts about feminism. In this case, the word 'feminist' is likely to appear frequently not only in the abusive texts but also in the rest of the texts. Therefore, our methods do not select that word as a keyword and look for other more discriminating words that indicate abuse in that particular context. Besides, in Chapter 3 we analyzed the bias of transformer-based models toward the keywords

[1] https://pan.webis.de/clef21/pan21-web/

extracted by M_{HMRF} with two different metrics and evaluated two strategies to mitigate the bias. The results suggested that the bias toward hateful keywords can be reduced when fine-tuning the models with hateful texts without the keywords. Likewise, we observed that this bias reduction may imply an improvement in the performance of abusive language detection.

Following, we present some additional experiments we carried out for further investigating these aspects in low-resource settings.

7.1.1 Experimental Setup

Dataset. Unlike the experiments in Part I, in this section we use the XHate-999 dataset (Glavaš et al., 2020). This is a dataset intended to explore abusive language detection and that we used in Chapters 4 and 6. This dataset includes three different domains: Fox News (GAO), Twitter/Facebook (TRAC), and Wikipedia (WUL). We also define ALL as the set of instances resulting from the union of all three domains. Each domain comprises different amounts of annotated data (abusive/non-abusive) in English for training, validation, and testing. GAO is the domain with the least amount of data and WUL is the one with the greatest amount. The test set is small: 99 instances in GAO, 300 instances in TRAC, and 600 in WUL. The English test instances are translated into five target languages: Albanian (SQ), Croatian (HR), German (DE), Russian (RU) and Turkish (TR), and we also include Spanish (ES). In this section, we focus on the set of texts in English. In order to simulate a low-resource setting, we ignore the training and validation sets. We only use the test set in our experiments: 90% of the instances from the test set is used for evaluation and the remaining 10% of the instances is used for fine-tuning the models, i.e. we use 10 instances for fine-tuning in GAO (and 89 for evaluation), while the corresponding numbers are 30 (270) for TRAC, 60 (540) for WUL and 100 (899) for ALL (GAO+TRAC+WUL).

External resource. As we suggested in Chapter 3 we use HurtLex (Bassignana et al., 2018), a lexicon that contains hateful words that are independent of a specific text collection. This allows us to compare the keywords extracted with our methods against hateful words taken from an external resource.

Models. Table 7.2 summarizes the transformer-based models that we use for the analysis of bias toward the keywords, and how bias can affect the performance of abusive language detection.

7.1.2 Analysis of Abusive Keywords

A characteristic that we observe in relation to M_{HMRF} is that the set of extracted keywords differs from the salient words of BERT. It is therefore to be

Models	Architecture	Link
BERT	bert-base-uncased	https://huggingface.co/bert-base-uncased
ROBERTA	roberta-base	https://huggingface.co/roberta-base

Table 7.2: Links to the pre-trained models used in bias analysis.

expected that this set of keywords differs from the set of keywords extracted using our M_{BERT} method (based on the BERT attention mechanism). In our experiments, we use the test set of each domain of XHate-999 as the input of M_{HMRF} and M_{BERT}. Table 7.3 illustrates the overlap between the obtained sets in terms of the keywords extracted by the methods and the percentage of overlap (the percentages were calculated taking into account that 50 keywords were extracted by each method). In addition, the overlap between each of these keyword sets and the words from HurtLex is shown.

	M_{HMRF}-M_{BERT}		M_{HMRF}-HurtLex		M_{BERT}-HurtLex	
	keywords	%	keywords	%	keywords	%
GAO	*good, black, work, person*	8	*jews, decent, terrorist, black, abuses*	10	*problem, minister, stupid, people, poor, coward, army, black*	16
TRAC	*country, citizen, thought, stupid, anyone, army, black, singh, person*	18	*stupid, mouth, dumb, god, kill, army, black, cow*	16	*problem, minister, stupid, people, poor, coward, army, black*	16
WUL	*time, want, stupid, real, people, stop, poor*	14	*moron, cock, idiot, stupid, dumb, bitch, shit, gay, die, wtf, asshole, cunt, poor, bastard, people, dick, fuck, penis, ass*	38	*problem, minister, stupid, people, poor, coward, army, black*	16
ALL	*country, time, want, stupid, stop, poor, army, black, person*	18	*moron, idiot, stupid, dumb, army, bitch, shit, gay, die, asshole, kill, cunt, poor, bastard, idiotic, dick, fuck, black, ass*	38	*problem, minister, stupid, people, poor, coward, army, black*	16

Table 7.3: Overlap between the abusive keywords extracted with M_{BERT}, M_{HMRF}, and terms from HurtLex.

Indeed, we note that the percentage of overlap between M_{HMRF} and M_{BERT} is low, generally less than 15%. However, we observe that in most cases the keywords extracted by both methods represent abusive words that characterize the content of the dataset. We also found an overlap of almost 50% with the words from HurtLex in the keywords extracted from WUL and ALL using M_{HMRF}. This can be an indication that with the simple statistics used in M_{HMRF}, it is possible to encounter abusive words in the datasets even if they do not match the salient words of BERT.

7.1.3 Bias and Performance Analysis

In Chapter 3, we evaluate two strategies to mitigate the bias of a model toward a set of keywords. These strategies are based on fine-tuning a model to make a small fit in its parameters with a very small learning rate ($1x10^{-7}$). The particularity of each strategy lies in the data used for the fit:

- V_1: Data only contains abusive texts without keywords.
- V_2: Data only contains non-abusive texts with keywords.

Since we are interested in investigating the bias of the models toward the keywords extracted with M_{HMRF} and M_{BERT}, in the following experiments we analyze each of the four V_{im} models (i = $\{1, 2\}$ and m = $\{M_{HMRF}$, $M_{BERT}\}$), where i denotes the bias mitigation strategy and m denotes the set of keywords considered.

Evaluation details. In each domain of the dataset, we use only 10% of the test set to fine-tune a transformer-based model and obtain an initial model. To obtain the models V_{im}, we then select from the validation set of the corresponding domain, the texts that satisfy V_i. We train with batches of size 2 in 3 epochs and evaluate with batches of 2 instances. The performance of the models is reported in terms of F1-score and we rely on McNemar's statistical test (Dietterich, 1998) to analyze if each variant of the initial model varies significantly with a significance level $\alpha = .05$, i.e. we compare each model V_{im} with the initial model. To estimate bias, we use the metric based on AUC-ROC described in Chapter 3, which balances overall performance with various aspects of bias.[2] *The lower the value of this metric, the higher the bias.*

7.1.3.1 Results and Discussion

Table 7.4 illustrates the estimated bias for each model in each domain. In these low-resource settings, we observe different results than in Chapter 3. In those previous experiments, an important finding was that the bias toward abusive keywords could be reduced when the models were fine-tuned with abusive texts in which the keywords did not appear (V_1).

		GAO		**TRAC**		**WUL**		**ALL**	
		M_{HMRF}	M_{BERT}	M_{HMRF}	M_{BERT}	M_{HMRF}	M_{BERT}	M_{HMRF}	M_{BERT}
BERT	Initial model	0.989	0.996	0.579	0.575	0.766	0.751	0.830	0.605
	$V_{1M_{HMRF}}$	0.595	-	0.549	-	0.958↓	-	0.785	-
	$V_{1M_{BERT}}$	-	0.588	-	0.548	-	0.869↓	-	0.575
	$V_{2M_{HMRF}}$	0.652	-	0.644↓	-	0.886↓	-	0.625	-
	$V_{2M_{BERT}}$	-	0.637	-	0.656↓	-	0.953↓	-	0.953↓
ROBERTA	Initial model	0.556	0.577	0.559	0.550	0.975	0.893	0.785	0.600
	$V_{1M_{HMRF}}$	0.581↓	-	0.541	-	0.935	-	0.937↓	-
	$V_{1M_{BERT}}$	-	0.577	-	0.609↓	-	0.814	-	0.588
	$V_{2M_{HMRF}}$	0.730↓	-	0.456	-	0.992↓	-	0.615	-
	$V_{2M_{BERT}}$	-	0.542	-	0.559↓	-	0.582	-	0.665↓

Table 7.4: Estimated bias for the initial model (no bias mitigation) and the models fitted for bias mitigation ($V_{1M_{HMRF}}$, $V_{1M_{BERT}}$, $V_{2M_{HMRF}}$, $V_{2M_{BERT}}$). The symbol ↓ identifies the cases in which the bias is mitigated.

In these new experiments, however, the analyzed strategies only succeed in mitigating the bias in a few cases. In the smallest domain (GAO), the

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

bias even increases in most cases. This suggests that to minimize the bias with V_1, a larger amount of data is probably required to correctly adjust the models to detect abusive language in texts without keywords. Note that we used very few instances: we start from a very small set from which only the instances that satisfy V_1 in each mitigation strategy are selected.

Table 7.5 shows the results of abusive language detection for each domain with the two transformer-based models. The V_{im} techniques seem to improve the performance of the model in a few cases, mainly in the smaller domain. However, this improvement does not seem to be associated with the mitigation of bias. We only observed correspondences between the improvement in model performance and the reduction in bias in four cases.

		GAO	**TRAC**	**WUL**	**ALL**
BERT	Initial model	0.4871	0.7961	0.8712	0.7403
	$V_{1M_{HMRF}}$	0.5348↑	0.7500	<u>0.8821</u>↑	0.7735↑
	$V_{1M_{BERT}}$	0.5444↑	0.7928	<u>0.8832</u>↑	0.7877↑
	$V_{2M_{HMRF}}$	0.5348↑	0.7129	0.8176	0.7101
	$V_{2M_{BERT}}$	0.5000↑	0.6809	0.8428	0.6769
ROBERTA	Initial model	0.5455	0.7525	0.9171	0.7677
	$V_{1M_{HMRF}}$	<u>0.7200</u>↑	0.7179	0.9016	<u>0.7967</u>↑
	$V_{1M_{BERT}}$	0.5455	0.7143	0.9005	0.8144↑
	$V_{2M_{HMRF}}$	0.4762	0.7500	0.8675	0.7445
	$V_{2M_{BERT}}$	0.5217	0.6809	0.8994	0.7203

Table 7.5: Performance for the initial model (no bias mitigation) and the models fitted for bias mitigation ($V_{1M_{HMRF}}$, $V_{1M_{BERT}}$, $V_{2M_{HMRF}}$, $V_{2M_{BERT}}$). The symbol ↑ identifies the cases of significant improvement ($\alpha = .05$). Underlined values indicate an increase in F1 related to a decrease in bias.

As we noted in Chapter 3, the selection of data size for the bias mitigation strategy appears to be an important factor in achieving effective results. For the low-resource settings that we examine in this section, we did not find the expected pattern of bias reduction and consequently did not observe a correlation between the mitigation of the bias and improvement in the performance of the models. Therefore, for future work, we suggest a more in-depth analysis to determine the optimal amount of data for mitigation strategies. Moreover, we consider it interesting to extend the analysis related to abusive keywords to other languages in future research, as we outlined in Chapter 2. Our experiments focus on English but note that both strategies can also be used for other languages. In the case of M_{BERT}, a BERT model trained for the respective language would have to be used.

7.2 Graph-Based Exploration

Graph neural networks have become a great alternative in various applications. Their ability to capture intrinsic patterns and dependencies, deal

with noisy data, and draw inductive reasoning makes them a powerful tool for analyzing graph-structured data. To exploit these advantages, in this dissertation, we evaluate their ability in abusive language detection.

In Part II, we presented our findings. First, we proposed a graph auto-encoder (GAE) framework for generating text embeddings. The primary idea is to start from a basic initial representation and use a GAE to transform these representations into a low-dimensional space in which texts of the same class are closer to each other. From this space, we extract a new representation (embeddings) that should allow a better distinction between the classes used to learn the representation space. In Chapter 4, we illustrate how embeddings achieve a better separation between abusive texts and the rest of the texts, than an original representation based on TFIDF. We verify these results with a fully connected neural network (FCNN). As input for the classifier, we use the original representation on the one hand and the embeddings obtained with our framework on the other hand. The classification results show the superiority of the embeddings. We explain this framework and the results obtained in Chapter 4. We also observed interesting results in comparison to the MBERT and XLM-R models. We found that classification based on the embeddings we generated performs significantly better on small datasets.

These results motivated us to evaluate the suitability of models based on graph neural networks in abusive language detection. Thus, we investigate the particular case of convolutional graph neural network (CGNN) for hate speech detection in Chapter 5. Again, we found that the GNN-based model outperforms the transformer-based models.

In this section, we extend the experiments to get an overview of the performance of GNN-based models in low-resource settings.

7.2.1 Experimental Setup

Dataset. In this section, we use the same dataset as in the previous section (see Section 7.1.1). In this section, we evaluate the seven languages for analysis in low-resource settings, using 10% of the test set for fine-tuning the models and the rest for testing. We also analyzed an environment where there is no data limitation (high-resource settings) for English, for which we added the XHate-999 training set to the partition used for fine-tuning the models.

Models. Table 7.6 summarizes the multilingual models that we use in this section. These models are used to compare the performance of GNN-based models with transformer-based models in low-resource settings.

Evaluation details. In this section, we use the architecture of the multilayer convolutional graph neural network studied in Chapter 5. We con-

Models	Architecture	Link
BERT-M	bert-base-multilingual-cased	https://huggingface.co/bert-base-multilingual-cased
MBERT	M-CLIP/M-BERT-Distil-40	https://huggingface.co/M-CLIP/M-BERT-Distil-40
XLM-R	xlm-roberta-base	https://huggingface.co/xlm-roberta-base

Table 7.6: Links to the pre-trained models used in graph-based exploration.

ducted empirical experiments to find out that two layers are sufficient to achieve optimal results with the model. In all evaluation domains, we observed a significant jump in results when we used two layers instead of a single layer. However, we observed that the results remained similar with a larger number of layers. We set the size of the convolutional layers to 32 and trained the model with batches of size 32 in 200 epochs. The performance of the models is reported in terms of F1-score and we rely on McNemar's test for analyzing statistical significance with a significance level $\alpha = .05$. The GNN-based model takes as input a matrix X in which each row corresponds to the representation of a text and a matrix A with a representative of the graph structure (adjacency matrix). We obtained the representation of the texts (vectors) based on TFIDF to generate X, and the inner product of these vectors to generate A.

7.2.2 Results and Discussion

		Resource Settings							
		High	**Low**						
		EN	EN	DE	RU	TR	HR	SQ	ES
GAO	BERT-M	0.6107	0.4171	0.6703	0.5694	0.4712	0.4276	0.4444	0.4222
	MBERT	0.6735	0.4222	0.5249	0.6703	0.4276	0.6411	0.5333	0.5764
	XLM-R	0.6314	0.5238	0.5828	0.6411	0.4667	0.4712	0.6122	0.3750
	GNN_{CGNN}	**0.6824**	**0.8696**	**0.9600**	**0.7692**	**0.8000**	**0.9231**	**0.8000**	**0.9600**
TRAC	BERT-M	0.6450	0.5601	0.5296	0.5249	0.5276	0.5536	0.5553	0.5072
	MBERT	0.6736	0.5640	0.5375	0.5445	0.4825	0.5101	0.5193	0.5296
	XLM-R	0.6477	0.6627	0.6936	0.6559	0.5801	0.6558	0.5725	0.5558
	GNN_{CGNN}	**0.6915**	**0.8302**	**0.7500**	**0.7344**	**0.7931**	**0.8654**	**0.7234**	**0.7234**
WUL	BERT-M	0.8200	0.7697	0.7221	0.7466	0.7233	0.7599	0.7221	0.6790
	MBERT	0.8256	0.8374	0.8103	0.8103	0.7882	0.7887	0.7556	0.7278
	XLM-R	0.8792	0.8768	**0.8547**	0.8942	0.8262	0.8599	0.8055	0.7991
	GNN_{CGNN}	**0.8823**	**0.9000**	0.7888	**0.9072**	**0.8901**	**0.8879**	**0.9898**	**0.8876**
ALL	BERT-M	0.7567	0.6664	0.7095	0.6360	0.6795	0.6261	0.6432	0.6167
	MBERT	0.7717	0.7267	0.7108	0.6998	0.7044	0.6796	0.6832	0.6811
	XLM-R	**0.7988**	**0.7547**	**0.7792**	**0.7920**	**0.7226**	**0.7600**	**0.7627**	0.6654
	GNN_{CGNN}	0.7801	0.7013	0.7178	0.7013	0.7013	0.7467	0.7013	**0.7013**

Table 7.7: Abusive language detection with the convolutional graph neural network and transformer-based models. Results are reported in terms of F1 and numbers in bold indicate the best significant ($\alpha = .05$) results for each language and domain.

Table 7.7 shows the results for abusive language detection in each domain and language. We find that the GNN-based model outperforms the

remaining models in most cases for the GAO, TRAC, and WUL domains. However, the XLM-R model seems to perform better in the domain with the largest amount of data (ALL). Also note that the improvement of the GNN-based model becomes smaller as the amount of data increases, i.e., the difference between the results of CGNN and the transformer-based models is larger for GAO and smaller for WUL.

Similarly, we observe that for English in high-resource settings, the difference between the results obtained with CGNN and the transformer-based models is smaller than for English in low-resource settings. These results confirm that GNN-based models may be the best choice when data is scarce.

7.2.2.1 Ablation Analysis

In the previous experiments, we used a graph convolutional operator (Kipf and Welling, 2017b) to create the CGNN model. This operator aggregates information from neighboring nodes, i.e., in each layer of the model, each node of the graph aggregates information from the neighbors of the node and updates its representation. Table 7.8 shows the results of an ablation analysis in which we tested variations of the GNN-based model concerning different types of layers (operators) used in the model design. The types of layers differ in their approaches to aggregating information from neighboring nodes. We evaluate two additional operators:

- Graph transformer operator (Shi et al., 2021b) (**TCGNN**):

  It uses self-attention mechanisms to weigh the importance of different nodes in the graph when updating a particular node's representation. The self-attention mechanism allows nodes to attend to other nodes with varying degrees of importance, capturing complex relationships in the graph.

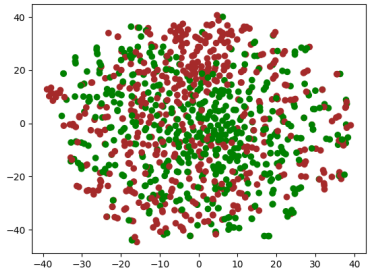
- GraphSAGE (Graph Sample and Aggregation) operator (Hamilton et al., 2017) (**SAGE**):

  It employs a sampling and aggregation strategy. Instead of considering all neighbors of a node, it samples a fixed-size neighborhood and aggregates information from these sampled neighbors. In our case, the sampled node representations are aggregated using the mean function.

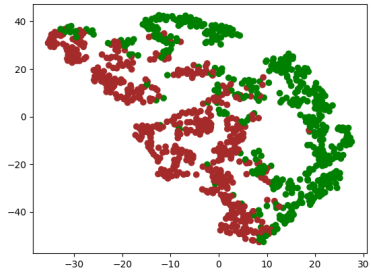
Figure 7.1 illustrates the representation space of the texts for the English example in a low-resource setting. We note that in the initial representation (TFIDF) in Figure 7.1a it is difficult to distinguish between instances of different classes. On the other hand, Figures 7.1b, 7.1c and 7.1d show a large separation after transforming this initial representation with the operators of the GNN-based models. Figures 7.2 to 7.7 show similar behavior for the rest of the languages.

		Resource Settings							
		High	**Low**						
		EN	EN	DE	RU	TR	HR	SQ	ES
GAO	CGNN	0.6824	**0.8696**	**0.9600**	**0.7692**	**0.8000**	**0.9231**	0.8000	**0.9600**
	TCGNN	**0.8315**	0.8462	0.9565	0.6923	0.7333	0.8571	**0.9231**	0.9091
	SAGE	0.6824	0.8148	0.9091	0.6677	0.6897	0.8462	0.8148	0.9167
TRAC	CGNN	0.6915	0.8302	0.7500	0.7344	0.7931	**0.8654**	0.7234	0.7234
	TCGNN	0.9620	**0.9231**	0.8713	0.7500	**0.8350**	0.7963	**0.9804**	**0.9307**
	SAGE	**0.9874**	0.8687	**0.8868**	**0.7579**	0.8000	0.8033	0.9000	0.8713
WUL	CGNN	0.8823	0.9000	0.7888	**0.9072**	**0.8901**	0.8879	0.9898	0.8876
	TCGNN	**0.8906**	**0.9744**	**0.9899**	0.8242	0.8287	**0.9263**	**0.9899**	**0.9691**
	SAGE	0.8894	0.9000	0.9314	0.8400	0.8343	0.8774	0.8796	0.8770
ALL	CGNN	0.7801	0.7013	0.7178	0.7013	0.7013	0.7467	0.7013	0.7013
	TCGNN	**0.7890**	**0.9565**	**0.9408**	**0.8389**	**0.8202**	**0.8882**	**0.9908**	**0.9876**
	SAGE	0.7856	0.8712	0.8780	0.8182	0.7419	0.8800	0.8758	0.9527

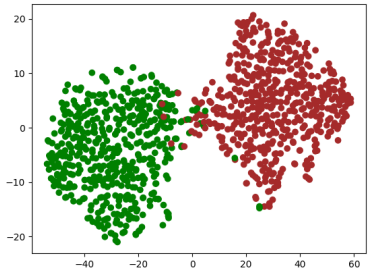
Table 7.8: Ablation analysis for the convolutional graph neural network. Results are reported in terms of F1 and numbers in bold indicate the best significant ($\alpha = .05$) results for each language and domain.



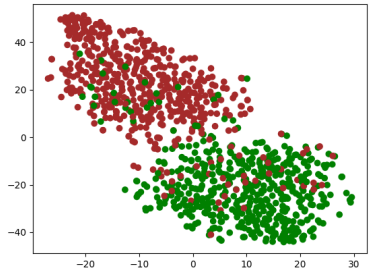
(a) TFIDF



(b) CGNN



(c) TCGNN

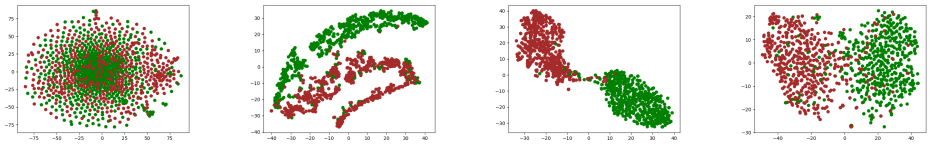


(d) SAGE

Figure 7.1: Representation of English texts in the low-resource setting with t-SNE (Pezzotti et al., 2017a).

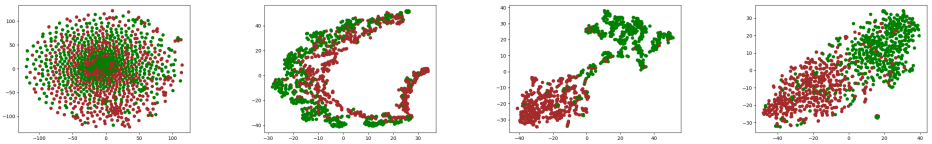
In many cases, the type of layer used has a significant impact on the results. Note, that in the ALL domain, TCGNN is better than CGNN in all languages. It even improves the results of XLM-R, which was superior to the

GNN-based model in previous experiments. These are important results that complement the findings from the previous experiments. Not only do GNN-based models appear to be a superior alternative for low-resource settings, but different architectures can improve abusive language detection depending on the used data.



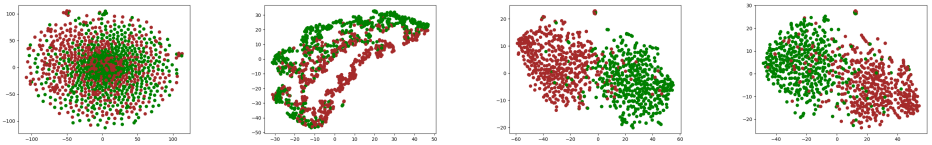
(a) TFIDF (b) CGNN (c) TCGNN (d) SAGE

Figure 7.2: Representation of German texts in the low-resource setting



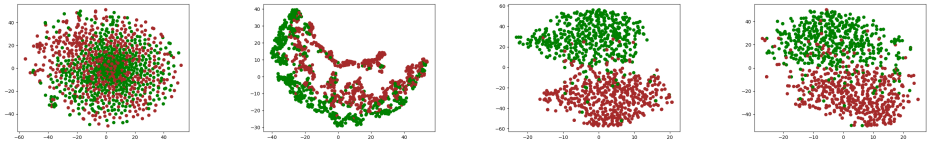
(a) TFIDF (b) CGNN (c) TCGNN (d) SAGE

Figure 7.3: Representation of Russian texts in the low-resource setting



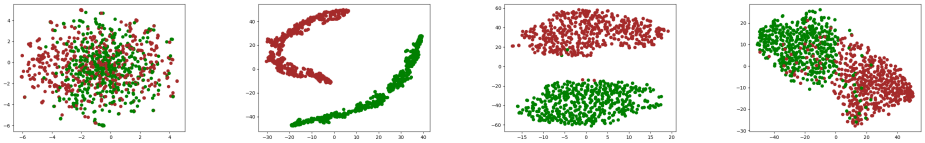
(a) TFIDF (b) CGNN (c) TCGNN (d) SAGE

Figure 7.4: Representation of Turkish texts in the low-resource setting



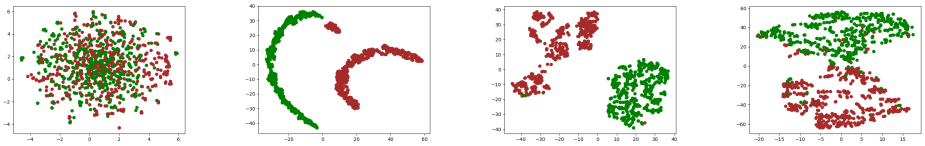
(a) TFIDF (b) CGNN (c) TCGNN (d) SAGE

Figure 7.5: Representation of Croatian texts in the low-resource setting



(a) TFIDF (b) CGNN (c) TCGNN (d) SAGE

Figure 7.6: Representation of Albanian texts in the low-resource setting



(a) TFIDF (b) CGNN (c) TCGNN (d) SAGE

Figure 7.7: Representation of Spanish texts in the low-resource setting.

7.3 Data Augmentation

In Part III we found that data augmentation techniques tend to improve the results of few-shot learning and that this improvement is more pronounced with a smaller amount of data to fine-tune the models. This suggests that data augmentation may be particularly beneficial in low-resource settings. In this section, we analyze this aspect in more detail. We evaluate how the amount of newly generated data affects the performance of abusive language detection when there is only a small collection of training data.

7.3.1 Experimental Setup

Dataset. In this section, we use the same dataset as in the previous section (see Section 7.1.1). In the experiments of this section, we focus on the set of English texts. For the low-resource setting, we use the test set (999 instances considering the three domains), of which we use 660 for training and 339 for testing. For the high-resource setting, we use the XHate-999 training set. From this large set of texts, we employ 6600 instances for training, while for testing, we use the same 339 instances as for the low-resource setting.

Models. Table 7.9 shows the models we use in our experiments. These are the models employed in the original paper of the XHate-999 dataset.

Models	Architecture	Link
MBERT	M-CLIP/M-BERT-Distil-40	https://huggingface.co/M-CLIP/M-BERT-Distil-40
XLM-R	xlm-roberta-base	https://huggingface.co/xlm-roberta-base

Table 7.9: Links to the pre-trained models used in data augmentation.

7.3.2 Data Augmentation

In Part III, we investigate strategies based on the Vicinal Risk Minimization principle and observe how the performance of MBERT improves especially when the amount of training data is small. This may be because overfitting is more likely in low-resource settings, so using new data to solve this problem can improve classification performance. In the following experiments, we study the Easy Data Augmentation (EDA) strategy Wei and Zou (2019). This strategy consists of four simple operations: i) synonym replacement, ii) random insertion, iii) random swapping, and iv) random deletion. Previous experimental results in the literature have shown that EDA can improve classification performance and in particular provide strong results for smaller datasets. To extend the analysis, we also use the SSMBA strategy (Ng et al., 2020), which was studied in Chapter 6. This strategy uses a pair of corruption and reconstruction functions to move randomly through a large amount of data.

Our experiments are based on examining the performance gain of the models when they are fine-tuned with an augmented text set compared to the performance when the original text set is used. To this end, we use the data augmentation strategies (EDA and SSMBA) to generate a certain number n of new instances for each text of the original text set. Our study focuses on analyzing how the performance of the models varies as a function of the value of n.

7.3.3 Results and Discussion

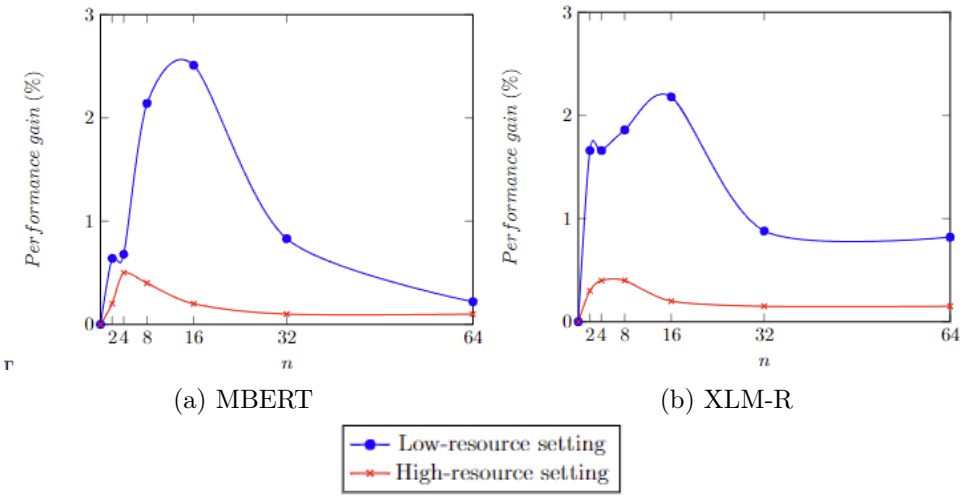
Table 7.10 shows the results in the low-resource setting for the MBERT and XLM-R models, respectively. In general, the use of a large amount of new data led to an increase in performance. Based on the results, generating 16 instances for each text in the training set seems to be an optimal amount. The best performance gains are achieved with this value of n and drop when a larger number of instances are added. Note that we start from a training set with 660 instances. For other datasets with a larger amount of data, it may be necessary to add fewer new instances per original text, as the models tend to generalize correctly with large amounts of data.

Figures 7.8a and 7.8b show the average of the performance gains with EDA and SSMBA in both low-resource setting and high-resource setting. We found that the gain is much larger when the initial text amount is small and that the largest gain is achieved in the high-resource setting with an n value

	MBERT		XLM-R	
n	EDA	SSMBA	EDA	SSMBA
2	1.26	0.02	1.86	1.46
4	0.68	0.67	2.01	1.30
8	2.00	2.27	2.20	1.51
16	**2.66**	**2.35**	**2.68**	**1.67**
32	1.33	0.32	0.36	1.40
64	0.23	0.20	0.62	1.02

Table 7.10: Performance gain (%) for data augmentation (EDA and SSMBA) in abusive language detection. The value n indicates the number of new instances generated for each training instance. Numbers in bold indicate the best performance gain.

of 4. This suggests that data augmentation is particularly beneficial when the amount of data is limited and that in cases where there is a sufficient amount of data, generating a large number of new instances may be unproductive.



(a) MBERT (b) XLM-R

Figure 7.8: Average performance gain in data augmentation (EDA and SSMBA) for various training set sizes. n is the number of generated augmented texts per original text.

7.4 Hate Speech Spreaders

In the previous sections, we discussed the three fundamental aspects that we focused on during the development of this doctoral thesis for abusive

language detection in low-resource settings. In this section, we introduce further experiments to study the presence of abusive content on social networks. In particular, we examine the characteristics of users who tend to spread this type of content on Twitter (Hate Speech Spreaders - HSS). We develop this research in two ways: i) analyzing whether a user is HSS based on the analysis of their publications and ii) analyzing the characteristics of the HSS network to study how they tend to connect and how a hateful message can be spread through this network. For the first point, we organized a shared task that allowed us to look at and compare different strategies and outcomes. For the second point, we conduct experiments based on graph analysis.

The rest of this section summarizes the experiments performed and our findings. This study aims to provide a more comprehensive view of the spread of hatred in a social network by going beyond the analysis of textual information. The aim is to provide tools to help combat the abusive language phenomenon.

7.4.1 Author Profiling Shared Task

The main objective of the *Profiling Hate Speech Spreaders on Twitter Task at PAN 2021*[3] task is to determine whether the author of a Twitter feed is a HSS or not (Rangel et al., 2021). In this task, a user is considered a HSS if they have at least 10 hateful messages among their last posts.

Corpus. We build a corpus for English and Spanish. In each language, the corpus consists of 300 users from which their last 200 tweets were selected. These users are distributed in a balanced way, i.e. there are 150 users per class (HSS and non-HSS). The division of the corpus into training and test sets was based on a ratio of 2/3 for training and the rest for testing.

Main Methods. Most of the proposed strategies used traditional machine learning models such as SVM, Random Forest, Logistic Regression, and Adaboost. Only a few participants used methods based on deep learning such as Fully Connected Neural Networks, Recurrent Neural Networks, Convolutional Neural Networks, and BERT. The most notable characteristic of the proposals is that most participants used a transformer-based approach to represent the texts, with BERT being the most commonly used transformer.

Results. In order to analyze and compare the approaches, we calculate the individual accuracy for the distinction between the two classes. Then we averaged the accuracy values per language to obtain the final ranking. The best result in English was obtained with BERT and Logistic Regression

[3] https:/ /pan.webis.de/clef21/pan21-web/author-profiling.html

(Dukic and Krzic, 2021). The best result in Spanish was obtained using a 100-dimensional word embedding representation to feed a Convolutional Neural Network (Siino et al., 2021). Other top participants used transformer-based architectures such as BERT, BERTTWEET, ROBERTa, BETO, and one team used a meta-classifier fed with combinations of n-grams. The general results show that it is possible to automatically identify HSS with high accuracy using only textual elements. However, we point out that false positives need to be taken into account as they are almost twice as frequent as false negatives, especially in Spanish, which could have ethical or legal implications.

7.4.2 Users Networks Analysis

Although the results of the task in the PAN 2021 gave us a vision of the possibility of characterizing possible HSS through their post, we are also interested in evaluating how these users tend to interact on Twitter. In the following experiments, we focus on investigating user networks through graph analysis.

7.4.2.1 User Network Construction

Datasets We use two datasets for hate speech detection, which we used in Chapter 3: FOUNTA (Founta et al., 2018) and W&H (W&H) (Waseem and Hovy, 2016). We chose these datasets because they contain the identifier of each tweet, which allowed us to obtain all the data needed for network construction. We use the Twitter API to download the information from each dataset. In this way, we obtain the author of each tweet and the list of retweets of each tweet.

Settings. After downloading the tweets for each dataset, we extracted the following data for each tweet t:

- Author U_t of t (Twitter user).
- Retweets list of t. In turn, we obtained the author of each retweet to generate the list of users R_{U_t} who have retweeted a tweet from U_t at least once.

Using this information, we created a graph for each dataset with the following characteristics:

- **Nodes**: Users. The matrix representation of the nodes is $X = I_{nxn}$ where n is the number of users in the graph, i.e. we use the identity matrix whose dimension is determined by n.

- **Edges**: Relation based on retweets. An edge e_{uv} exists if the user v is in the list of R_u. The weight of this edge is the number of times that v has retweeted a tweet from u. We then created the adjacency matrix A_{nxn} with this information.

- **Node classes**: We label users in one of the HSS or non-HSS classes. We consider a user to be HSS if they have at least one tweet labeled as hateful in the original dataset.

7.4.2.2 Network Characteristics

Table 7.11 gives an overview of the characteristics of the graphs generated for each dataset. Looking at these characteristics, we notice that the graphs do not turn out to be fully connected. There is a large number of isolated nodes so only 19% (2646) and 51.8% (1143) of the nodes have at least one connection to other nodes in FOUNTA and W&H respectively. Moreover, the power law exponent is less than 2, which is not sufficient to say that the graph degree distribution follows the power law distribution.

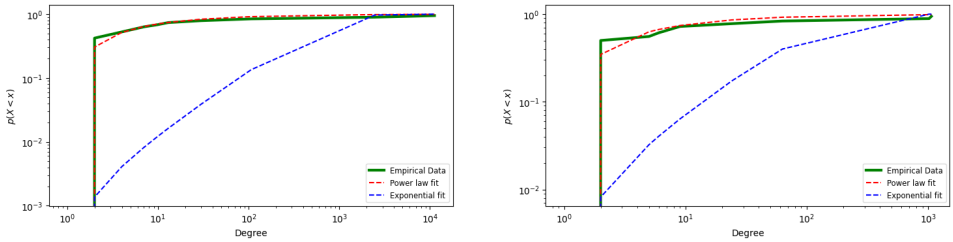
Caracteristics	**FOUNTA**	**W&H**
Nodes	13884	2208
# HSS nodes/# non-HSS nodes	4814/9070	819/1389
Edges	1589	1090
Isolated nodes	11238	1065
Loops	22	3
Density	0.0000082	0.0002237
Minimum degree	0	0
Maximum degree	28	343
Average degree	0.23	0.9873
Assortative coefficiente	-0.013661	-0.373988
# of connected component	12317	1162
Size of giant component	29	572
Power law exponent	1.525	1.614

Table 7.11: User network characteristics.

In order to verify this finding, we visualize in Figure 7.9 the cumulative distribution function of the degree distribution of the graphs, together with the cumulative distribution function of the power law distribution and the exponential distribution. For both FOUNTA and W&H, we observe a strong similarity between the real graph data and the power law distribution. This indicates that the degree distribution of the graphs follows the power law distribution. We also validate this result in Table 7.12, which shows the result of comparing two distributions.[4] If the value of the likelihood coefficient is

[4]To compare distributions we use the *distribution_compare* method from the *powerlaw* python library.

greater than zero, distribution 1 is preferable. In both graphs, the values are large, which confirms the comment about Figure 7.9.



(a) FOUNTA (b) W&H

Figure 7.9: Comparison between distributions.

		Likelihood coefficient	
Distribution 1	Distribution 2	**FOUNTA**	**W&H**
Power law	Exponential	76.8423	48.35

Table 7.12: Distribution comparison.

7.4.2.3 Influence Maximization

The presence of a power law distribution indicates that a few nodes, known as hubs, play a crucial role in connecting different parts of the network. Removing a random node is less likely to disrupt the network than removing a hub. This is an important conclusion that directs the investigation toward the search for hubs that are HSS to control the spread of hatred on Twitter.

Note that a hateful tweet originated from an isolated node or a node with few connections (no hub) does not easily spread throughout the network. To study this phenomenon, we rely on the field of influence maximization in graphs. Influence maximization is a concept in network theory and social network analysis that involves identifying a small set of nodes in a graph, called seeds, that when targeted with certain information, can maximize the spread of information throughout the network (Li et al., 2018; Patwardhan et al., 2023).

We use the Independent Cascade Model (Chen et al., 2009), a diffusion model for modeling the diffusion of information (influence). This model describes how information spreads from activated nodes to their neighbors over time. Our goal is to determine the amount of seeds necessary to disseminate information to a certain percentage of the network. To determine the seed set, we first select a random node as an active node and add more nodes at each stage of an iterative process. In one stage of the process, the highest degree node that is not active is selected among the neighboring nodes of

the active nodes. Then this node is activated with a probability $p = 0.01$. The set of seeds (active node) is used to simulate information propagation and determine the number of nodes to which the information reaches. The process is completed when this quantity is the desired one.

Table 7.13 shows the number of seeds needed to reach different percentages of the networks. In the networks of both datasets, the percentage of seeds is very close to the percentage of the network to be reached. This result refers to seed sets of non-hub nodes and confirms that what matters most is to find the HSS hubs because the HSS users that are not hubs do not seem to disseminate information significantly.

	% of seeds	
% of graph to reach	**FOUNTA**	**W&H**
10%	3.4%	8.7%
30%	24.1%	28.8%
50%	44.8%	49.1%
75%	68.9%	74.4%

Table 7.13: Percentage of needed seeds in Influence Maximization.

7.4.2.4 Comparison of Real Networks with Random Graphs

Another way to characterize real graphs is to compare them with random graphs. Random graphs are mathematical models for graphs in which the edges between nodes are added based on a random process. They are well-known models that are often used in graph theory to study the properties of graphs that result from random connections between nodes. There are different models of random graphs and we use four of them in our research:

- Erdős-Rényi (Erdős and Renyi, 1959): In a graph with n nodes, every node in it is connected to another node with a probability p. We determined p as the ratio between the number of edges in our real network divided by the number of edges of the complete graph with the same number of nodes.

- Barabási-Albert (Barabási and Albert, 1999): A graph of nodes is grown by attaching new nodes, each with m edges that are preferentially attached to existing nodes with a high degree.

- Chung-Lu (Chung and Lu, 2002): Given a sequence of expected node degrees $W = \{w_0, w_1, ..., w_{n-1}\}$ of length n this algorithm assigns an edge between node u and node v with probability $p = \frac{w_u w_v}{\sum_k w_k}$.

- Watts–Strogatz (Watts and Strogatz, 1998): This model has been used to explain the observed small-world phenomenon in various real-world networks, including social networks.

Graph representation. For each graph, we create a set of subgraphs. Then we represent each subgraph as a vector of the features listed in Table 7.11. In this way, we have a collection of data consisting of the feature vectors whose label is the original graph from which the corresponding subgraph was extracted.

Comparison strategy. We rely on the classification task to distinguish between two sets of feature vectors (subgraphs) extracted from random graphs. We train a fully connected neural network for each pair of random graphs. Then, with each of these classifiers, we compute the probability that the subgraphs of a real graph belong to each of the random graphs used to train the classifier. If the classifier discriminates for one of the classes, then the underlying random graph generation model is better in the sense that it generates graphs with certain properties.

Tables 7.14 and 7.15 show the average of the probabilities calculated for each pair of random graphs for FOUNTA and W&H, respectively. The results show that it is not possible to characterize the graphs constructed from real data with any of the random graphs examined.

	Erdős-Rényi	Barabási-Albert	Chung-Lu	Watts–Strogatz
Erdős-Rényi	-	0.4487	0.4357	0.4384
Barabási-Albert	0.5513	-	0.4400	0.4337
Chung-Lu	0.5643	0.5600	-	0.5000
Watts–Strogatz	0.5616	0.5663	0.5000	-

Table 7.14: Pairwise model comparison for FOUNTA.

	Erdős-Rényi	Barabási-Albert	Chung-Lu	Watts–Strogatz
Erdős-Rényi	-	0.5515	0.4655	0.6226
Barabási-Albert	0.4485	-	0.4731	0.6307
Chung-Lu	0.5345	0.5269	-	0.6814
Watts–Strogatz	0.3774	0.3693	0.3186	-

Table 7.15: Pairwise model comparison for W&H.

7.4.2.5 Node Classification

Node classification is a task in which the labels of neighboring nodes are used to predict missing node labels in a graph. Our latest analysis of user networks focuses on this task to investigate how GNN-based models can help identify HSS users from a graph of users and their relationships. We use a GNN-based model with two convolutional layers (CGNN). We first consider the user graphs described above (see Section 7.4.2.1) to obtain user-level classification results (CGNN_{Users}). Then we add the information about

the texts of each user (CGNN$_{Users\&Texts}$). In this second experiment, we transform the matrix with the information of the nodes X. We replace the identity matrix I_{nxn} (n is the number of nodes) with a text representation matrix. In row i, which corresponds to the user u_i, we place the vector obtained with TFIDF from the concatenation of all texts of u_i. In both experiments, we used 70% of the users to train the model and the rest for evaluation.

	FOUNTA	**W&H**
$CGNN_{Users}$	0.6379	0.7004
$CGNN_{Users\&Texts}$	0.8952	0.7871

Table 7.16: Graph neural networks for user classification as hate speech spreader.

Table 7.16 shows the results for each dataset. The results show that the CGNN is able to classify HSS users based on the relationships in the graph. Furthermore, we find that the performance can be improved by including other information such as user texts.

7.4.2.6 Conclusions and Future Work

In this section, we examine different aspects of the abusive language phenomenon considering the users of the social network Twitter. On the one hand, we find that it is possible to automatically identify HSS users based on a stream of publications. On the other hand, we investigate different features of two datasets extracted from Twitter. The results suggest that user networks are characterized by a few users who play a crucial role in connecting different parts of the network. They are the most important to identify, as nodes with few connections do not seem to achieve widespread dissemination of information. These are interesting results that serve as a starting point for future research, in which various aspects must be taken into account. Regarding the construction of the graph, the follower-followee relationship can be investigated, as well as other user information can be considered for its representation. Instead of the identity matrix, we can use a matrix where each row, corresponding to a user u, contains a vector of features of u extracted with the Twitter API, such as the number of followers, the number of posts, etc. Furthermore, other criteria can also be evaluated to define a user as HSS. Finally, it should be noted that GNN-based models offer the possibility to consider other tasks besides node classification, such as link prediction, which can help predict the connection between a pair of users in the graph with an incomplete adjacency matrix.

7.5 Ethical Discussion

Our work may have some ethical concerns. First, we must point out that the goal of our research is solely to minimize the harm that the abusive language phenomenon can cause on social networks. Our tools should only be used for the benefit of users. For example, our findings can help develop systems to detect and prevent the spread of abusive language in environments with low information availability. There are also some ethical concerns regarding the collection and dissemination of data for the creation of the dataset proposed in the shared task at PAN 2021. Likewise, there may be concerns about user privacy when investigating Section 7.4. We must clarify that the identity of the users used throughout the research of this PhD has been kept completely anonymous, both in our experiments and in the publication of the results. It is in no way our intention to stigmatize the users we used for our study.

Chapter 8

Conclusions and Future Work

8.1 Conclusions

Abusive language detection is crucial to fostering a safe, inclusive, and positive online environment. While significant progress has been made in this task thanks to advances in natural language processing and artificial intelligence, these advances also bring with them a number of challenges. Detecting abusive language in an environment with limited data is a major challenge. The models may not have enough examples to understand the nuances of the language. Translating models from languages with many resources to languages with few resources is not trivial and may result in lower accuracy.

The work presented in this thesis has focused on the study of abusive language detection considering different aspects. In Part I, we investigated how models can reflect existing biases in the training data that can lead to unfair detection of abusive language. The first step in this investigation was to find a set of terms where abusive detection might be biased. To this end, we proposed two methods to extract potentially abusive keywords from datasets. One of the methods is based on the BERT attention mechanism and the other on statistics computed from word frequencies related to the class of abusive texts. Although the keywords extracted by the two methods do not overlap much, we found that both methods mainly extract abusive words. Once the keywords were extracted, the second step of our research was to investigate: 1) the bias of the models toward these keywords, 2) how this bias can be mitigated, and 3) how the performance of the models is affected by mitigating the bias. The experimental results showed that the bias can be reduced when fine-tuning the models with abusive texts in which the keywords are not present and that this reduction can mean a performance improvement. In Part II, we investigated the role of models based on graph neural networks for abusive language detection. On the one hand, we proposed a text representation framework designed to discriminate abusive texts from other texts. On the other hand, we evaluated the use of models based

on convolutional graph neural networks for classifying texts as abusive or non-abusive. In general, we found that this type of models has the potential to outperform transformer-based models for detecting abusive language. Next, in Part III, we used data augmentation techniques to increase the size of the training dataset for the detection of abusive language. We considered two well-known techniques based on the principle of vicinal risk minimization and proposed a variant for one of these techniques. In the experiments, we evaluated how the results of few-shot learning (the use of very little data to fine-tune a pre-trained transformer-based model) are improved by increasing the size of the available data using those data augmentation techniques. Finally, in Part IV we further analyzed the results that were obtained in the parts mentioned above, and we described further experiments we have done.

In particular, in the framework of our PhD thesis, we explored abusive language detection in low-resource settings. The results we observed in the course of our study allow us to answer the research questions posed in the introduction of this thesis:

Research question about bias in the models

- RQ1: *Could bias toward potential abusive keywords in the models affect the performance of abusive language detection in low-resource settings?*

  The result of our research showed that modifying the bias toward potential abusive keywords in the models generally varies the performance of abusive language detection. However, in low-resource settings, it was not possible to determine how the bias needs to be varied in order to improve the performance. In Chapter 3, we observed how the bias toward abusive keywords in a model can be mitigated by adjusting the model accordingly using texts with certain features. Specifically, we noticed that the bias can be mitigated by using abusive texts in which the abusive keywords do not appear. We verified the mitigation of bias using two different metrics to measure bias (a fairness-based metric and a metric based on ROC-AUC) and found a correspondence between this reduction and an increase in the performance of the model. While this is a promising result, we did not observe the same behavior in low-resource settings. We verified it by extending the experiments of Chapter 3 to small datasets in Part IV. On the one hand, we found that bias was only reduced in a few cases. The reason may lie in the lack of texts of the environment where we reproduced the experiments, which prevented a good fit of the model to appropriately mitigate the bias. Consequently, although the bias modification affected the performance of the models in low-resource settings, the results are not conclusive enough to establish a correspondence between the decrease/increase in bias and the variation in performance.

Research question about graph-based models

- RQ2: *What is the contribution of models based on graph neural networks for abusive language detection in low-resource settings?*

  According to our investigation, graph neural network models are promising for abusive language detection, especially in low-resource settings. In Part II, we evaluated the suitability of graph auto-encoders to obtain a discriminatory representation between abusive and non-abusive texts. We also evaluated a convolutional graph neural network for the detection of abusive language. Under conditions of scarce data, graph-based models showed better performance than transformer-based models. In order to verify these results, we conducted further experiments in Part IV to evaluate the performance of graph-based models in low-resource settings. First, we verified the superiority of graph neural networks over transformer-based models in this type of setting. On the other hand, we found that different types of operators in graph neural networks can improve the results of classical convolutional graph neural networks.

Research question about data augmentation

- RQ3: *What is the contribution of data augmentation for abusive language detection in low-resource settings?*

  Our study revealed that data augmentation can be an effective strategy to improve abusive language detection in low-resource settings. In Part III, we evaluated how the cross-lingual few-shot learning task can be enhanced with three techniques based on the principle of vicinal risk minimization (SSMBA, MIXUP, and MIXAG). To do this, we consider a model trained to detect abuse in English and fine-tune it with some examples for another target language. Then, the number of examples in the target language was augmented with vicinal risk minimization techniques. An improvement in results was observed after using a model fine-tuned with this number of new samples. Later, in Part IV we extend the experiments using the well-known Easy Data Augmentation (EDA) strategy. First, we found that the performance of abusive language detection enhanced as the amount of data increased and noted that this improvement is particularly beneficial in low-resource settings. Second, we examined the number of new examples that should be generated per initial instance and found that for smaller training sets, generating many augmented instances leads to large performance improvements. In contrast, for larger training sets, adding more than four augmented instances per original text was unhelpful. We believe that this is because models tend to generalize properly when large amounts of real data are available.

To sum up, we believe that the answers to the research questions show that both the use of models based on graph neural networks as well as data augmentation techniques can lead to the improvement of abusive language detection in low-resource settings. Besides, we noted that bias mitigation strategies can fail when the data is limited and that it is, therefore, necessary to resort to other strategies in these settings.

Additionally, we introduced a study on abusive language detection from the user's perspective. We found that it is possible to automatically identify potential haters based on a stream of publications. Furthermore, we noted that important insights can be gained from analyzing the graph of relationships between users. In particular, we observed that a few users who play a crucial role in connecting different parts of the network may be the most important to identify to prevent the spread of abusive messages on Twitter.

8.2 Future Work

In the framework of this PhD thesis, we addressed the task of abusive language detection and identified future research directions that we intend to explore in order to extend the study conducted. One of these directions is to investigate further strategies to mitigate bias with limited data. We found that one way to mitigate the bias toward abusive keywords in a model is to fit it with a set of abusive texts without the presence of any of these keywords. We verified that this strategy fails when the number of texts is very small since the model seems to be unable to adapt to reduce the bias. Therefore, we want to study in future research more sophisticated strategies to mitigate bias when there is not enough data. We believe that this will allow us to establish a correspondence between variation in bias and performance in low-resource settings.

Another direction we want to investigate concerns models based on graph neural networks. We acknowledged their potential for abusive language detection and believe that the study should be extended to explore the flexibility of this type of models in representing different types of data. For example, we can represent not only texts but also abusive keywords or phrases as graph nodes and also represent a different type of relationship between the graph nodes. In addition, we are interested in the task of multimodal abusive language detection, where we will combine images and texts. In the framework of our PhD, we did some preliminary work to investigate abusive language detection from a multimodal perspective. In (De la Peña Sarracén et al., 2020), we proposed a multimodal model that combines BERT for text analysis and VGG19[1] for images.

Finally, we are interested in extending the study on abusive language detection from the perspective of the users. In Part IV, we found attractive

[1] https://pytorch.org/vision/main/models/generated/torchvision.models.vgg19.html

results that serve as a starting point for future research. We noticed that by creating a graph where nodes represent users and edges represent the number of retweets between users, we can analyze features that help us understand the possible behaviors in the network. For example, we found that there are some important nodes known as hubs. They are the most important ones to locate to prevent abusive messages from spreading through the network. We also observed that with this graph representation and with models based on convolutional graph neural networks, it is possible to classify users as hate speech spreaders with reasonable performance. In this direction, we intend to investigate different representations for the graphs in terms of i) the properties of the users for the nodes and ii) the relationship between the users for the edges. Furthermore, we want to extend our study to a larger dataset. Finally, we believe that it is worth investigating other social networks besides Twitter where abusive comments can also affect the relationship between users.

8.3 Publications

The different contributions of this thesis have been materialized in several publications. Below we sum up the different scientific contributions highlighting their quality using the reference scoring system, i.e. the Core Conference Ranking and the journal impact factor.

1. **De la Peña Sarracén, G.L.** and Rosso, P. (2023). Systematic Keyword and Bias Analyses in Hate Speech Detection. Information Processing & Management, 60(5), pp. 103433.
   (Impact Factor: 7.466, Q1).

2. **De la Peña Sarracén, G.L.** and Rosso, P. (2023). Offensive Keyword Extraction Based on the Attention Mechanism of BERT and the Eigenvector Centrality using a Graph Representation. Personal and Ubiquitous Computing, 27(1), pp. 45-57.
   (Impact Factor: 3.406, Q2).

3. **De la Peña Sarracén, G.L.**, Rosso, P., Litschko, R., Glavas, G., and Ponzetto, S. P. (2023). Vicinal Risk Minimization for Few-Shot Cross-lingual Transfer in Abusive Language Detection. Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 4069-4085. **(Core A*)**.

4. Bevendorff J., Chulvi B., **De la Peña Sarracén, G.L.**, Kestemont M., Manjavacas E., Markov I., Mayerl M., Potthast M., Rangel F., Rosso P., Stamatatos E., Stein B., Wiegmann M., Wolska M., Zangerle E. (2021). Overview of PAN 2021: Authorship Verification, Profiling Hate Speech Spreaders on Twitter, and Style Change Detection (2021).

In: Proc. 43rd European Conf. on Information Retrieval, ECIR-2021, Springer-Verlag, LNCS(12657), pp. 567-573. **(Core A)**.

5. **De la Peña Sarracén, G.L.** and Rosso, P. (2022). Unsupervised Embeddings with Graph Auto-Encoders for Multi-Domain and Multilingual Hate Speech Detection. Proceedings of the Thirteenth Language Resources and Evaluation Conference, pp. 2196-2204. **(Core B)**.

6. **De la Peña Sarracén, G.L.** and Rosso, P. (2022). Convolutional Graph Neural Networks for Hate Speech Detection in Data-Poor Settings. International Conference on Applications of Natural Language to Information Systems, pp. 16-24. Springer International Publishing. **(Core C)**.

7. Chinea Rios, M., Müller, T., **De la Peña Sarracén, G.L.**, Rangel, F., and Franco Salvador, M. (2022). Zero and Few-shot Learning for Author Profiling. In International Conference on Applications of Natural Language to Information Systems, pp. 333-344. Springer International Publishing. **(Core C)**.

8. Rangel, F., **De la Peña Sarracén, G.L.**, Chulvi, B., Fersini, E., and Rosso, P. (2021). Profiling Hate Speech Spreaders on Twitter Task at PAN 2021. In Proceedings of the Working Notes of CLEF 2021, Conference and Labs of the Evaluation Forum, pp. 1772-1789.

9. Bevendorff J., Chulvi B., **De la Peña Sarracén, G.L.**, Kestemont M., Manjavacas E., Markov I., Mayerl M., Potthast M., Rangel F., Rosso P., Stamatatos E., Stein B., Wiegmann M., Wolska M., Zangerle E. (2021). Overview of PAN 2021: Authorship Verification, Profiling Hate Speech Spreaders on Twitter, and Style Change Detection. (extended version). In: Proc. 12th Int. Conf. of the CLEF Association, CLEF 2021, Virtual Event, September 21–24, CLEF-2021, Springer-Verlag, LNCS(12880), pp. 419-431.

10. **De la Peña Sarracén, G.L.** and Rosso, P. (2021). Multi-task Learning to Analyze the Influence of Offensive Language in Hate Speech Detection. In Multimodal Hate Speech Workshop 2021, pp. 13-18.

11. Korenčić, D., Grubišić, I., **De la Peña Sarracén, G.L.**, Toselli, A.H., Chulvi, B., and Rosso, P. (2022). Tackling Covid-19 Conspiracies on Twitter using BERT Ensembles, GPT-3 Augmentation, and Graph NNs. In: Working Notes Proceedings of the MediaEval 2022 Workshop Bergen, Norway and Online, 12-13 January. Vol-3583.

12. **De la Peña Sarracén, G.L.**, Rosso, P., and Giachanou, A. (2020). PRHLT-UPV at SemEval-2020 Task 8: Study of Multimodal Techniques for Memes Analysis. In Proceedings of the Fourteenth Workshop on Semantic Evaluation, pp. 908-915: Barcelona, Spain, collocated with The 28th International Conference on Computational Lingustics (COLING-2020).

13. **De la Peña Sarracén, G.L.** and Rosso, P. (2020). PRHLT-UPV at Semeval-2020 task 12: Bert for Multilingual Offensive Language Detection. In Proceedings of the Fourteenth Workshop on Semantic Evaluation, pp. 1605-1614: Barcelona, Spain, collocated with The 28th International Conference on Computational Lingustics (COLING-2020).

Bibliography

Oliver Adams, Adam Makarucha, Graham Neubig, Steven Bird, and Trevor Cohn. 2017. Cross-Lingual Word Embeddings for Low-Resource Language Modeling. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 937–947, Valencia, Spain. Association for Computational Linguistics.

Khurshid Ahmad, Lee Gillam, Lena Tostevin, et al. 1999. University of Surrey Participation in TREC8: Weirdness Indexing for Logical Document Extrapolation and Retrieval (WILDER). In *TREC*, pages 1–8.

Usman Ahmed and Jerry Chun-Wei Lin. 2022. Deep Explainable Hate Speech Active Learning on Social Media Data. *IEEE Transactions on Computational Social Systems*, pages 1–11.

Muhammad Pervez Akhter, Zheng Jiangbin, Irfan Raza Naqvi, Mohammed AbdelMajeed, and Tehseen Zia. 2021. Abusive Language Detection from Social Media Comments using Conventional Machine Learning and Deep Learning Approaches. *Multimedia Systems*, pages 1–16.

Peyman Alavi, Pouria Nikvand, and Mehrnoush Shamsfard. 2021. Offensive Language Detection with BERT-based models, By Customizing Attention Probabilities. *CoRR*, abs/2110.05133.

Raza Ali, Umar Farooq, Umair Arshad, Waseem Shahzad, and Mirza Omer Beg. 2022. Hate Speech Detection on Twitter using Transfer Learning. *Computer Speech & Language*, 74:101365.

Fatimah Alkomah and Xiaogang Ma. 2022. A Literature Review of Textual Hate Speech Detection Methods and Datasets. *Information*, 13(6):273.

Sai Saket Aluru, Binny Mathew, Punyajoy Saha, and Animesh Mukherjee. 2020. Deep Learning Models for Multilingual Hate Speech Detection. *CoRR*, abs/2004.06465.

Sai Saketh Aluru, Binny Mathew, Punyajoy Saha, and Animesh Mukherjee. 2021. A Deep Dive into Multilingual Hate Speech Classification. In *Ma-*

chine Learning and Knowledge Discovery in Databases. Applied Data Science and Demo Track: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part V, pages 423–439. Springer.

Manish Anand, Kishan Bhushan Sahay, Mohammed Altaf Ahmed, Daniyar Sultan, Radha Raman Chandan, and Bharat Singh. 2023. Deep Learning and Natural Language Processing in Computation for Offensive Language Detection in Online Social Networks by Feature Selection and Ensemble Classification Techniques. *Theoretical Computer Science*, 943:203–218.

Antreas Antoniou and Amos Storkey. 2019. Assume, Augment and Learn: Unsupervised Few-shot Meta-learning via Random Labels and Data Augmentation. *CoRR*, abs/1902.09884.

Xiong Ao, Xin Yu, Derong Liu, and Hongkang Tian. 2020. News Keywords Extraction Algorithm based on TextRank and Classified TF-IDF. In *2020 International Wireless Communications and Mobile Computing (IWCMC)*, pages 1364–1369. IEEE.

Aymé Arango, Jorge Pérez, and Barbara Poblete. 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*, pages 45–54.

Farid Arthaud, Rachel Bawden, and Alexandra Birch. 2021. Few-shot learning through contextual data augmentation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1049–1062, Online. Association for Computational Linguistics.

Ubaid Azam, Hammad Rizwan, and Asim Karim. 2022. Exploring Data Augmentation Strategies for Hate Speech Detection in Roman Urdu. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4523–4531, Marseille, France. European Language Resources Association.

Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. 2017. Deep Learning for Hate Speech Detection in Tweets. In *Proceedings of the 26th international conference on World Wide Web companion*, pages 759–760.

Agathe Balayn, Jie Yang, Zoltan Szlavik, and Alessandro Bozzon. 2021. Automatic Identification of Harmful, Aggressive, Abusive, and Offensive Language on the Web: A Survey of Technical Biases Informed by Psychology Literature. *ACM Transactions on Social Computing (TSC)*, 4(3):1–56.

Esma Balkir, Isar Nejadgholi, Kathleen C Fraser, and Svetlana Kiritchenko. 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*, pages 2672–2686.

Albert-László Barabási and Réka Albert. 1999. Emergence of Scaling in Random Networks. *science*, 286(5439):509–512.

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Nozza Debora, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, Manuela Sanguinetti, et al. 2019a. Semeval-2019 Task 5: Multilingual Detection of Hate Speech against Immigrants and Women in Twitter. In *13th International Workshop on Semantic Evaluation*, pages 54–63. Association for Computational Linguistics.

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Nozza Debora, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, Manuela Sanguinetti, et al. 2019b. Semeval-2019 Task 5: Multilingual Detection of Hate Speech against Immigrants and Women in Twitter. In *13th International Workshop on Semantic Evaluation*, pages 54–63. Association for Computational Linguistics.

Elisa Bassignana, Valerio Basile, and Viviana Patti. 2018. Hurtlex: A Multilingual Lexicon of Words to Hurt. In *5th Italian Conference on Computational Linguistics, CLiC-it 2018*, volume 2253, pages 1–6. CEUR-WS.

Markus Bayer, Marc-André Kaufhold, and Christian Reuter. 2022. A Survey on Data Augmentation for Text Classification. *ACM Computing Surveys*, 55(7):1–39.

Michael W Berry and Jacob Kogan. 2010. *Text Mining: Applications and Theory*. John Wiley & Sons.

Irina Bigoulaeva, Viktor Hangya, and Alexander Fraser. 2021a. Cross-Lingual Transfer Learning for Hate Speech Detection. In *Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 15–25, Kyiv. Association for Computational Linguistics.

Irina Bigoulaeva, Viktor Hangya, and Alexander Fraser. 2021b. Cross-Lingual Transfer Learning for Hate Speech Detection. In *Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 15–25. Association for Computational Linguistics.

Michał Bilewicz and Wiktor Soral. 2020. Hate Speech Epidemic. The Dynamic Effects of Derogatory Language on Intergroup Relations and Political Radicalization. *Political Psychology*, 41:3–33.

Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2019. Nuanced Metrics for Measuring Unintended Bias with Real Data for Text Classification. In *Companion Proceedings of The 2019 World Wide Web Conference*, pages 491–500.

Cristina Bosco, Dell'Orletta Felice, Fabio Poletto, Manuela Sanguinetti, Tesconi Maurizio, et al. 2018. Overview of the Evalita 2018 Hate Speech Detection Task. In *Ceur workshop proceedings*, volume 2263, pages 1–9. CEUR.

Florian Boudin. 2013. A Comparison of Centrality Measures for Graph-Based Keyphrase Extraction. In *Proceedings of the sixth international joint conference on natural language processing*, pages 834–838.

Sergey Brin and Lawrence Page. 1998. The Anatomy of a Large-scale Hypertextual Web Search Engine. In *Proceedings of the Seventh International Conference on World Wide Web*, pages 107–117.

Stefan Büttcher, Charles LA Clarke, and Gordon V Cormack. 2016. *Information retrieval: Implementing and evaluating search engines*. Mit Press.

Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Jorge, Célia Nunes, and Adam Jatowt. 2020. YAKE! Keyword Extraction from Single Documents Using Multiple Local Features. *Information Sciences*, 509:257–289.

Tommaso Caselli, Valerio Basile, Jelena Mitrović, and Michael Granitzer. 2021a. HateBERT: Retraining BERT for Abusive Language Detection in English. In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, pages 17–25, Online. Association for Computational Linguistics.

Tommaso Caselli, Arjan Schelhaas, Marieke Weultjes, Folkert Leistra, Hylke van der Veen, Gerben Timmerman, and Malvina Nissim. 2021b. DALC: the Dutch Abusive Language Corpus. In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, pages 54–66, Online. Association for Computational Linguistics.

Galo Castillo-López, Arij Riabi, and Djamé Seddah. 2023. Analyzing Zero-Shot transfer Scenarios across Spanish Variants for Hate Speech Detection. In *Tenth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2023)*, pages 1–13.

Camilla Casula, Alessio Palmero Aprosio, Stefano Menini, and Sara Tonelli. 2020. FBK-DH at SemEval-2020 Task 12: Using Multi-channel BERT for Multilingual Offensive Language Detection. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1539–1545.

Camilla Casula and Sara Tonelli. 2020. Hate Speech Detection with Machine-Translated Data: The Role of Annotation Scheme, Class Imbalance and Undersampling. In *Proceedings of the Seventh Italian Conference on Computational Linguistics, CLiC-it 2020*, volume 2769. CEUR-WS. org.

Daniel Cer, Yinfei Yang, Sheng yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018a. Universal Sentence Encoder. *CoRR*, abs/1803.11175.

Daniel Cer, Yinfei Yang, Sheng yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018b. Universal Sentence Encoder. *CoRR*, abs/1803.11175.

Olivier Chapelle, Jason Weston, Léon Bottou, and Vladimir Vapnik. 2000. Vicinal Risk Minimization. *Advances in Neural Information Processing Systems*, 13:395—401.

Sneha Chaudhari, Gungor Polatkan, Rohan Ramanath, and Varun Mithal. 2019. An Attentive Survey of Attention Models. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 12:1 – 32.

Jiaao Chen, Derek Tam, Colin Raffel, Mohit Bansal, and Diyi Yang. 2023. An Empirical Survey of Data Augmentation for Limited Data Learning in NLP. *Transactions of the Association for Computational Linguistics*, 11:191–211.

Wei Chen, Yajun Wang, and Siyu Yang. 2009. Efficient Influence Maximization in Social Networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 199–208.

Lu Cheng, Ahmadreza Mosallanezhad, Yasin N Silva, Deborah L Hall, and Huan Liu. 2022. Bias Mitigation for Toxicity Detection via Sequential Decisions. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1750–1760.

Fan Chung and Linyuan Lu. 2002. Connected Components in Random Graphs with given Expected Degree Sequences. *Annals of combinatorics*, 6(2):125–145.

Yi-Ling Chung, Elizaveta Kuzmenko, Serra Sinem Tekiroglu, and Marco Guerini. 2019. CONAN–COunter NArratives through Nichesourcing: a Multilingual Dataset of Responses to Fight Online Hate Speech. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2819–2829. Association for Computational Linguistics.

Isobelle Clarke and Jack Grieve. 2017. [dimensions of abusive language on twitter]. In *Proceedings of the first workshop on abusive language online*, pages 1–10.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020a. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020b. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Michele Corazza, Stefano Menini, Elena Cabrio, Sara Tonelli, and Serena Villata. 2020a. A Multilingual Evaluation for Online Hate Speech Detection. *ACM Transactions on Internet Technology (TOIT)*, 20(2):1–22.

Michele Corazza, Stefano Menini, Elena Cabrio, Sara Tonelli, and Serena Villata. 2020b. Hybrid Emoji-Based Masked Language Models for Zero-Shot Abusive Language Detection. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 943–949, Online. Association for Computational Linguistics.

Haixing Dai, Zhengliang Liu, Wenxiong Liao, Xiaoke Huang, Zihao Wu, Lin Zhao, Wei Liu, Ninghao Liu, Sheng Li, Dajiang Zhu, et al. 2023. ChatAug: Leveraging ChatGPT for Text Data Augmentation. *CoRR*, abs/2302.13007.

Wenliang Dai, Tiezheng Yu, Zihan Liu, and Pascale Fung. 2020. Kungfupanda at SemEval-2020 Task 12: BERT-Based Multi-Task Learning for Offensive Language Detection. pages 2060–2066.

Marina Danilevsky, Kun Qian, Ranit Aharonov, Yannis Katsis, Ban Kawas, and Prithviraj Sen. 2020. A Survey of the State of Explainable AI for Natural Language Processing. pages 447–459.

Mithun Das, Somnath Banerjee, and Animesh Mukherjee. 2022. Data Bootstrapping Approaches to Improve Low Resource Abusive Language Detection for Indic Languages. In *Proceedings of the 33rd ACM Conference on Hypertext and Social Media*, pages 32–42.

Subhajeet Das, Koushikk Bhattacharyya, and Sonali Sarkar. 2023. Performance Analysis of Logistic Regression, Naive Bayes, KNN, Decision Tree, Random Forest and SVM on Hate Speech Detection from Twitter. *International Research Journal of Innovations in Engineering and Technology*, 7(3):24.

Gretel Liz De la Pena Sarracén, Reynaldo Gil Pons, Carlos Enrique Muniz Cuza, and Paolo Rosso. 2018. Hate Speech Detection using Attention-based LSTM. *EVALITA evaluation of NLP and speech tools for Italian*, 12:235.

Gretel Liz De la Peña Sarracén and Paolo Rosso. 2021. Offensive Keyword Extraction based on the Attention Mechanism of BERT and the Eigenvector Centrality using a Graph Representation. *Personal and Ubiquitous Computing*, pages 1–13.

Gretel Liz De la Peña Sarracén and Paolo Rosso. 2023. Systematic Keyword and Bias Analyses in Hate Speech Detection. *Information Processing & Management*, 60(5):103433.

Gretel Liz De la Peña Sarracén, Paolo Rosso, and Anastasia Giachanou. 2020. PRHLT-UPV at SemEval-2020 Task 8: Study of Multimodal Techniques for Memes Analysis. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 908–915.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019a. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019b. Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding.

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

Thomas G Dietterich. 1998. Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. *Neural computation*, 10(7):1895–1923.

Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2018. Measuring and Mitigating Unintended Bias in Text Classification. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 67–73.

Ashwin Geet D'Sa, Irina Illina, Dominique Fohr, Dietrich Klakow, and Dana Ruiter. 2020. Label Propagation-Based Semi-Supervised Learning for Hate Speech Classification. In *Proceedings of the First Workshop on Insights from Negative Results in NLP*, pages 54–59, Online. Association for Computational Linguistics.

David Dukic and Ana Sovic Krzic. 2021. Detection of Hate Speech Spreaders with BERT. In *CLEF (Working Notes)*, pages 1910–1919.

Mai ElSherief, Shirin Nilizadeh, Dana Nguyen, Giovanni Vigna, and Elizabeth Belding. 2018. Peer to Peer Hate: Hate Speech Instigators and Their Targets. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 12.

P Erdős and A Renyi. 1959. On Random Graphs. *Publ. Math. Debrecen*, 6:290–297.

Juuso Eronen, Michal Ptaszynski, Fumito Masui, Masaki Arata, Gniewosz Leliwa, and Michal Wroczynski. 2022. Transfer Language Selection for Zero-shot Cross-lingual Abusive Language Detection. *Information Processing & Management*, 59(4):102981.

Steven Y Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A Survey of Data Augmentation Approaches for NLP. pages 968–988.

Rodrigo Fernandes de Mello, Moacir Antonelli Ponti, Rodrigo Fernandes de Mello, and Moacir Antonelli Ponti. 2018. Statistical Learning Theory. *Machine Learning: A Practical Approach on the Statistical Learning Theory*, pages 75–128.

Elisabetta Fersini, Paolo Rosso, and Maria Anzovino. 2018. Overview of The Task on Automatic Misogyny Identification at IberEval 2018. *Ibereval sepln*, 2150:214–228.

Anderson Almeida Firmino, Cláudio Souza de Baptista, and Anselmo Cardoso de Paiva. 2021. Using Cross Lingual Learning for Detecting Hate Speech in Portuguese. In *Database and Expert Systems Applications: 32nd International Conference, DEXA 2021, Virtual Event, September 27–30, 2021, Proceedings, Part II*, pages 170–175. Springer.

Nazanin Firoozeh, Adeline Nazarenko, Fabrice Alizon, and Béatrice Daille. 2020. Keyword Extraction: Issues and Methods. *Natural Language Engineering*, 26(3):259–291.

Paula Fortuna and Sérgio Nunes. 2018a. A Survey on Automatic Detection of Hate Speech in Text. *ACM Computing Surveys (CSUR)*, 51(4):1–30.

Paula Fortuna and Sérgio Nunes. 2018b. A Survey on Automatic Detection of Hate Speech in Text. *ACM Computing Surveys (CSUR)*, 51(4):1–30.

Paula Fortuna, Juan Soler-Company, and Leo Wanner. 2021. How well do Hate Speech, Toxicity, Abusive and Offensive Language Classification Models Generalize across Datasets? *Information Processing & Management*, 58(3):102524.

Antigoni Founta, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis. 2018. Large Scale Crowdsourcing and Characterization of Twitter Abusive Behavior. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 12.

Simona Frenda, Viviana Patti, and Paolo Rosso. 2022. Killing Me Softly: Creative and Cognitive Aspects of Implicitness in Abusive Language Online. *Natural Language Engineering (JNLE)*, pages 1–22.

Simona Frenda, Viviana Patti, and Paolo Rosso. 2023. Killing me Softly: Creative and Cognitive Aspects of Implicitness in Abusive Language Online. *Natural Language Engineering*, 29(6):1516–1537.

Björn Gambäck and Utpal Kumar Sikdar. 2017. Using Convolutional Neural Networks to Classify Hate-Speech. In *Proceedings of the First Workshop on Abusive Language Online*, pages 85–90, Vancouver, BC, Canada. Association for Computational Linguistics.

Ismael Garrido-Muñoz, Arturo Montejo-Ráez, Fernando Martínez-Santiago, and L Alfonso Ureña-López. 2021. A Survey on Bias in Deep NLP. *Applied Sciences*, 11(7):3184.

Purnama Sari Br Ginting, Budhi Irawan, and Casi Setianingsih. 2019. Hate Speech Detection on Twitter using Multinomial Logistic Regression Classification Method. In *2019 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS)*, pages 105–111. IEEE.

Goran Glavaš, Mladen Karan, and Ivan Vulić. 2020. XHate-999: Analyzing and Detecting Abusive Language Across Domains and Languages. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6350–6365.

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*, pages 2–12.

Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30:1025–1035.

Kazi Saidul Hasan and Vincent Ng. 2014. Automatic Keyphrase Extraction: A Survey of the State of the Art. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1262–1273.

Xinghua Hu and Bin Wu. 2006. Automatic Keyword Extraction using Linguistic Features. In *Sixth IEEE International Conference on Data Mining-Workshops (ICDMW'06)*, pages 19–23. IEEE.

Vebjørn Isaksen and Björn Gambäck. 2020. Using Transfer-based Language Models to Detect Hateful and Offensive Language Online. In *Proceedings of the Fourth Workshop on Online Abuse and Harms*, pages 16–27.

Archika Jain and Sandhya Sharma. 2022. A Survey on Identification of Hate Speech on Social Media Post. In *2022 3rd International Conference on Computing, Analytics and Networks (ICAN)*, pages 1–6. IEEE.

David Jurgens, Libby Hemphill, and Eshwar Chandrasekharan. 2019. A Just and Comprehensive Strategy for Using NLP to Address Online Abuse. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3658–3666, Florence, Italy. Association for Computational Linguistics.

Kamil Kanclerz, Alicja Figas, Marcin Gruza, Tomasz Kajdanowicz, Jan Kocon, Daria Puchalska, and Przemyslaw Kazienko. 2021. Controversy and Conformity: from Generalized to Personalized Aggressiveness Detection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5915–5926, Online. Association for Computational Linguistics.

Shailendra Singh Kathait, Shubhrita Tiwari, Anubha Varshney, and Ajit Sharma. 2017. Unsupervised Key-phrase Extraction using Noun Phrases. *International Journal of Computer Applications*, 162(1):1–5.

Jasmeen Kaur and Vishal Gupta. 2010. Effective Approaches for Extraction of Keywords. *International Journal of Computer Science Issues (IJCSI)*, 7(6):144.

Marwa Khairy, Tarek M Mahmoud, and Tarek Abd-El-Hafeez. 2021. Automatic Detection of Cyberbullying and Abusive Language in Arabic Content on Social Networks: A Survey. *Procedia Computer Science*, 189:156–166.

Aman Khullar, Daniel Nkemelu, Cuong V Nguyen, and Michael L Best. 2023. Hate Speech Detection in Limited Data Contexts using Synthetic Data Generation. *ACM Journal on Computing and Sustainable Societies*.

Kelvin Kiema Kiilu, George Okeyo, Richard Rimiru, and Kennedy Ogada. 2018. Using Naïve Bayes Algorithm in Detection of Hate Tweets. *International Journal of Scientific and Research Publications*, 8(3):99–107.

Diederik P Kingma and Jimmy Ba. 2015a. Adam: A Method for Stochastic Optimization.

Diederik P. Kingma and Jimmy Ba. 2015b. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Thomas N Kipf and Max Welling. 2016. Variational Graph Auto-Encoders. *CoRR*, abs/1611.07308.

Thomas N. Kipf and Max Welling. 2017a. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the 5th International Conference on Learning Representations*, ICLR '17.

Thomas N. Kipf and Max Welling. 2017b. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*.

Svetlana Kiritchenko, Isar Nejadgholi, and Kathleen C Fraser. 2021. Confronting Abusive Language Online: A Survey from the Ethical and Human Rights Perspective. *Journal of Artificial Intelligence Research*, 71:431–478.

SV Kogilavani, S Malliga, KR Jaiabinaya, M Malini, and M Manisha Kokila. 2023. Characterization and Mechanical Properties of Offensive Language Taxonomy and Detection Techniques. *Materials Today: Proceedings*, 81:630–633. International Virtual Conference on Sustainable Materials (IVCSM-2k20).

Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, et al. 2020. Captum: A Unified and Generic Model Interpretability Library for Pytorch. *CoRR*, abs/2009.07896.

Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, and Hannaneh Hajishirzi. 2019. Text Generation from Knowledge Graphs with Graph Transformers. 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)*, pages 2284–2293. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *International Conference on Learning Representations*.

Siddique Latif, Aun Zaidi, Heriberto Cuayahuitl, Fahad Shamshad, Moazzam Shoukat, and Junaid Qadir. 2023. Transformers in Speech Processing: A Survey. *CoRR*, abs/2303.11607.

Eric Lavergne, Rajkumar Saini, György Kovács, and Killian Murphy. 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, EVALITA*, volume 2765.

João Augusto Leite, Diego Silva, Kalina Bontcheva, and Carolina Scarton. 2020. Toxic Language Detection in Social Media for Brazilian Portuguese: New Dataset and Multilingual Analysis. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 914–924, Suzhou, China. Association for Computational Linguistics.

Guohao Li, Matthias Muller, Ali Thabet, and Bernard Ghanem. 2019. Deepgcns: Can GCNs go as Deep as CNNs? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9267–9276.

Yuchen Li, Ju Fan, Yanhao Wang, and Kian-Lee Tan. 2018. Influence Maximization on Social Graphs: A Survey. *IEEE Transactions on Knowledge and Data Engineering*, 30(10):1852–1872.

Patrick Littell, David R. Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner, and Lori Levin. 2017. URIEL and LANG2VEC: Representing Languages as Typological, Geographical, and Phylogenetic Vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 8–14, Valencia, Spain. Association for Computational Linguistics.

Ruibo Liu, Guangxuan Xu, and Soroush Vosoughi. 2020. Enhanced Offensive Language Detection through Data Augmentation.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019a. Roberta: A Robustly Optimized BERT Pretraining Approach. *CoRR*, abs/1907.11692.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A Robustly Optimized BERT Pretraining Approach. *CoRR*, abs/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.

Ashe Magalhaes. 2019. Automating Online Hate Speech Detection: A Survey of Deep Learning Approaches. Master's thesis, School of Informatics, University of Edinburgh.

Jitendra Singh Malik, Guansong Pang, and Anton van den Hengel. 2022. Deep Learning for Hate Speech Detection: A Comparative Study. *CoRR*, abs/2202.09517.

Shervin Malmasi and Marcos Zampieri. 2018. Challenges in Discriminating Profanity from Hate Speech. *Journal of Experimental & Theoretical Artificial Intelligence*, 30(2):187–202.

Errol Mamani-Condori and José Ochoa-Luna. 2021. Aggressive Language Detection Using VGCN-Bert for Spanish Texts. In *Intelligent Systems: 10th Brazilian Conference, BRACIS 2021, Virtual Event, November 29–December 3, 2021, Proceedings, Part II 10*, pages 359–373. Springer.

Thomas Mandl, Sandip Modha, Prasenjit Majumder, Daksh Patel, Mohana Dave, Chintak Mandlia, and Aditya Patel. 2019a. Overview of the HASOC Track at Fire 2019: Hate Speech and Offensive Content Identification in Indo-European Languages. In *Proceedings of the 11th Forum for Information Retrieval Evaluation*, pages 14–17.

Thomas Mandl, Sandip Modha, Prasenjit Majumder, Daksh Patel, Mohana Dave, Chintak Mandlia, and Aditya Patel. 2019b. Overview of the HASOC Track at FIRE 2019: Hate Speech and Offensive Content Identification in Indo-European Languages. In *Proceedings of the 11th Forum for Information Retrieval Evaluation*, FIRE '19, page 14–17. Association for Computing Machinery.

Marta Marchiori Manerba and Sara Tonelli. 2021. Fine-grained Fairness Analysis of Abusive Language Detection Systems with Checklist. In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, pages 81–91.

Sanguinetti Manuela, Comandini Gloria, Elisa Di Nuovo, Simona Frenda, MARCO ANTONIO Stranisci, Cristina Bosco, Caselli Tommaso, Viviana Patti, Russo Irene, 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 (EVALITA 2020)*, pages 1–9. CEUR.

Lucia I Merlo, Berta Chulvi, Reynier Ortega, and Paolo Rosso. 2023. When Humour Hurts: Linguistic Features to Foster Explainability. (70):85–98.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing Order into Text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411.

Pushkar Mishra, Marco Del Tredici, Helen Yannakoudakis, and Ekaterina Shutova. 2019. Abusive Language Detection with Graph Convolutional Networks.

Khouloud Mnassri, Praboda Rajapaksha, Reza Farahbakhsh, and Noel Crespi. 2022. BERT-based Ensemble Approaches for Hate Speech Detection. In *GLOBECOM 2022-2022 IEEE Global Communications Conference*, pages 4649–4654. IEEE.

Marzieh Mozafari, Reza Farahbakhsh, and Noel Crespi. 2019. A BERT-based Transfer Learning Approach for Hate Speech Detection in Online Social Media. In *International Conference on Complex Networks and Their Applications*, pages 928–940. Springer.

Marzieh Mozafari, Reza Farahbakhsh, and Noel Crespi. 2020a. A BERT-based Transfer Learning Approach for Hate Speech Detection in Online Social Media. In *Complex Networks and Their Applications VIII: Volume 1 Proceedings of the Eighth International Conference on Complex Networks and Their Applications COMPLEX NETWORKS 2019 8*, pages 928–940. Springer.

Marzieh Mozafari, Reza Farahbakhsh, and Noël Crespi. 2020b. Hate Speech Detection and Racial Bias Mitigation in Social Media based on BERT Model. *PloS one*, 15(8):e0237861.

Marzieh Mozafari, Reza Farahbakhsh, and Noel Crespi. 2022. Cross-Lingual Few-Shot Hate Speech and Offensive Language Detection Using Meta Learning. *IEEE Access*, 10:14880–14896.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji,

Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual Generalization through Multitask Finetuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15991–16111, Toronto, Canada. Association for Computational Linguistics.

Raymond T Mutanga, Nalindren Naicker, and Oludayo O Olugbara. 2020. Hate Speech Detection in Twitter using Transformer Methods. *International Journal of Advanced Computer Science and Applications*, 11(9).

Arianna Muti, Francesco Fernicola, and Alberto Barrón-Cedeño. 2022. Misogyny and Aggressiveness Tend to Come Together and Together We Address Them. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4142–4148, Marseille, France. European Language Resources Association.

Matthew A Napierala. 2012. What is the Bonferroni correction? *Aaos Now*, pages 40–41.

Zara Nasar, Syed Waqar Jaffry, and Muhammad Kamran Malik. 2019. Textual Keyword Extraction and Summarization: State-of-the-Art. *Information Processing & Management*, 56(6):102088.

Francimaria RS Nascimento, George DC Cavalcanti, and Márjory Da Costa-Abreu. 2022. Unintended Bias Evaluation: An Analysis of Hate Speech Detection and Gender Bias Mitigation on Social Media Using Ensemble Learning. *Expert Systems with Applications*, 201:117032.

Mark EJ Newman. 2008. The Mathematics of Networks. *The new palgrave encyclopedia of economics*, 2(2008):1–12.

Nathan Ng, Kyunghyun Cho, and Marzyeh Ghassemi. 2020. SSMBA: Self-Supervised Manifold Based Data Augmentation for Improving Out-of-Domain Robustness. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1268–1283, Online. Association for Computational Linguistics.

Daniel Nkemelu, Harshil Shah, Michael Best, and Irfan Essa. 2022. Tackling Hate Speech in Low-resource Languages with Context Experts. In *Proceedings of the 2022 International Conference on Information and Communication Technologies and Development*, pages 1–11.

Debora Nozza. 2021. Exposing the limits of Zero-shot Cross-lingual Hate Speech Detection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 907–914, Online. Association for Computational Linguistics.

Debora Nozza, Claudia Volpetti, and Elisabetta Fersini. 2019. Unintended Bias in Misogyny Detection. In *IEEE/WIC/ACM International Conference on Web Intelligence*, pages 149–155.

Endang Wahyu Pamungkas, Valerio Basile, and Viviana Patti. 2023. Towards Multidomain and Multilingual Abusive Language Detection: A Survey. *Personal and Ubiquitous Computing*, 27(1):17–43.

Endang Wahyu Pamungkas and Viviana Patti. 2019. Cross-domain and Cross-lingual Abusive Language Detection: A Hybrid Approach with Deep Learning and a Multilingual Lexicon. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 363–370, Florence, Italy. Association for Computational Linguistics.

Chongyu Pan, Jian Huang, Jianxing Gong, and Xingsheng Yuan. 2019. Few-Shot Transfer Learning for Text Classification with Lightweight Word Embedding Based Models. *IEEE Access*, 7:53296–53304.

Sinno Jialin Pan and Qiang Yang. 2010. A Survey on Transfer Learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

Mei-hong Pana, Hongyi Xin, and Hongbin Shen. 2023. Semantic Transformation-Based Data Augmentation for Few-Shot Learning. *Available at SSRN 4321351*.

Raghavendra Pappagari, Piotr Zelasko, Jesús Villalba, Yishay Carmiel, and Najim Dehak. 2019. Hierarchical Transformers for Long Document Classification. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 838–844. IEEE.

Archit Parnami and Minwoo Lee. 2022. Learning from Few Examples: A Summary of Approaches to Few-Shot Learning. *CoRR*, abs/2203.04291.

Siddharth Patwardhan, Filippo Radicchi, and Santo Fortunato. 2023. Influence Maximization: Divide and Conquer. *Physical Review E*, 107(5):054306.

Hao Peng, Jianxin Li, Yu He, Yaopeng Liu, Mengjiao Bao, Lihong Wang, Yangqiu Song, and Qiang Yang. 2018. Large-Scale Hierarchical Text Classification with Recursively Regularized Deep Graph-CNN. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, page 1063–1072. International World Wide Web Conferences Steering Committee.

Nicola Pezzotti, Boudewijn P. F. Lelieveldt, Laurens van der Maaten, Thomas Höllt, Elmar Eisemann, and Anna Vilanova. 2017a. Approximated and User Steerable tSNE for Progressive Visual Analytics. *IEEE Transactions on Visualization and Computer Graphics*, 23(7):1739–1752.

Nicola Pezzotti, Boudewijn P. F. Lelieveldt, Laurens van der Maaten, Thomas Höllt, Elmar Eisemann, and Anna Vilanova. 2017b. Approximated and User Steerable tSNE for Progressive Visual Analytics. *IEEE Transactions on Visualization and Computer Graphics*, 23(7):1739–1752.

Georgios K Pitsilis, Heri Ramampiaro, and Helge Langseth. 2018a. Detecting Offensive Language in Tweets using Deep Learning. *CoRR*, abs/1801.04433.

Georgios K Pitsilis, Heri Ramampiaro, and Helge Langseth. 2018b. Effective Hate Speech Detection in Twitter Data using Recurrent Neural Networks. *Applied Intelligence*, 48:4730–4742.

Fabio Poletto, Valerio Basile, Manuela Sanguinetti, Cristina Bosco, and Viviana Patti. 2021. Resources and Benchmark Corpora for Hate Speech Detection: A Systematic Review. *Language Resources and Evaluation*, 55:477–523.

David MW Powers. 2015. What the F-measure doesn't Measure: Features, Flaws, Fallacies and Fixes. *CoRR*, abs/1503.06410.

Rahul Pradhan, Ankur Chaturvedi, Aprna Tripathi, and Dilip Kumar Sharma. 2020. A Review on Offensive Language Detection. *Advances in Data and Information Sciences: Proceedings of ICDIS 2019*, pages 433–439.

Michal Ptaszynski, Agata Pieciukiewicz, and Paweł Dybała. 2019. Results of the PolEval 2019 Shared Task 6: First Dataset and Open Shared Task for Automatic Cyberbullying Detection in Polish Twitter. *Warszawa: Institute of Computer Sciences. Polish Academy of Sciences.*

TTA Putri, S Sriadhi, RD Sari, R Rahmadani, and HD Hutahaean. 2020. A Comparison of Classification Algorithms for Hate Speech Detection. In *Iop conference series: Materials science and engineering*, volume 830, page 032006. IOP Publishing.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving Language Understanding by Generative Pre-Training.

Santhosh Rajamanickam, Pushkar Mishra, Helen Yannakoudakis, and Ekaterina Shutova. 2020. Joint Modelling of Emotion and Abusive Language Detection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4270–4279, Online. Association for Computational Linguistics.

Tharindu Ranasinghe and Marcos Zampieri. 2020a. Multilingual Offensive Language Identification with Cross-lingual Embeddings. pages 5838–5844.

Tharindu Ranasinghe and Marcos Zampieri. 2020b. Multilingual Offensive Language Identification with Cross-lingual Embeddings. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5838–5844, Online. Association for Computational Linguistics.

Tharindu Ranasinghe and Marcos Zampieri. 2021. Multilingual Offensive Language Identification for Low-Resource Languages. *Transactions on Asian and Low-Resource Language Information Processing*, 21(1):1–13.

Francisco Rangel, Gretel Liz de la Peña-Sarracén, María Alberta Chulvi-Ferriols, Elisabetta Fersini, and Paolo Rosso. 2021. Profiling Hate Speech Spreaders on Twitter Task at PAN 2021. In *Proceedings of the Working Notes of CLEF 2021, Conference and Labs of the Evaluation Forum, Bucharest, Romania, September 21st to 24th, 2021*, pages 1772–1789. CEUR.

Julian Risch, Robin Ruff, and Ralf Krestel. 2020. Offensive Language Detection Explained. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pages 137–143, Marseille, France. European Language Resources Association (ELRA).

Julian Risch, Anke Stoll, Lena Wilms, and Michael Wiegand. 2021. Overview of the GermEval 2021 Shared Task on The Identification of Toxic, Engaging, and Fact-claiming Comments. In *Proceedings of the GermEval 2021 Shared Task on the Identification of Toxic, Engaging, and Fact-Claiming Comments*, pages 1–12.

Giulia Rizzi, Francesca Gasparini, Aurora Saibene, Paolo Rosso, and Elisabetta Fersini. 2023. Recognizing Misogynous Memes: Biased Models and Tricky Archetypes. *Information Processing & Management*, 60(5):103474.

Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at TREC-3. *Nist Special Publication Sp*, 109:109.

Sebastián E Rodríguez, Héctor Allende-Cid, and Héctor Allende. 2021. Detecting Hate Speech in Cross-lingual and Multi-lingual Settings Using Language Agnostic Representations. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications: 25th Iberoamerican Congress, CIARP 2021, Porto, Portugal, May 10–13, 2021, Revised Selected Papers 25*, pages 77–87. Springer.

Hugo Rosa, Nádia Pereira, Ricardo Ribeiro, Paula Costa Ferreira, Joao Paulo Carvalho, Sofia Oliveira, Luísa Coheur, Paula Paulino, AM Veiga Simão, and Isabel Trancoso. 2019. Automatic Cyberbullying Detection: A Systematic Review. *Computers in Human Behavior*, 93:333–345.

Andy Rosenbaum, Saleh Soltan, Wael Hamza, Marco Damonte, Isabel Groves, and Amir Saffari. 2022. CLASP: Few-Shot Cross-Lingual Data Augmentation for Semantic Parsing. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 444–462, Online only. Association for Computational Linguistics.

Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Marcos Zampieri, and Preslav Nakov. 2021. SOLID: A large-scale semi-supervised dataset for offensive language identification. ACL/IJCNLP 2021:915–928.

Dhruva Sahrawat, Debanjan Mahata, Mayank Kulkarni, Haimin Zhang, Rakesh Gosangi, Amanda Stent, Agniv Sharma, Yaman Kumar, Rajiv Ratn Shah, and Roger Zimmermann. 2020. Keyphrase Extraction from Scholarly Articles as Sequence Labeling using Contextualized Embeddings. pages 328–335.

Arum Sucia Saksesi, Muhammad Nasrun, and Casi Setianingsih. 2018. Analysis Text of Hate Speech Detection using Recurrent Neural Network. In *2018 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC)*, pages 242–248. IEEE.

Niloofar Safi Samghabadi, Parth Patwa, PYKL Srinivas, Prerana Mukherjee, Amitava Das, and Thamar Solorio. 2020. Aggression and Misogyny Detection Using BERT: A Multi-task Approach. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pages 126–131.

Javier Sánchez-Junquera, Berta Chulvi, Paolo Rosso, and Simone Paolo Ponzetto. 2021a. How do you Speak about Immigrants? Taxonomy and Stereoimmigrants Dataset for Identifying Stereotypes about Immigrants. *Applied Sciences*, 11(8):3610.

Javier Sánchez-Junquera, Paolo Rosso, Manuel Montes, Berta Chulvi, et al. 2021b. Masking and BERT-based Models for Stereotype Identication. *Procesamiento del Lenguaje Natural (SEPLN)*, 67:83–94.

Anna Schmidt and Michael Wiegand. 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*, pages 1–10.

Fabian David Schmidt, Ivan Vulić, and Goran Glavaš. 2022. Don't Stop Fine-Tuning: On Training Regimes for Few-Shot Cross-Lingual Transfer with Multilingual Language Models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10725–10742,

Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Philip Sedgwick. 2012. Pearson's Correlation Coefficient. *Bmj*, 345.

Nina Sevani, Iwan A Soenandi, Jeremy Wijaya, et al. 2021. Detection of Hate Speech by Employing Support Vector Machine with Word2Vec Model. In *2021 7th International Conference on Electrical, Electronics and Information Engineering (ICEEIE)*, pages 1–5. IEEE.

Axler Sheldon, Bourdon Paul, and Ramey Wade. 2001. Harmonic Function Theory. *Graduate Texts in Mathematics*, 137.

Xiayang Shi, Xinyi Liu, Chun Xu, Yuanyuan Huang, Fang Chen, and Shaolin Zhu. 2022. Cross-lingual offensive speech identification with transfer learning for low-resource languages. *Computers and Electrical Engineering*, 101:108005.

Yukai Shi, Sen Zhang, Chenxing Zhou, Xiaodan Liang, Xiaojun Yang, and Liang Lin. 2021a. GTAE: Graph Transformer–Based Auto-Encoders for Linguistic-Constrained Text Style Transfer. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 12(3):1–16.

Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjing Wang, and Yu Sun. 2021b. Masked Label Prediction: Unified Message Passing Model for Semi-Supervised Classification. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 1548–1554. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Yunsheng Shi, Zhengjie Huang, Wenjin Wang, Hui Zhong, Shikun Feng, and Yu Sun. 2021c. Masked Label Prediction: Unified Message Passing Model for Semi-supervised Classification. pages 1548–1554.

Wesam Shishah and Ricky Maulana Fajri. 2022. Large Comparative Study of Recent Computational Approach in Automatic Hate Speech Detection. *TEM Journal*, 11(1):82.

Oleh Shliazhko, Alena Fenogenova, Maria Tikhonova, Vladislav Mikhailov, Anastasia Kozlova, and Tatiana Shavrina. 2022. MGPT: Few-Shot Learners Go Multilingual. *CoRR*, abs/2204.07580.

Elena Shushkevich and John Cardiff. 2019. Automatic Misogyny Detection in Social Media: A Survey. *Computación y Sistemas*, 23(4):1159–1164.

Gudbjartur Ingi Sigurbergsson and Leon Derczynski. 2020. Offensive Language and Hate Speech Detection for Danish. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3498–3508, Marseille, France. European Language Resources Association.

Marco Siino, Elisa Di Nuovo, Ilenia Tinnirello, Marco La Cascia, et al. 2021. Detection of Hate Speech Spreaders using Convolutional Neural Networks. In *CLEF (Working Notes)*, pages 2126–2136.

Sumer Singh and Sheng Li. 2021. Exploiting Auxiliary Data for Offensive Language Detection with Bidirectional Transformers. In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, pages 1–5, Online. Association for Computational Linguistics.

Levent Soykan, Cihan Karsak, Ilknur Durgar Elkahlout, and Burak Aytan. 2022. A Comparison of Machine Learning Techniques for Turkish Profanity Detection. In *Proceedings of the Second International Workshop on Resources and Techniques for User Information in Abusive Language Analysis*, pages 16–24, Marseille, France. European Language Resources Association.

Lukas Stappen, Fabian Brunn, and Björn Schuller. 2020a. Cross-lingual Zero- and Few-shot Hate Speech Detection Utilising Frozen Transformer Language Models and AXEL. *CoRR*, abs/2004.13850.

Lukas Stappen, Fabian Brunn, and Björn Schuller. 2020b. Cross-lingual Zero-and Few-shot Hate Speech Detection Utilising Frozen Transformer Language Models and AXEL. *CoRR*, abs/2004.13850.

Lichao Sun, Congying Xia, Wenpeng Yin, Tingting Liang, Philip Yu, and Lifang He. 2020. Mixup-Transformer: Dynamic Data Augmentation for NLP Tasks. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3436–3440, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Steve Durairaj Swamy, Anupam Jamatia, and Björn Gambäck. 2019. Studying Generalisability across Abusive Language Detection Datasets. In *Proceedings of the 23rd conference on computational natural language learning (CoNLL)*, pages 940–950.

Harrison Uglow, Martin Zlocha, and Szymon Zmyślony. 2019. An Exploration of State-of-the-Art Methods for Offensive Language Detection. *CoRR*, abs/1903.07445.

Neeraj Vashistha and Arkaitz Zubiaga. 2021. Online Multilingual Hate Speech Detection: Experimenting with Hindi and English Social Media. *Information*, 12(1).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, ukasz Kaiser, and Illia Polosukhin. 2017a. Attention is All You Need. In *Advances in neural information processing systems*, pages 5998–6008.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017b. Attention is All you Need. 30.

Abhishek Velankar, Hrushikesh Patil, and Raviraj Joshi. 2022. A Review of Challenges in Machine Learning based Automated Hate Speech Detection. *arXiv preprint arXiv:2209.05294*.

Bertie Vidgen and Leon Derczynski. 2020. Directions in Abusive Language Training Data, A Systematic Review: Garbage in, Garbage out. *Plos one*, 15(12):e0243300.

Bertie Vidgen, Tristan Thrush, Zeerak Waseem, and Douwe Kiela. 2021. Learning from the Worst: Dynamically Generated Datasets to Improve Online Hate Detection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1667–1682. Association for Computational Linguistics.

Michael S Waltman and Ashely A Mattheis. 2017. Understanding Hate Speech. In *Oxford research encyclopedia of communication*.

JianYuan Wang, KeXin Liu, YuCheng Zhang, Biao Leng, and JinHu Lu. 2023. Recent Advances of Few-Shot Learning Methods and Applications. *Science China Technological Sciences*, pages 1–25.

Shuohuan Wang, Jiaxiang Liu, Xuan Ouyang, and Yu Sun. 2020a. Galileo at SemEval-2020 Task 12: Multi-lingual Learning for Offensive Language Identification using Pre-trained Language Models. pages 1448–1455.

Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. 2020b. Generalizing from a Few Examples: A Survey on Few-shot Learning. *ACM computing surveys (csur)*, 53(3):1–34.

Abid Hussain Wani, Nahida Shafi Molvi, and Sheikh Ishrah Ashraf. 2019. Detection of Hate and Offensive Speech in Text. In *International Conference on Intelligent Human Computer Interaction*, pages 87–93. Springer.

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. Association for Computational Linguistics.

Duncan J Watts and Steven H Strogatz. 1998. Collective Dynamics of Small-World Networks. *nature*, 393(6684):440–442.

Jason Wei and Kai Zou. 2019. EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. In *Proceedings of*

the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Gregor Wiedemann, Seid Muhie Yimam, and Chris Biemann. 2020. UHH-LT at SemEval-2020 Task 12: Fine-tuning of Pre-trained Transformer Networks for Offensive Language Detection. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1638–1644.

Michael Wiegand, Josef Ruppenhofer, and Thomas Kleinbauer. 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)*, pages 602–608.

Genta Winata, Shijie Wu, Mayank Kulkarni, Thamar Solorio, and Daniel Preotiuc-Pietro. 2022. Cross-lingual Few-Shot Learning on Unseen Languages. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 777–791, Online only. Association for Computational Linguistics.

Ian H Witten, Gordon W Paynter, Eibe Frank, Carl Gutwin, and Craig G Nevill-Manning. 2005. KEA: Practical Automated Keyphrase Extraction. In *Design and Usability of Digital Libraries: Case Studies in the Asia Pacific*, pages 129–152. IGI global.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. 2021. A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24.

Mengzhou Xia, Anjalie Field, and Yulia Tsvetkov. 2020. Demoting Racial Bias in Hate Speech Detection. In *Proceedings of the Eighth International Workshop on Natural Language Processing for Social Media*, pages 7–14, Online. Association for Computational Linguistics.

Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. 2018. Representation Learning on Graphs with Jumping Knowledge Networks. In *International Conference on Machine Learning*, pages 5453–5462. PMLR.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized Autoregressive Pretraining

for Language Understanding. *Advances in neural information processing systems*, 32.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019a. Graph Convolutional Networks for Text Classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 7370–7377.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019b. Graph Convolutional Networks for Text Classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7370–7377.

Mahsa Yarmohammadi, Xutai Ma, Sorami Hisamoto, Muhammad Rahman, Yiming Wang, Hainan Xu, Daniel Povey, Philipp Koehn, and Kevin Duh. 2019. Robust Document Representations for Cross-Lingual Information Retrieval in Low-Resource Settings. In *Proceedings of Machine Translation Summit XVII: Research Track*, pages 12–20, Dublin, Ireland. European Association for Machine Translation.

Wenjie Yin and Arkaitz Zubiaga. 2021. Towards Generalisable Hate Speech Detection: A Review on Obstacles and Solutions. *PeerJ Computer Science*, 7:e598.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019a. Predicting the Type and Target of Offensive Posts in Social Media. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 1415–1420.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019b. Predicting the Type and Target of Offensive Posts in Social Media. pages 1415–1420.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019c. Semeval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media (Offenseval). pages 75–86.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. 2020a. SemEval-2020 Task 12: Multilingual Offensive Language Identification in Social Media (OffensEval 2020). pages 1425–1447.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. 2020b. SemEval-2020 Task 12: Multilingual Offensive Language Identification in Social Media (OffensEval 2020). pages 1425–1447.

Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. 2018. MIXUP: Beyond Empirical Risk Minimization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Xin Zhang, Miao Jiang, Honghui Chen, Chonghao Chen, and Jianming Zheng. 2022. Cloze-Style Data Augmentation for Few-Shot Intent Recognition. *Mathematics*, 10(18):3358.

Mengjie Zhao, Yi Zhu, Ehsan Shareghi, Ivan Vulić, Roi Reichart, Anna Korhonen, and Hinrich Schütze. 2021. A Closer Look at Few-Shot Crosslingual Transfer: The Choice of Shots Matters. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5751–5767, Online. Association for Computational Linguistics.

Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2020. Graph Neural Networks: A Review of Methods and Applications. *AI Open*, 1:57–81.

Jing Zhou, Yanan Zheng, Jie Tang, Li Jian, and Zhilin Yang. 2022. FlipDA: Effective and Robust Data Augmentation for Few-Shot Learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8646–8665, Dublin, Ireland. Association for Computational Linguistics.

Haris Bin Zia, Ignacio Castro, Arkaitz Zubiaga, and Gareth Tyson. 2022. Improving Zero-Shot Cross-Lingual Hate Speech Detection with Pseudo-Label Fine-Tuning of Transformer Language Models. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 16, pages 1435–1439.