



UNIVERSITÀ DEGLI STUDI  
DI TRENTO

---

DEPARTMENT OF INFORMATION ENGINEERING AND COMPUTER SCIENCE  
**ICT International Doctoral School**

# RISE AND PITFALLS OF SYNTHETIC DATA FOR ABUSIVE LANGUAGE DETECTION

**Camilla Casula**

**Advisor:**

Sara Tonelli

Fondazione Bruno Kessler

---

October 2024



# Abstract

*Synthetic data has been proposed as a method to potentially mitigate a number of issues with existing models and datasets for abusive language detection online, such as negative psychological impact on annotators, privacy issues, dataset obsolescence and representation bias. However, previous work on the topic has mostly focused on downstream task performance of models, without paying much attention to the evaluation of other aspects.*

*In this thesis, we carry out a series of experiments and analyses on synthetic data for abusive language detection going beyond performance, with the goal of assessing both the potential and the pitfalls of synthetic data from a qualitative point of view. More specifically, we study synthetic data for abusive language detection in English focusing on four aspects: robustness, examining the ability of models trained on synthetic data to generalize to out-of-distribution scenarios; fairness, with an exploration of the representation of identity groups; privacy, exploring the use of entirely synthetic datasets to avoid sharing user-generated data; and finally we consider the quality of the synthetic data, through a manual annotation and analysis of how realistic and representative of real data synthetic data can be with regards to abusive language.*

## **Keywords**

Abusive language detection, Synthetic data, Data augmentation



*To my dad, Dario Casula,  
who gave his all for me to get  
chances he was never given.*



# Acknowledgements

First and foremost, I would like to express my undying gratitude to my supervisor, Sara Tonelli, for being the best mentor and advisor I could possibly ask for. I am also grateful to the examination committee: Debora Nozza, Tommaso Caselli, and Mauro Dragoni, in particular to the referees for this work, who took the time to review it, suggest improvements, and raise important questions.

I would like to thank my colleagues and friends at FBK, in particular the DH unit. I am especially thankful to my seniors Alan, Alessio, Elisa, Marco, and Stefano, who took on the role of mentoring and guiding me along this journey. A thank you also goes to Marco G., who years ago planted the initial idea that eventually evolved into my PhD topic.

I am very thankful to the University of Groningen CL group, who welcomed me as one of their own when I visited their lab.

An enormous thank you goes to my friends, both close and far away, and to my family, whose support has been constant and fundamental.

I am eternally grateful to my partner Marco for, well, everything. A list of the ways in which he has supported me would be taller than he is.

Finally, I cannot describe my gratitude towards my parents, Dario and Lorena, who sacrificed so much to grant me the luxury of choosing what I wanted to do with my life. To my dad, whose love and pride fuel my resolve like nothing else, even now that his voice is a memory. And to my mom, who is and always will be my number one role model.





# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Research Questions . . . . .	4
1.2	Contributions . . . . .	5
1.3	Context . . . . .	6
1.4	Structure of the Thesis . . . . .	7
<b>2</b>	<b>Background</b>	<b>9</b>
2.1	Abusive Language . . . . .	9
2.1.1	Terminology . . . . .	10
2.1.2	Detecting Abuse . . . . .	12
2.1.3	Issues . . . . .	15
2.2	Synthetic Data . . . . .	18
2.2.1	Data Augmentation and Generation . . . . .	19
2.2.2	Large Language Models . . . . .	20
2.2.3	Synthetic Data for Abusive Language Detection . . . . .	22
<b>3</b>	<b>Datasets</b>	<b>25</b>
3.1	Founta et al. . . . .	25
3.2	Offensive Language Identification Dataset . . . . .	26
3.3	Social Bias Inference Corpus . . . . .	27
3.4	Measuring Hate Speech Corpus . . . . .	27
3.5	Multi-Domain Agreement . . . . .	29

3.6	HateCheck . . . . .	30
3.7	Data Splits and Preprocessing . . . . .	31
<b>4</b>	<b>Robustness</b>	<b>33</b>
4.1	Methods . . . . .	35
4.2	Experimental Setting . . . . .	38
4.2.1	Number of Training Instances . . . . .	39
4.2.2	Prompting . . . . .	40
4.2.3	Classifier Filtering Thresholds . . . . .	42
4.2.4	Baselines . . . . .	43
4.3	Results . . . . .	44
4.4	Qualitative Analysis . . . . .	51
4.4.1	Examples of Generated Texts . . . . .	51
4.4.2	Lexical Analysis . . . . .	54
4.4.3	HateCheck Analysis . . . . .	57
4.5	Conclusions on Robustness . . . . .	62
<b>5</b>	<b>Fairness</b>	<b>63</b>
5.1	Fairness Experiments Data . . . . .	64
5.2	Methodology . . . . .	66
5.2.1	Generative Models . . . . .	67
5.2.2	Finetuning, Few-Shot Prompting, and Identity Group Information . . . . .	68
5.3	Experimental Setup . . . . .	71
5.3.1	Implementation details with Generative DA . . . . .	72
5.3.2	System comparison . . . . .	73
5.4	Results and Discussion . . . . .	74
5.4.1	Generative DA . . . . .	74
5.4.2	Mixture of Generative DA and EDA . . . . .	76
5.5	Qualitative Analysis . . . . .	77

5.5.1	Manual Annotation . . . . .	79
5.5.2	HateCheck Analysis . . . . .	81
5.6	Conclusions on Fairness . . . . .	84
<b>6</b>	<b>Privacy</b>	<b>87</b>
6.1	Methods . . . . .	89
6.1.1	Rewriting Original Examples . . . . .	90
6.1.2	Filtering . . . . .	92
6.2	Evaluation . . . . .	93
6.2.1	Classification Results . . . . .	94
6.2.2	Qualitative Analysis . . . . .	95
6.3	Conclusions on Privacy . . . . .	98
<b>7</b>	<b>Realism and Quality</b>	<b>101</b>
7.1	Synthetic Data Generation . . . . .	102
7.1.1	Prompting . . . . .	104
7.1.2	Filtering . . . . .	104
7.2	Extrinsic Evaluation . . . . .	105
7.3	Intrinsic Evaluation . . . . .	110
7.3.1	Realism of Synthetic Texts . . . . .	112
7.3.2	Redistribution of Hateful Texts . . . . .	114
7.3.3	Redistribution of Target Identities . . . . .	117
7.4	Conclusions on Realism and Quality . . . . .	122
<b>8</b>	<b>Conclusions</b>	<b>125</b>
	<b>Bibliography</b>	<b>129</b>
<b>A</b>	<b>Variationist</b>	<b>155</b>
A.1	Tool Design . . . . .	156
A.1.1	Design Principles . . . . .	156

A.1.2	Core Elements and Functionalities . . . . .	157
A.2	Implementation and Usage . . . . .	161
A.2.1	User-facing Classes . . . . .	161
A.2.2	Data Interchange . . . . .	163
A.2.3	Example Usage . . . . .	163
A.3	Conclusion . . . . .	163
<b>B</b>	<b>Annotation Guidelines (Chapter 7)</b>	<b>165</b>

# List of Tables

- 4.1 Average macro- $F_1$  scores (over 10 runs) obtained by RoBERTa-base fine-tuned on augmented data, starting with 500 gold examples.  $F_1$  scores for the minority class are in parentheses. Grey cells contain within-dataset results, while the others contain cross-dataset results. Asterisks denote statistically significant results (compared to no augmentation). The best result for each train-test dataset combination is in bold. . . . . 45
  
- 4.2 Average macro- $F_1$  scores (over 10 runs) obtained by RoBERTa-base fine-tuned on augmented data, starting with 2,000 gold examples.  $F_1$  scores for the minority class are in parentheses. Grey cells contain within-dataset results, while the others contain cross-dataset results. Asterisks denote statistically significant results (compared to no augmentation). The best result for each train-test dataset combination is in bold. . . . . 46

4.3	Average macro-F <sub>1</sub> scores (over 10 runs) obtained by BERT-base-uncased fine-tuned on augmented data, starting with 500 gold examples. F <sub>1</sub> scores for the minority class are in parentheses. Grey cells contain within-dataset results, while the others contain cross-dataset results. Asterisks denote statistically significant results (compared to no augmentation). The best result for each train-test dataset combination is in bold. . . . .	47
4.4	Average macro-F <sub>1</sub> scores (over 10 runs) obtained by BERT-base-uncased fine-tuned on augmented data, starting with 2,000 gold examples. F <sub>1</sub> scores for the minority class are in parentheses. Grey cells contain within-dataset results, while the others contain cross-dataset results. Asterisks denote statistically significant results (compared to no augmentation). The best result for each train-test dataset combination is in bold. . . . .	48
4.5	Examples of sequences generated by GPT-2 large models trained on 500 gold examples for every combination of prompting and dataset. . . . .	52
4.6	Examples of sequences generated by GPT-2 large models trained on 2,000 gold examples for every combination of prompting and dataset. . . . .	53
4.7	Top tokens for the <i>offensive</i> class in the gold data and in the generated data when starting with 500 examples, computed using the PMI implementation of Ramponi and Tonelli (2022). The indices refer to the ranking of importance of the tokens in the gold data, while the order of the tokens reflect their informativeness for the offensive class in the generated data. . . . .	55

4.8	Top tokens for the <i>offensive</i> class in the gold data and in the generated data when starting with 2,000 examples, computed using the PMI implementation of Ramponi and Tonelli (2022). The indices refer to the ranking of importance of the tokens in the gold data, while the order of the tokens reflect their informativeness for the offensive class in the generated data. . . . .	56
4.9	Accuracy on the first 25 functional HateCheck tests of RoBERTa models trained on 500 gold examples and on the augmented data. . . . .	60
4.10	Accuracy on the different types of targets in the HateCheck tests of RoBERTa models trained on 500 gold examples and on the augmented data. . . . .	60
4.11	Accuracy on the first 25 functional HateCheck tests of RoBERTa models trained on 2,000 gold examples and on the augmented data. . . . .	61
4.12	Accuracy on the different types of targets in the HateCheck tests of RoBERTa models trained on 2,000 gold examples and on the augmented data. . . . .	61
5.1	Examples from the dataset after our aggregation operations, with a hate speech label and a list of target identity groups mentioned or referred to in the text. . . . .	66
5.2	Templates used for fine-tuning and prompting generative models during the generation step. . . . .	70

5.3	DeBERTa results (macro- $F_1$ and hate-class $F_1$ ) with generative DA, averaged over 5 runs $\pm stdev$ , overall and by target ( <i>Gender, Race, Origin, Sexuality, Religion, Disability, and Age</i> ). Statistical significance is calculated against the <i>no augmentation</i> baseline. $\star$ : highly statistically significant ( $\tau = 0.2$ ), $\diamond$ : statistically significant ( $\tau = 0.5$ ). $n(h)$ = number of <i>hateful</i> synthetic examples preserved after filtering. .	74
5.4	DeBERTa results of generative DA + EDA overall and by target, averaged over 5 runs $\pm stdev$ . Statistical significance is calculated against the results obtained with EDA. $\diamond$ : statistically significant against EDA alone ( $\tau = 0.5$ ). . . . .	78
5.5	Generated texts labeled as correct by human annotators in terms of labels, target categories, and realism. N/A refers to cases in which all of the generated texts were nonsensical (0% realistic), with impossible assignment of labels or categories. We also report the model performance from Table 5.3 in terms of Macro- $F_1$ and Hate $F_1$ , in order to make comparisons between model performance and manual annotation results easier. . . . .	80
5.6	DeBERTa results on HateCheck (hate- $F_1$ ) by target identity, averaged across 5 runs. $p.$ is an abbreviation for <i>people</i> . Statistical significance is calculated against the results obtained with EDA. $\diamond$ : statistically significant ( $\tau = 0.5$ ). . .	82



6.1	Average results over 5 runs in terms of Macro- $F_1$ ( $M-F_1$ ) and Abusive-class $F_1$ ( $Ab-F_1$ ) $\pm$ stdev. Grey cells denote out of distribution / cross-dataset performance. The $n(train)$ column indicates the number of initial examples for gold data and the number of synthetic instances that passed filtering and are therefore used for training. . . . .	95
6.2	Lexical diversity measures on the original and synthetic data for both datasets. . . . .	96
6.3	Top 10 most relevant tokens for the abusive class in the original gold data and in the synthetic data for both datasets, as calculated using the <code>npw_relevance</code> metric in Variationist. 97	97
7.1	Results of Roberta Large models trained on synthetic data only (average of 5 runs $\pm$ stdev). Grey cells indicate out-of-distribution performance. <i>Filter:No</i> means that only paraphrased sequences too similar to the original ones and ill-formatted texts were discarded. <i>Filter:Yes</i> means that <i>classifier filtering</i> was applied. . . . .	107
7.2	Results of Roberta Base models trained on synthetic data only (average of 5 runs $\pm$ stdev). Grey cells indicate out-of-distribution performance. <i>Filter:No</i> means that only paraphrased sequences too similar to the original ones and ill-formatted texts were discarded. <i>Filter:Yes</i> means that <i>classifier filtering</i> was applied. . . . .	108

7.3	Results of DeBERTa Base models trained on synthetic data only (average of 5 runs $\pm$ stdev). Grey cells indicate out-of-distribution performance. <i>Filter:No</i> means that only paraphrased sequences too similar to the original ones and ill-formatted texts were discarded. <i>Filter:Yes</i> means that <i>classifier filtering</i> was applied. . . . .	108
7.4	Synthetic text realism annotations. . . . .	114
7.5	Top- $k = 10$ most informative tokens for the <i>hateful</i> class, according to the PMI metric across targets of hate in GOLD and SYNTHETIC posts paraphrased using Llama-2 Chat 7B, Mistral 7B Instruct, and Mixtral 8x7B Instruct. . . . .	121

# List of Figures

- 4.1 The generative DA pipeline we follow in our *robustness* experiments, from gold data to filtered syntetic data. . . . . 36
- 4.2 Summary of the types of prompts we use for training models in our *robustness* experiments. In this example, the **label** is *not offensive*, and the **text** is *You go girl!*. . . . . 40
- 5.1 Identity group distribution in the MHS corpus. . . . . 65
- 5.2 The generative DA pipeline, from gold data to filtered syntetic data. . . . . 67
- 6.1 Original example and its corresponding synthetic rewriting. 88
- 7.1 Distribution of hateful and non hateful texts in the subset of gold and synthetic data created using the Llama 2 Chat 7B model. . . . . 115
- 7.2 Distribution of hateful and non hateful texts in the subset of gold and synthetic data created using the Mistral 7B Instruct model. . . . . 115
- 7.3 Distribution of hateful and non hateful texts in the manually labeled subset of gold and synthetic data created using the Mixtral 8x7B Instruct model. . . . . 116
- 7.4 Target identity redistribution in synthetic texts created with Llama 2 Chat 7B. . . . . 118

7.5	Target identity redistribution in synthetic texts created with Mistral 7B Instruct. . . . .	119
7.6	Target identity redistribution with the Mixtral 8x7B Instruct model. . . . .	120
A.1	Example showcasing the four steps for inspecting data and visualizing results using VARIATIONIST. . . . .	161

# Chapter 1

## Introduction

In 2022, in Kenya, Meta was sued by an employee of a content moderation outsourcing firm, claiming extremely low pay and widespread trauma among the company's content moderators, who suffered severe post-traumatic stress disorder after being exposed to graphic and violent content as part of their job (Perrigo, 2022). In 2023 another tech giant, ByteDance, was accused of failing to protect the mental health of its content moderators in Kenya, among other allegations (Perrigo, 2023). Further accounts report poor working conditions and mental health risks for content moderators in Colombia, Brazil, and other countries as well (McIntyre et al., 2022; McIntyre, 2023). In general, tech companies typically employ content moderators (often from the Global South) through outsourcing to moderate content that is consumed across the globe, for a market that is currently estimated at a minimum of 8 billion US dollars, although these workers often do not earn acceptable wages (Jackson, 2024; Graham and Ferrari, 2022). These workers are employed not only to directly review and act on harmful content on social media, but also to produce new data that will then be used to train machine learning (ML) models for automatic content moderation (Jackson, 2024).

Mental health risks for annotators are among the reasons why the

use of synthetic data has been proposed for training machine learning models to detect abusive content (Juuti et al., 2020), especially in light of the increasingly realistic texts large language models (LLMs) are able to generate (Yang et al., 2024).

A large portion of online content moderation tools and procedures are proprietary to specific platforms, as in many cases the responsibility to remove potentially harmful content falls upon them from a legal standpoint.<sup>1</sup> This results in the content moderation pipelines that are actually employed on platforms being largely opaque to the general public (Roberts, 2019; Graham and Ferrari, 2022). While the decision-making processes of social media platforms are mostly locked behind closed corporate doors, a large body of research on the topic of abusive and offensive content detection has emerged in recent years. In this work, we will focus on textual content, although efforts have been also made regarding audio, image, video, and multimodal content as well (Gomez et al., 2020; Ibañez et al., 2021, among others).

The field of abusive content detection has gained a large amount of traction in recent years. In particular, since 2018, there has been an increased effort from the research community in establishing common guidelines and benchmarks (e.g. Basile et al. (2019); Zampieri et al. (2019b, 2020)). More recently, there has been an expanding body of work proposing the use of synthetic data to mitigate some of the known issues with data and models typically used for this task (Wullach et al., 2021; D’Sa et al., 2021; Hartvigsen et al., 2022, among others).

In this thesis, we will focus on the use of synthetic data for detecting abusive language from a two-fold perspective. First, detailing how over the past few years the use of synthetic data has been popularized

---

<sup>1</sup>For instance, in the EU, this responsibility falls under the Digital Services Act (European Parliament and Council of the European Union, 2022).

thanks to large language models (LLMs) achieving better performance and realism in generating text, keeping in mind also other aspects that are often neglected in machine learning research, such as fairness or privacy. Second, investigating the pitfalls and qualitative implications of this approach, as well as exploring the possible risks of using synthetic data for this task.

**⚠ Warning:** *this work contains potentially offensive or upsetting examples.*

## 1.1 Research Questions

The aim of this work is that of investigating both the potential and the risks of approaches based on synthetic data for subjective tasks such as abusive language detection. In particular, we are interested in going beyond mere *performance*, which is the main focus of the overwhelming majority of papers in the machine learning field (Birhane et al., 2022). Instead, the main focus of our experiments and analyses will be on aspects that have received much less attention in the past, such as *robustness*, *fairness*, *privacy*, and *realism* (for a discussion of ML values and their representation we refer the reader to Birhane et al. (2022)). We nonetheless also consider performance, in order for our work to be contextualized within the larger body of research on this topic.

The overarching research questions whose thread will be followed along this entire work are related to the two sides of the synthetic data coin, i.e. its potential and its drawbacks:

- **RQ1:** Can we expect synthetic data to be able to improve current abusive language detection models?
  - **RQ1.1** From a *performance* standpoint;
  - **RQ1.2** From a model *robustness* standpoint;
  - **RQ1.3** From a model and data *fairness* standpoint;
- **RQ2:** What are the implications of synthetic data in terms of *quality*?
  - **RQ2.1** Can synthetic texts mitigate *privacy* issues?
  - **RQ2.2** Are synthetic texts *realistic* enough to emulate real data?
  - **RQ2.3** Do any model improvements come at *other costs*?



## 1.2 Contributions

The contributions of this thesis, alongside the main relative publications, are detailed below.

- An in-depth evaluation of advantages and disadvantages of synthetic data for abusive language and hate speech detection, across a variety of models and architectures, from the point of view of *performance* and *robustness*.
  - Camilla Casula and Sara Tonelli. 2023. Generation-Based Data Augmentation for Offensive Language Detection: Is It Worth It?. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3359–3377, Dubrovnik, Croatia. Association for Computational Linguistics.
- An analysis of the implications of synthetic data with regards to the distribution of identity group mentions, and the impact of these changes on *fairness*.
  - Camilla Casula and Sara Tonelli. 2024. A Target-Aware Analysis of Data Augmentation for Hate Speech Detection. *arXiv preprint arXiv:2410.08053*.
- An exploration of the potential of synthetic data to mitigate issues related to *privacy*.
  - Camilla Casula, Elisa Leonardelli, and Sara Tonelli. 2024. Don't Augment, Rewrite? Assessing Abusive Language Detection with Synthetic Data. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 11240–11247, Bangkok, Thai-

land and virtual meeting. Association for Computational Linguistics.

- A manual investigation of the language present in synthetic data, with details on how it might lead models to rely on spurious correlations when trained on it. This type of analysis could serve to future researchers aiming at using synthetic data to know about potential pitfalls they could look out for.
  - Camilla Casula, Sebastiano Vecellio Salto, Alan Ramponi, and Sara Tonelli. Delving into Qualitative Implications of Synthetic Data for Hate Speech Detection. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. To appear. Association for Computational Linguistics.
- A tool to inspect textual datasets in a highly customizable and modular way, which was used across many of the experiments included in this thesis: Variationist.
  - Alan Ramponi\*, Camilla Casula\*, and Stefano Menini. 2024. Variationist: Exploring Multifaceted Variation and Bias in Written Language Data. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 346–354, Bangkok, Thailand. Association for Computational Linguistics.

### 1.3 Context

Before dealing with the specifics of this work, it is important to address some preliminary concepts regarding the speed at which new methods

---

<sup>1\*</sup> Equal contribution.

and models in natural language processing (NLP) evolve and exponentially grow in size (Bender et al., 2021; Zhao et al., 2023), although the details of the language models that are now the state of the art in NLP will be discussed in Section 2.2.2.

In particular, this thesis features experiments carried out over the span of three years using a variety of language models. Some of the models used in the experiments of this work, such as GPT-2 (1.5B parameters, Radford et al. (2019)) were considered *large* in 2021 (Bender et al., 2021), and had already been ‘downgraded’ to the rank of *smaller* pretrained language models by 2023 (Zhao et al., 2023). As a consequence, with the rapid expansion of model sizes and capabilities, comparisons across approaches have become more difficult to carry out, since a method that might work well with one model might not work as well with another one, even a mere few months later. Where possible, the experimental setups across the chapters of this work were kept as consistent as possible. However, readers should be aware that, in addition to being thematically ordered based on the research questions and topics of interest, the experimental chapters should also be considered from a chronological point of view, with earlier chapters featuring ‘older’ models and later chapters being more reflective of the current state of the art with regards to (currently large) language models.

## 1.4 Structure of the Thesis

The chapters of this thesis follow roughly the same thread as the research questions, focusing on one often overlooked aspect of the evaluation of machine learning models at a time. The experiments in each chapter will be focused on one research question mostly, but overall, the answer to a question might come from different sections, and one single section

might help answer more than one question.

**Chapter 2** introduces a series of background concepts that are important for the contextualization and understanding of this work, including a summarization of previous work on abusive language, data augmentation and creation of synthetic data using large language models.

**Chapter 3** provides an overview of the datasets we will use in our experiments in the following chapters.

**Chapter 4** presents an analysis of the impact of data augmentation for abusive language detection with regards to the *robustness* of models, intended as their ability to generalize to out-of-distribution contexts, by training and testing models on four abusive language detection datasets.

**Chapter 5** focuses on the impact of synthetic data on *fairness*, with an exploration of the representation of target identity groups and the possibility of including identity information during the synthetic data creation step.

**Chapter 6** addresses synthetic data for abusive language detection from a *privacy* standpoint, investigating the possibility of entirely substituting abusive language detection datasets with synthetic data for privacy reasons, without having to share real user-generated data.

**Chapter 7** examines synthetic data for the task of abusive language detection from the point of view of *realism* and *quality*, with an in-depth manual analysis of synthetic texts, discussing (and challenging) common assumptions regarding synthetic data.

**Chapter 8** concludes this work by summarizing its contributions, discussing its limitations and potential future directions.

# Chapter 2

## Background

In this chapter, we examine a series of background concepts that are necessary for contextualizing this thesis. First, we discuss some preliminary concepts pertaining to abusive language and hate speech in Section 2.1, including terminological issues (Section 2.1.1), widely used methods for detecting abusive content (Section 2.1.2), and issues that can arise both in datasets and in models used for this task (Section 2.1.3). We then move on to the topic of synthetic data in Section 2.2, examining how synthetic data is typically created in NLP (Section 2.2.1), discussing some fundamentals about large language models (Section 2.2.2), and finally reviewing previous work on synthetic data for hate speech and abusive language detection (Section 2.2.3).

### 2.1 Abusive Language

As communication over the Internet has become for our society as frequent as face-to-face, in-person communication, both the good and the ugly aspects of the spectrum of human interaction have entered online spaces. In particular, as the amount of hateful, harmful, or offensive content posted online has increased dramatically over the last decades, methods to automatically detect, measure and deal with problematic

content in some form have been developed and widely employed.

The terms *abusive language*, *offensive language*, *hate speech*, and *toxic content* have all been used in NLP research work to refer to similar phenomena common on social media (Schmidt and Wiegand, 2017). However, none of these terms has collectively agreed-on definitions, resulting in variety across works and corpora in terms of definitions and annotation guidelines. Similarly, scholars in disciplines such as philosophy and psychology have also struggled to find unified definitions and terminology on the matter (Waqas et al., 2019; Anderson and Barnes, 2023). We will therefore start by discussing the way in which these terms will be used in the present work.

### 2.1.1 Terminology

Most NLP works dealing with abusive language and hate speech provide a definition for the term they use, but even with clear definitions there can be differences with regards to what is considered offensive, toxic, hateful, or abusive. For instance, some might include non-insulting profanities in their definition of offensiveness (Zampieri et al., 2019a), while others might not. In addition to this, in some cases research work aimed at abusive language detection in general actually addresses more specific forms of abusive behavior, such as misogyny or racism (Vidgen et al., 2019).

The main issue with defining abusive language lies in the fact that any set of criteria used to define it cannot be objectively “correct”, since the phenomenon is subjective in nature (Vidgen et al., 2019; Basile, 2020; Vidgen and Derczynski, 2020). The problem of subjectivity, moreover, cannot and arguably should not be ‘circumvented’ by using more *seemingly* universally accepted definitions, such as laws or terms of service. In fact, according to Vidgen et al. (2019), the definition of what makes

a message abusive cannot be reduced to legal definitions or platform guidelines, as the latter is often potentially influenced by financial interests, and both can often be overly generic due to the ways in which these are typically formulated.

A form of abusive language that has received much attention in scholarly work from a number of disciplines is *hate speech*, in part due to its potential legal consequences (Anderson and Barnes, 2023). According to Post (2009), *hate speech* can be defined on the basis of four elements, out of which more than one can co-occur in a single definition:

- The *harms* it will cause, such as discrimination, psychological or physical harms;
- Its *content*, i.e. what it conveys from a semantic point of view;
- Its *intrinsic properties*, i.e. the words it uses, by which account, for instance, all utterances containing slurs would automatically be hate speech (Parvaresh (2023) refers to this as *lexical approach*);
- Its connection to principles of *dignity*, e.g. the potential to undermine the dignity and social standing of its targets.

In this work, we choose to focus on the broader phenomenon of *abusive content*, for which we employ a mixed definition, based on both harms and content: the definition of *abusive language* provided by Caselli et al. (2020).

*Hurtful language that a speaker uses to insult or offend another individual or a group of individuals based on their personal qualities, appearance, social status, opinions, statements, or actions.*

This definition, as stated by its authors, can include *hate speech*, *derogatory language*, *profanity*, and more. In this work we will therefore refer to

*abusive language* as a form of language that possesses these characteristics, sometimes switching between dealing with *abusive language* in general and with more specific forms of abuse, such as *hate speech* (Caselli et al., 2020), depending on the specific analysis of each chapter.<sup>1</sup>

### A Note on Profanities and Obfuscation

There are a multitude of ways in which one can present profanities in scientific publications (Kirk et al., 2022a; Nozza and Hovy, 2023). In an attempt to make the examples featured in this work as clearly understandable and accessible as possible, while avoiding unnecessary potential psychological discomfort to any readers, we choose to only obfuscate terms that are in and of themselves derogatory towards individuals or groups based on some characteristics (i.e., slurs), in accordance with the definition above. Therefore, profanities that by themselves do not necessarily refer to people, such as ‘*fuck*’, will be unobfuscated, while derogatory terms such as ‘*b\*tch*’ will be obfuscated by substituting the first vowel with a star symbol, as seen in Nozza and Hovy (2023).

### 2.1.2 Detecting Abuse

Most works on automatic abusive language identification employ supervised machine learning models. Support Vector Machines (Cortes and Vapnik, 2004) were very popular for the task until 2017 (Schmidt and Wiegand, 2017), while from 2018 on deep-learning models have been the most widely used. More specifically, Transformer-based models (Vaswani et al., 2017) have become the standard for detecting abu-

---

<sup>1</sup>*Hate speech* is defined by the UN Strategy and Plan of Action on Hate Speech as ‘*Any kind of communication in speech, writing or behaviour, that attacks or uses pejorative or discriminatory language with reference to a person or a group on the basis of who they are, in other words, based on their religion, ethnicity, nationality, race, colour, descent, gender or other identity factor. This is often rooted in, and generates, intolerance and hatred, and in certain contexts can be demeaning and divisive*’.



sive messages online, with encoder-only classifiers such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) fine-tuned on annotated data being still a widely popular choice to date (Zampieri et al., 2020; Caselli et al., 2021a; Kirk et al., 2022b). More recently, with the increased zero-shot performances of models, researchers have experimented with zero-shot setups using LLMs (Nozza, 2021; Plaza-del arco et al., 2023), which still appear to perform worse than smaller models fine-tuned on labeled data (Edwards and Camacho-Collados, 2024).

### Datasets

Most labeled datasets for abusive language detection are created starting from Twitter (now X) data, mostly because Twitter data collection APIs were the more easily accessible for a long time compared to other platforms (Vidgen and Derczynski, 2020). Other less widely used sources for data include Facebook (Del Vigna et al., 2017; Kumar et al., 2018), Instagram (Vargas et al., 2021; Parvaresh, 2023), Wikipedia comments (Rawat et al., 2019), Reddit (Sachdeva et al., 2022), Youtube (Ashraf et al., 2021), StormFront (Sap et al., 2020) and Gab (Kennedy et al., 2022).

The most common ways of annotating datasets are crowdsourcing, in which large amounts of workers annotate examples via dedicated platforms, and expert annotation. Although the definition of an *expert* in relation to abusive language is fuzzy, expert annotations are reported to produce higher quality data (Vidgen and Derczynski, 2020). On the other hand, however, crowdsourcing makes annotation more convenient and allows the creation of larger datasets, even though the quality of the annotations might suffer in some cases.

The biggest problem in gathering data to annotate for abusive language is that it is simply (and fortunately, from a societal point of view) rarer than non-abusive language. Founta et al. (2018) estimate the pres-

ence of messages containing abuse overall at a maximum of 3% of all posts on Twitter. Since machine learning models work best when trained on large amounts of data, even the minority class (in this case, the hateful one) should be represented through a significant number of samples in a dataset. For instance, starting from a normal distribution of tweets, it would take over 33,000 posts to be annotated for a system to be trained on only 1,000 hateful posts. In addition to this, with one class composing 3% of the total data and the other the remaining 97%, the class imbalance would make it very difficult for the model to reliably classify the minority class without risking overfitting.

In order to increase the number of potentially abusive tweets in the data sampling stage and make the classes more balanced, researchers in previous works have proposed different methods. The most popular method is keyword sampling, in which a list of words potentially linked to abuse is used to filter the posts before beginning the annotation process (Founta et al., 2018; Zampieri et al., 2019a; Waseem and Hovy, 2016). This sampling method, however, tends to inject biases into the data, which can cause models to associate surface patterns or spurious artifacts with labels (Nozza et al., 2019; Zhou et al., 2021; Ramponi and Tonelli, 2022). For example, a model can learn to associate the mere presence of a specific term or topic with a message being abusive, even in cases in which it should not.

Another method consists in selecting specific pages or social networks which are expected to contain more abusive language than normal. For instance, Del Vigna et al. (2017) and Kumar et al. (2018) select a number of Facebook pages about news, politics, and various topics which are expected to generate discussions among users. Although in principle this form of sampling is seemingly more neutral, it still has the potential to inject bias into models, similarly to keyword sampling (Wiegand et al.,

2019).

To mitigate data sampling bias problems, Founta et al. (2018) instead use a method they name *boosted random sampling*, in which potentially abusive messages are pre-selected using different kinds of heuristics, exploiting sentiment analysis and lexical features. While technically less biased, though, this kind of sampling can still lead to biased models (Zhou et al., 2021).

Besides using various forms of sampling, classes in datasets are often artificially balanced in order to increase the number of abusive or offensive examples. Hateful or abusive messages in published datasets can range between 1% and 100% of all data. With highly imbalanced classes, the risk with machine learning is that the model will overfit the majority class, and metrics like accuracy, precision or recall become less informative because of the imbalance. Because of this, having balanced classes is usually preferred in machine learning. However, in the case of abusive language detection, perfectly balanced classes can imply having models learn that abusive messages are just as frequent as non-abusive ones, leading to a high number of false positives. The compromise most dataset creators have found between the extremely low 1-3% of real distributions and the non-ideal 50% is to have between 20% and 40% of abusive examples in a dataset, with the average of most existing offensive language detection datasets being around 36% (Vidgen and Derczynski, 2020).

### 2.1.3 Issues

A number of challenges and issues with how abusive language detection datasets are created and maintained, as well as with the outputs of abusive language detection models, have arisen over time. In this section, we mention some of the most relevant ones.

### Negative Impact on Annotators

Annotating abusive language can take a toll on content moderators and annotators of abusive content, potentially causing triggering effects, psychological and emotional harm, burnout, desensitization, and more due to the nature of their work (Roberts, 2019; Steiger et al., 2021). While the effects of this are more clear on commercial content moderators, it is still the case for annotators of research-related abusive content datasets (Vidgen and Derczynski, 2020). Synthetic data has been proposed as a potential method to reduce the amount of human-annotated data, and therefore the impact of the task on annotators (Vidgen and Derczynski, 2020; Madukwe et al., 2022), in addition to other methods such as psychologically supporting annotators (Vidgen and Derczynski, 2020).

### Privacy

As previously discussed, most data for abusive language detection comes from user-generated social media data, which can contain personal or sensitive information. Due to this, a large number of datasets are shared as a list of post ID numbers, without the actual text contained in the original post. This means that for most Twitter datasets, which are the majority of all abusive language detection datasets, one needs to retrieve and download each tweet directly from the platform, starting from the tweet ID, which can often be difficult.<sup>2</sup>

In general, synthetic data is deemed easier to share, as its privacy concerns are more limited (Vidgen and Derczynski, 2020; Bayer et al., 2022). In fact, privacy concerns are the main historical motivator behind the use of synthetic data in a variety of disciplines (Jordon et al., 2022; Whitney and Norman, 2024).

---

<sup>2</sup>This process was further complicated by the 2023 changes in the Twitter/X APIs, showing that even changes of ownership of private platforms can impact research on this topic.

### **Dataset Decay and Obsolescence**

Since social media networks typically have internal moderation systems devoted to identifying and removing content that violates platform guidelines, it is fairly frequent that posts annotated as abusive in research datasets are removed from the platform, making them irretrievable after a certain amount of time if they are shared as post IDs as discussed above. Because of this, many datasets tend to decrease in size over time (Vidgen et al., 2019).

Besides being subject to decay, datasets can also become obsolete over time with regards to their content. Online conversation topics and linguistic patterns change rapidly, and it has been shown that annotated datasets used to train models that are then tested on diachronically distant examples lead to worse performance (Florio et al., 2020). In other words, the performance of classifiers trained on data from a specific period of time tends to decrease as time passes.

### **Representation Bias**

Another major issue is related to the representation of minority groups considered as targets, which is rather unbalanced, potentially affecting the robustness and fairness of hate speech detection systems. For example, misogyny and racism have been covered in several datasets (Bhattacharya et al., 2020; Zeinert et al., 2021; Guest et al., 2021; Bosco et al., 2023), while other phenomena and targets have received much less attention in past work, such as religious hate (Ramponi et al., 2022) or hate against LGBTQIA+ people (Chakravarthi et al., 2021; Locatelli et al., 2023), which have only recently started to receive more attention. Furthermore, phenomena such as ageism and ableism have been only marginally addressed, and no specific dataset representing these types

of offenses has been created. This disparity affects in turn system fairness, because offenses against less-represented targets will typically be classified with a lower accuracy, further impacting communities that are already marginalized (Talat et al., 2021).

## 2.2 Synthetic Data

Whitney and Norman (2024) categorize *synthetic data* into two separate categories, based on how derivative the data is with respect to a real-world dataset. *Generated data* refers to an ideally ‘novel’ output<sup>3</sup> that is produced by a generative model, while *augmented data* refers to any real-world data instance that was modified in some way, for instance via perturbations such as synonym replacement or random word deletion (Wei and Zou, 2019).

The two terms have also been used interchangeably in the past in some computational linguistics work. Data *augmentation* as a method started to be employed much earlier than generation-based synthetic data, and in general it refers to a family of approaches aimed at increasing the diversity of existing real-word manually annotated data (which we refer to as *gold data*) without collecting new samples (Feng et al., 2021). On the other hand, synthetic data created through *generation* typically relies on very large language models that can produce new texts based on their inner representations of the data they were trained on, without needing task-specific labeled data, for example through prompting. *Synthetic data* is, in NLP, often used to refer to scenarios in which classifiers are trained *only* on artificially created data, while *data augmentation* (DA) typically entails using both gold and synthetic data, in order

---

<sup>3</sup>Synthetic data can hardly ever be entirely novel, as it is always based on representations of real data (Whitney and Norman, 2024). As aptly put by Seaver (2018), ‘if you cannot see a human in the loop, you just need to look for a bigger loop’.

to improve the performance or generalization of models, although the usage of the two terms is not consistent across works. In the present work, we consider *data generation* to refer to any process that creates new data without relying on previously available labeled data (aside from the generative model’s representation of the data it was trained on), and *data augmentation* any process that, starting from existing data, creates new data that is derived from it.

While DA is widely used to make models more robust across many machine learning applications, especially in computer vision (Perez and Wang, 2017), it has not been as frequently adopted or researched in NLP (Bayer et al., 2022; Pellicer et al., 2023) until recently, with generative LLMs being capable of generating realistic text. Indeed, models trained on synthetic data have in some cases been reported to achieve similar or, in some cases, better performance than models trained on gold data on a variety of NLP applications (Feng et al., 2021; Chen et al., 2023), although with mixed results with regards to subjective tasks (Li et al., 2023).

### 2.2.1 Data Augmentation and Generation

Data augmentation (DA), as previously mentioned, is a process in which the diversity of training data is artificially increased without directly collecting more examples, which can often be helpful to mitigate data scarcity issues in machine learning applications (Chen et al., 2023). There are different methods that can be used for data augmentation, which generally fall into three types (Feng et al., 2021):

- **Rule-based techniques**, based on predetermined transformations, which are easy and quick to implement. However, they often offer only marginal benefits in terms of performance and variety com-

pared with other methods;

- **Interpolation techniques**, based on the interpolation of inputs and labels of multiple real examples. This method, also referred to as Mixed Sample Data Augmentation (MSDA), is inspired by existing techniques in computer vision, and is usually employed when dealing with multi-modal data;
- **Model-based techniques**, which are now widely used and reported to produce higher-quality data, although they are more computationally expensive. Some examples of model-based techniques include backtranslation (in which a sequence is translated from language A to language B and then translated back from language B to language A (Sennrich et al., 2016)), and model-based token replacement using token embeddings. This category also includes DA approaches that are based on generative LLMs, which we refer to as *Generative DA* (Kumar et al., 2020; Hartvigsen et al., 2022).

More recently, following works that found generation-based DA to be promising, a series of studies have also investigated the possibility of *generating* new data in a ‘zero-shot’, scenario, in which no gold data is used to either further train a generative model or as an example to follow (Li et al., 2023), simply relying on the inner representations of the model to create new data. This zero-shot generation setup has, however, been found to be associated with significantly worse performances of models for subjective NLP tasks (Li et al., 2023).

### 2.2.2 Large Language Models

In this work, we employ large language models both for generating text and for classifying it. Since 2017, the most widely used and state of



the art models in NLP have been based on the Transformer architecture (Vaswani et al., 2017). This architecture uses Multi-Head Self-Attention, an attention mechanism that made it possible to reduce training times and costs compared to its predecessors, as well as allowed models to handle dependencies in longer sequences compared to previously state-of-the-art models in NLP.

The Transformer is based on an encode-decoder model architecture, intended initially for sequence-to-sequence tasks (Sutskever et al., 2014). However, it is now commonly used also in a single Transformer layer stack configuration, with different forms of self-attention based on the intended use, such as classification (e.g. BERT (Devlin et al., 2019) and its derivatives), for which only the *encoder* stack is typically employed, or text generation (e.g. GPT (Radford et al., 2018) and its successors), for which only the *decoder* stack can be implemented.

### Text Generation

The decoder components of Transformers models are usually built for auto-regressive generation, in which the probability of each token to be generated is conditioned on the previous tokens occurring in a text sequence. The probability of a given sequence of tokens of length  $n$ ,  $W_{0:n} = \{w_0, w_1, \dots, w_n\}$ , is calculated as

$$P(W_{0:n}) = \prod_{i=1}^n P(w_i | w_{0:i-1}) \quad (2.1)$$

where  $w_0$  is the initial context token, which typically is a model-specific beginning-of-sequence token.

The probability of a sequence is used to generate the output text token by token. In order to choose what token to generate at each time step, several decoding strategies can be used.

Greedy algorithms are based on selecting the token or the sequence with the highest probability, in a deterministic fashion. Meanwhile, beam search consists in saving the top- $n$  candidate tokens for generation at each time step, in which  $n$  is defined as *beam size* or *width* (Vijayakumar et al., 2016).

Since the distribution of word probabilities in text written by humans has a much higher variance than that of text generated using greedy decoding, sampling from the next token probability distribution has been proposed as a solution to increase variety, making generated texts supposedly more natural. However, pure sampling has been shown to produce incoherent text in some cases (Holtzman et al., 2020).

In order to allow for more realistic generated text while preserving coherence, Fan et al. (2018) propose *top- $k$*  sampling, in which a token is sampled from the  $k$  tokens with the highest probability at a given time step instead of from the entire distribution. Using this method, very unlikely tokens are pruned out, and the generated text is less repetitive than text generated using beam search decoding. Similarly, Holtzman et al. (2020) propose *nucleus sampling*, also called *top- $p$*  sampling, in which the head of the distribution is truncated, but instead of sampling from the top  $k$  tokens, the sampling is done over the smallest set of tokens whose probabilities sum up to the parameter  $p$ .

### 2.2.3 Synthetic Data for Abusive Language Detection

The generation of synthetic labeled data, as previously discussed, has also been proposed as a potential solution for some of the issues with abusive language detection, as well as to improve performance. Indeed, synthetic data, although inherently non-authentic, can potentially mimic real behavior and represent many types of abuse if properly created, ideally mitigating many of the issues with current abusive lan-

guage datasets (Vidgen and Derczynski, 2020).

Juuti et al. (2020), for example, perform data augmentation for toxic language classification using different methods, from the simpler over-sampling and word replacement approaches to more complex ones exploiting the GPT-2 (Radford et al., 2019) generative model. They fine-tune GPT-2 on data labeled for toxicity, in order to create a model that will generate toxic text used for data augmentation of existing labeled data. However, since their focus is augmenting existing data, they create synthetic data for the minority class only, so as to ‘fix’ the natural class imbalance of the toxicity detection data. They find that, in very low-resource scenarios, models trained on data augmented using GPT-2 perform better than models trained on the original data alone.

Similarly, Liu et al. (2020) use a conditional variant of GPT-2 based on reinforcement learning, where lexical features for each class are extracted from the entire dataset and then used for generation.

Wullach et al. (2021) and D’Sa et al. (2021) also use GPT-2 to augment synthetic hate speech data, showing that the addition of large amounts of synthetic data helps classification performance when starting from datasets containing thousands of labeled instances.

Fanton et al. (2021), on the other hand, combine GPT-2 and human validation to create counter-narratives covering multiple hate targets, while Nouri (2022) uses GPT-2 to generate training samples containing offensive text for the task of offensive task detection.

More recently, Ocampo et al. (2023) have applied data augmentation to increase the number of instances for the minority class in implicit and subtle examples of hate speech. De la Peña Sarracén et al. (2023) propose a variant of vicinal risk minimization (Chapelle et al., 2000) to generate synthetic samples in the vicinity of the gold examples in a multilingual setting using a multilingual GPT model. Further-

more, Hartvigsen et al. (2022) use manually curated (through a human-in-the-loop process) prompts to generate implicitly hateful sequences with GPT-3 (Brown et al., 2020).

Overall, there does not appear to be a consensus with regards to the approaches that work best when dealing with sythetic data for this task, with different works coming to different conclusions. In particular, there have been very few attempts to explore sythetic data for abusive language detection beyond classification performance on benchmark datasets, which is one of the main aims of this work.

# Chapter 3

## Datasets

In this section, we introduce the datasets we will use in the experiments and analyses of the following chapters, explaining our reasons for using each of them, as well as their main characteristics. Overall, we employ a total of 6 English language datasets annotated for various forms of abusive language, coming from different sources and focusing on different facets of the phenomenon of abusive language online.

### 3.1 Founta et al.

The Founta et al. (2018) dataset<sup>1</sup> is among the most widely used abusive language datasets in the literature, and it has been already employed for generative data augmentation (Wullach et al., 2021; D’Sa et al., 2021). It contains around 100k Twitter posts annotated by crowdsourced workers using four labels: *hateful* (7.5%), *abusive* (11%), *normal* (59%), and *spam* (22.5%). In this schema, the *abusive* class encompasses a series of different phenomena, which the authors decide to merge after two preliminary annotation studies: abusiveness, offensiveness and aggressiveness, due to the fact that both offensiveness and aggressiveness are found to be heavily correlated with abusiveness.

---

<sup>1</sup><https://zenodo.org/record/3678559>

The definitions provided to the annotators of this dataset are:

- **Abusive language:** *Any strongly impolite, rude or hurtful language using profanity, that can show a debasement of someone or something, or show intense emotion;*
- **Hate speech:** *Language used to express hatred towards a targeted individual or group, or is intended to be derogatory, to humiliate, or to insult the members of the group, on the basis of attributes such as race, religion, ethnic origin, sexual orientation, disability, or gender.*

In order to keep a binary classification setup that is consistent with the other datasets we use in our experiments, in all of our experimental setups we discard the messages annotated as *spam*, and we group the *hateful* and *abusive* classes together into one single *abusive* class, following Leonardelli et al. (2021).

## 3.2 Offensive Language Identification Dataset

The Offensive Language Identification Dataset, OLID (Zampieri et al., 2019a),<sup>2</sup> consists of 14,200 Twitter posts annotated for offensive language, with two more fine-grained levels of annotation regarding the target of the offense. In our experiments, we only consider the broader binary level of annotation, for which 33% of the dataset is labeled as *offensive*. The definition of *offensiveness* provided by the creators of this dataset is:

*Posts containing any form of non-acceptable language (profanity) or a targeted offense, which can be veiled or direct. This includes insults, threats, and posts containing profane language or swear words.*

---

<sup>2</sup><https://sites.google.com/site/offensevalsharedtask/olid>

The test set we pair with this dataset is SOLID (Rosenthal et al., 2021), which was used in the OffenseEval 2020 shared task, so the results on this test set are directly comparable with those of models used in the shared task. The SOLID test set follows the same annotation guidelines and process as OLID.

### 3.3 Social Bias Inference Corpus

The Social Bias Inference Corpus, SBIC<sup>3</sup> (Sap et al., 2020), contains 40k posts from Twitter, Reddit, and Stormfront, of which 44.8% are annotated as *offensive*. The Twitter portion of this dataset partially overlaps with the Founta et al. (2018) dataset.

Offensiveness, as stated by the authors, *denotes the overall rudeness, disrespect, or toxicity of a post. We consider whether a post could be considered “offensive to anyone.* While this dataset provides fine-grained annotations on social biases, we only consider the categorical *offensiveness* labels in our experiments, in order to compare it with the other datasets we experiment with. Given that this label can take on three values (*yes*, *no*, and *maybe*), we only keep the binary *yes/no* labels in order to facilitate comparisons with the other datasets, discarding examples annotated as *maybe*.

### 3.4 Measuring Hate Speech Corpus

Measuring Hate Speech (MHS) Corpus (Kennedy et al., 2020; Sachdeva et al., 2022) is a dataset consisting of social media posts in English from Reddit, Twitter, and YouTube. The MHS corpus is annotated according to different levels of hatefulness, from supportive to genocidal speech,

---

<sup>3</sup>[homes.cs.washington.edu/~msap/social-bias-frames/](https://homes.cs.washington.edu/~msap/social-bias-frames/)

offering insight into different aspects of hate speech. In particular, the authors exploit Rasch Measurement Theory to map each example in the dataset to a hate speech score rather than a binary *hateful/non-hateful* label. However, Kennedy et al. (2020) also include a binary “hate speech” label in the questionnaires to be assigned by annotators. They additionally conduct a comparison between the continuous score and the binary hate speech score, finding that while the continuous measure can better capture the extremity of hate speech, the two are moderately correlated. Given the scope of our work, we use the binary labels instead of the continuous hate speech scores in our experiments, in order to frame the task as classification rather than regression and to be able to test our models on out-of-distribution data (see Section 5.5.2).

Given that the MHS dataset, following data perspectivism (Cabitza et al., 2023), is released with disaggregated annotations, we perform some aggregation operations in order to use it for our experiments, resulting in each example having a unique label and set of targets. First, we consider each example to be about or targeting all the identity groups identified by at least half of the annotators who annotated it. For example, if out of 5 annotators 3 annotated the target identity group ‘gender’, we will consider this identity group to be the gold target annotation for that example. Additionally, instead of the hate speech continuous score that is present in the dataset, we use the hatespeech label, which can only take three values (0: *non hateful*, 1: *unclear*, 2: *hateful*). We do this in order to frame the task as classification rather than regression for benchmarking purposes, in line with most of the previous work on hate speech detection, in which the task is treated as a classification task. We binarize the three classes by averaging all the annotations for a given post, mapping it to *hateful* if the average score is higher than 1 and to *non hateful* if



it is lower.<sup>4</sup> After this process, we are left with 35,243 annotated posts, of which 9,046 are annotated as containing hate speech.

### 3.5 Multi-Domain Agreement

The Multi-Domain Agreement (MDA) dataset by Leonardelli et al. (2021)<sup>5</sup> is annotated for both offensive language and agreement level among 5 annotators with regards to the offensiveness of a post. It contains 10,753 tweets dealing with three widely discussed topics on Twitter in 2019 and 2020: the Black Lives Matter movement, the 2020 US elections, and Covid-19. Offensive tweets constitute 31% of the dataset, and *offensiveness* is defined to annotators as:

*Profanity, strongly impolite, rude, violent or vulgar language expressed with angry, fighting or hurtful words in order to insult or debase a targeted individual or group. This language can be derogatory on the basis of attributes such as race, religion, ethnic origin, sexual orientation, disability, or gender. Also sarcastic or humorous expressions, if they are meant to offend or hurt one or more persons, are included in this category.*

For this dataset, we again use the binary offensive/not offensive label, and we always preserve the default data splits provided by the authors. In particular, we use a specific dataset configuration that was also used by the authors for their experiments, with 2,160 class-balanced training examples, 540 development examples, and 3,057 test examples.

---

<sup>4</sup>While we are aware this does not exploit the most novel and interesting features of the MHS dataset, the exploration of annotator (dis)agreement with regards to data augmentation is beyond the scope of this work, and is left for future research.

<sup>5</sup><https://github.com/dhfbk/annotators-agreement-dataset>

## 3.6 HateCheck

In order to both test the out-of-distribution generalization of models and to explore their weaknesses, we also employ the HateCheck test suite (Röttger et al., 2021), consisting of 3,727 adversarial examples tailored at finding weaknesses of hate speech detection models. This dataset is created starting from a series of pre-defined templates aimed at testing different capabilities of hate speech detection models. The HateCheck test examples are divided into 29 functionalities, including various groups of targets of hate speech for each of the tests.

More specifically, the 29 tests fall into 11 classes:

- *Derogation*: 1. Expression of strong negative emotions, 2. Description using very negative attributes, 3. Dehumanization, 4. Implicit derogation;
- *Threatening language*: 5. Direct threat, 6. Threat as a normative statement;
- *Slur usage*: 7. Hate expressed using slur, 8. Non-hateful homonyms of slurs, 9. Reclaimed slurs;
- *Profanity usage*: 10. Hate expressed using profanity, 11. Non-hateful use of profanity;
- *Pronoun reference*: 12. Hate expressed through reference in subsequent clauses, 13. Hate expressed through reference in subsequent sentences;
- *Negation*: 14. Hate expressed using negated positive statement, 15. Non-hate expressed using negated hateful statement;
- *Phrasing*: 16. Hate phrased as a question, 17. Hate phrased as an opinion;

- *Non-hateful group identifiers*: 18. Neutral statements using protected group identifiers, 19. Positive statements using protected group identifiers;
- *Counter speech*: 20. Denouncements of hate that quote it, 21. Denouncements of hate that make direct reference to it;
- *Abuse against non-protected targets*: 22. Abuse targeted at objects, 23. Abuse targeted at individuals, 24. Abuse targeted at non protected groups;
- *Spelling variations*: 25. Swaps of adjacent characters, 26. Missing characters, 27. Missing word boundaries, 28. Added spaces between characters, 29. Leet speak.

### 3.7 Data Splits and Preprocessing

We use the default train/test splits of each dataset, where available and unless otherwise stated. For Founta et al. (2018), which has no default splits, we randomly partition the data into train and test using an 80/20 split. We also remove the substring “RT:” from the beginning of sequences in the Founta et al. (2018) dataset, since it is extremely common and it could be a confounder for the model. In addition to this, it has been found to be associated with hate speech in this dataset (Ramponi and Tonelli, 2022). For all datasets, we replace URLs and user mentions with URL and @USER respectively. We then remove all duplicates.

Since there is a partial overlap between SBIC and Founta et al. (2018), we remove instances that are present in the test set of either dataset from the training data of the other, to ensure fair cross-dataset evaluation.



# Chapter 4

## Robustness

While generative data augmentation (Section 2.2.1) has been shown to be potentially useful for the task of detecting offensive and abusive language online, several aspects and implications of it remain underexplored, especially with regards to the impact of data augmentation on model *robustness*, intended as the ability to generalize to out-of-distribution scenarios (Ramponi et al., 2022). Generative DA has, in fact, mostly been shown to work for offensive language detection when starting with a single specific dataset and using a specific generation setup, with no investigation of the impact of different generation configurations on the quality of the augmented data, as well as little exploration of cross-dataset or cross-domain performance.

In this chapter, we perform a series of experiments with the aim of exploring research questions **RQ1.1**, examining the *performance* of models trained on data augmented with synthetic texts, and more importantly **RQ1.2**, investigating the *robustness* implication of a generative data augmentation setup. In order to study different aspects and axes of variation across setups, we follow previous work on data augmentation, starting with a small set of gold data and augmenting it with synthetic data.

We proceed as follows for our experiments:

- i)* we train and test our models using four English offensive language datasets, testing both within dataset and cross-dataset performance;
- ii)* we simulate two low-resource scenarios, in which we start with different quantities of gold examples (500 or 2,000 texts);
- iii)* we compare four different generation prompting setups, of which two were used in previous work and two are novel;
- iv)* we experiment with different thresholds for filtering the generated data prior to using it for training.

Furthermore, we conduct a qualitative analysis on the generated data, with a focus on *lexical bias*. In order to do this, we compute the correlation between tokens in offensive texts using a PMI-based metric<sup>1</sup>, and we test the models trained on augmented data on the **HateCheck** suite (Röttger et al., 2021, Sec. 3.6), which includes a series of functional tests aimed at finding model weaknesses.

We use four English datasets annotated for offensive or abusive language for training and testing our models:

- The **Founta** et al. (2018) dataset (Sec. 3.1);
- The **OLID** dataset (Sec. 3.2);
- The Multi-Domain Agreement (**MDA**) dataset (Sec. 3.5);
- The Social Bias Inference Corpus (**SBIC**) (Sec. 3.3).

The choice of datasets was dictated by the fact that we aimed to represent different types of resources. The first two (**Founta** and **OLID**)

---

<sup>1</sup>For this, we use the Variationist tool (Appendix A).

are widely used in related work, and we deem them as more straightforward to classify, since standard BERT-based approaches trained and tested on these datasets yield results above 0.90 macro- $F_1$  (Zhou et al., 2021; Zampieri et al., 2020). In contrast, the **MDA** dataset was explicitly created to study disagreement among annotators focusing on different topics, so it contains more challenging instances. On this dataset, the best performance reported by the authors is  $\sim 0.75$  macro- $F_1$  (Leonardelli et al., 2021). Finally, the **SBIC** dataset includes data from different sources, with annotations for diverse targets of hate. The best classification result reported by the authors is  $\sim 0.80$   $F_1$  (Sap et al., 2020). Intuitively, the different characteristics of these datasets should allow us to assess the out-of-domain behavior of models when doing cross-dataset testing.

## 4.1 Methods

In order to compare the performance of different data augmentation setups, both novel and already employed in previous work, we implement the process displayed in Figure 4.1, which is composed of the following steps:

1. We randomly undersample the training data, obtaining the data subset  $X$  consisting of  $n$  examples (with  $n \in \{500; 2000\}$ ) (Sec. 4.2.1).
2. We fine-tune the pre-trained classification model  $C$  on  $X$ , obtaining  $C_X$ , which is used as a baseline and filtering classifier.
3. Depending on the type of generation input (Sec. 4.2.2) the pre-trained generation model  $G$  is fine-tuned on the available training data  $X$ , obtaining  $G_X$ .
4. The generative model  $G_X$  is used to generate synthetic examples.

5. The examples generated by  $G_X$  are pre-processed and then filtered based on the probability assigned to them by the classification model  $C_X$  (Sec. 4.2.3).
6. The generated data is merged with the gold data  $X$  to create the augmented dataset  $X_{aug}$ .
7. The classifier  $C$  is fine-tuned on the augmented dataset  $X_{aug}$  to create  $C_{X_{aug}}$ .

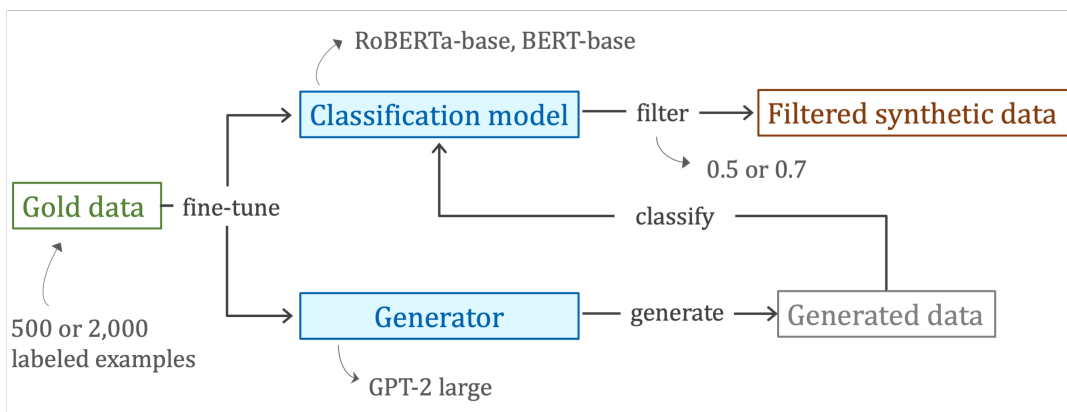


Figure 4.1: The generative DA pipeline we follow in our *robustness* experiments, from gold data to filtered synthetic data.

**Model choice** We generate synthetic data using GPT-2 large (Radford et al., 2019, 774M parameters).<sup>2</sup> Some recent works exploit the generative capabilities of newer models for the creation of new datasets, either in human-in-the-loop setups (Liu et al., 2022) or in very resource-

<sup>2</sup>We performed preliminary experiments using GPT-2 small (117M parameters) as well, finding that overall the generated data had a similar impact on classification performance.



intensive scenarios (Hartvigsen et al., 2022). However, we choose to experiment with GPT-2 for this set of experiments because it is freely accessible and it can be easily fine-tuned, and we aim for our results to be comparable with those of previous work where this DA method was found effective for this task, which used GPT-2 as well (e.g. Juuti et al. (2020) and Wullach et al. (2021)).

**Model Details** For classification, we run our experiments with the BERT base uncased model (110M parameters) (Devlin et al., 2019) and with RoBERTa base (125M parameters) (Liu et al., 2019). We use the Huggingface implementation (Wolf et al., 2020) for all models. In both cases we use the default Huggingface `TrainingArguments` class hyperparameters, with batch size set to 32. Classifiers and generative models are trained for 3 epochs.

Before generation, we fine-tune GPT-2 large using the default Huggingface hyperparameters, setting the batch size to 2, adding learning rate warm-up with a ratio of 0.02 and weight decay of 0.01. For fine-tuning GPT-2, the input texts are grouped into documents of maximum length 512 tokens and separated using end-of-sequence tokens.

After fine-tuning, the generation step is similar for all models. We use *top p* decoding (Holtzman et al., 2020) with  $p = 0.9$  and we set the minimum and maximum lengths of generated sequences to 5 and 100 tokens respectively. We also blacklist the sequence “@USER” so that it will not be generated, since it is very frequent in the normalized training data.

In all setups, we aim at augmenting the gold data with 2,000 synthetic examples. This number is chosen to at least double the available training data in all setups, and it is kept constant for easier model comparison. We generate 6,000 sequences for each setup, to increase the chances that

enough acceptable sequences will be generated. This estimate is based on the approach of Wullach et al. (2021), who preserve roughly 1/3 of the generated texts after filtering.

All experiments were run on a NVIDIA Quadro RTX 5000 GPU in  $\sim 80$  hours total, including both training and inference for all setups.

## 4.2 Experimental Setting

We structure our experiments along three axes of variation, with the aim of assessing their impact on model performance. The explored dimensions are further detailed in the following subsections.

- **Number of training instances.** In order to simulate two low-resource scenarios where different amounts of gold data are available, we train both classification and generative models with different amounts of labeled instances. Our aim is that of assessing how much the usefulness of generative DA changes when starting with datasets of different sizes (Sec. 4.2.1).
- **Prompting.** Different methods can be used for steering the generation towards one label or the other. We use two methods found in previous works, as well as two novel methods, to assess whether certain prompting methods lead to differences in synthetic data quality (Sec. 4.2.2).
- **Classifier filtering thresholds.** Since prompting methods are not always enough to steer the model into generating correct sequence-label pairs (Kumar et al., 2020), classifiers can be used to confirm or discard the label assignments made by the generative model (Anaby-Tavor et al., 2020; Wullach et al., 2021). In our experiments, we feed the generated sequences to a classifier (our baseline) and use the

probability given by the classifier to each generated sequence to either accept the label assigned by the generator or discard the sequence entirely. We experiment with two probability thresholds, in order to assess whether the confidence of the classifier is associated with generated data quality (Sec. 4.2.3).

Each model is tested on its own test data (within-dataset) and on the test data for the other datasets (cross-dataset).

### 4.2.1 Number of Training Instances

Each experiment is performed on varying amounts of training data, randomly sampling  $n = 500$  or  $2,000$  examples from each dataset, equally split between the two labels. We use 500 examples as the smallest sample size for our experiments since the smallest dataset size for this task found by Vidgen and Derczynski (2020) is 469 examples. We use 2,000 examples as the larger sample size given that it is still a relatively small dataset size for deep learning approaches and it reflects the size of many offensive language detection datasets.

We balance the sampling by class to avoid imbalance between gold and augmented data, consistently keeping this proportion even across all experiments. For the **MDA** dataset, sampling is stratified by agreement level as well. Balancing the classes might make our setup less “realistic”, given that it does not reflect the actual label distribution of each dataset. However, it is a way for us to control the impact of class balance differences between datasets on cross-dataset performance. It also helps to avoid differences in class balance between the gold data and the generated data, which could cause differences in model performance between setups regardless of the actual quality of the generated data.

Out of the available data,  $1/5$  ( $n = 500$ ) or  $1/10$  ( $n = 2,000$ ) is held



Figure 4.2: Summary of the types of prompts we use for training models in our *robustness* experiments. In this example, the **label** is *not offensive*, and the **text** is *You go girl!*.

out for validation purposes. For instance, in the data with 500 examples, 400 are used for training the models and 100 are kept for validation.

### 4.2.2 Prompting

We fine-tune GPT-2 using four data formatting setups. Two of the setups have been employed in previous works, while two are novel and aim at exploring the ability of the model to leverage natural language task descriptions for label assignment. A visual summary of the four prompts is shown in Figure 4.2.

#### Label tag prompting [*tag-prompt*]

Following the prompting type in Anaby-Tavor et al. (2020), we fine-tune the generator  $G$  by pre-pending the label  $y$  to each training sequence  $x$ ,

dividing the two with the separator “[SEP]”. In this setup, the inputs are concatenated into documents as follows:

“ $y_1$  [SEP]  $x_1$  [EOS]  $y_2$  [SEP] ...”

At generation time, the model is prompted with the desired label  $y$  followed by the separation token, and it is expected to generate a sequence belonging to the  $y$  class.

#### **Label in natural language prompting [*nl-prompt*].**

This is the first input setup we propose. It is inspired by the findings of Schick and Schütze (2021), in which natural language descriptions of tasks are found to be helpful for few-shot classification tasks. In this setup, the generator  $G$  is trained on sequences so that the label  $y$  is contextualized within the text using natural language. The training documents for fine-tuning the generators are structured as:

“This message is  $y_1$ .  $x_1$  [EOS] This ...”

Where  $y$  corresponds to *offensive* or *not offensive* depending on the label. At generation time, the model is prompted with “This message is  $y$ ”, where  $y$  is the desired label. The sequence produced after the prompt is expected to belong to the  $y$  class.

#### **Cloze question prompting [*cloze-prompt*]**

Again inspired by the findings in Schick and Schütze (2021), we propose another setup that exploits the capability of large language models of learning from patterns in natural language. In this case, however, the prompt relies on the auto-regressive nature of GPT-2, in which the probability of each token is modeled on the previous tokens. The main aim behind this setup is assessing whether placing the label information at the beginning or at the end of the sequence affects the quality of the

generated data. In this setup, each sequence  $x$  is followed by the cloze question “Is that offensive?” and the label is placed at the end of the sequence, in the form of a Yes/No answer.

“ $x_1$ . Is that offensive? {Y/N} [EOS] ...”

At generation time, the model receives no prompting, and it is expected to generate both the sequence and the cloze question / answer pair in the correct format. This type of prompting is more prone than the previously listed ones to generating sequences that will eventually be discarded, since it is expected to not only correctly generate sequences and assign them to a label, but also to produce a cloze question that follows a specific format.

#### One model per label [1/label]

This setup requires no actual prompting to steer the generation, since it involves one model for each label rather than one model for all labels. Following Juuti et al. (2020) and Wullach et al. (2021), the training dataset  $X$  is divided into  $X_o$  and  $X_n$  based on the *offensive* or *non-offensive* labels. The generative model  $G$  is then fine-tuned on  $X_o$  and  $X_n$  separately, producing two models for the generation of new data:  $G_o$  and  $G_n$ . In this setup, the messages are simply concatenated into documents and separated by end of sequence ([EOS]) tokens:

“ $x_1$  [EOS]  $x_2$  [EOS]  $x_3$  ...”

At generation time, each model is expected to generate sequences belonging to the class it was fine-tuned on.

### 4.2.3 Classifier Filtering Thresholds

After generation, the synthetic sequences are stripped of any prompting and automatically assigned the label that emerged during generation.

We discard any sequence that is  $\leq 5$  characters long, and normalize the generated data by removing user mentions and URLs, as well as duplicates (as described in Section 3.7).

We then feed the sequences into the baseline classifier trained on the same gold data as the generative model that produced them. Depending on the label probability assigned by the classifier to the generated sequences, these are accepted considering the following thresholds:

- The label predicted by the classifier matches the label assigned during the generation phase (label probability  $p > 0.5$ )
- The classifier predicts the same label assigned during generation with  $p > 0.7$ <sup>3</sup>

After filtering, we randomly select 2,000 generated examples from the accepted ones in each setup. In some rare cases, the number of accepted sequences after filtering can be lower than 2,000. In that case, we sample with replacement so that we can still use 2,000 synthetic examples and have similar training sizes across experiments, albeit sometimes with repeated sequences.

#### 4.2.4 Baselines

As baselines, we employ a BERT-base-uncased and a RoBERTa-base classifier trained on the same gold data used to fine-tune GPT-2 in each setup. We also report in our experimental results the performance of classifiers trained using simple random oversampling as a DA strategy, in which a number of randomly selected training examples appear multiple times during training. We match the number of oversampled instances with the number of synthetic examples we use for augmenting

---

<sup>3</sup>This is the same threshold used by Wullach et al. (2021).

the training data in each setup, split evenly across labels. Using over-sampling as a baseline allows us to compare more resource-intensive DA methods such as the ones we are evaluating with a simpler strategy.

### 4.3 Results

In this section we report the results of our experiments. Each experiment is run 10 times, with different random seeds. We report in our tables the average of these results. The metrics we use to evaluate models are macro- $F_1$  score and minority class  $F_1$  score.

In order to reliably compare the distributions of results across runs, we use Almost Stochastic Order (ASO) (Dror et al., 2019; Del Barrio et al., 2018) in its implementation by Ulmer et al. (2022). Following their findings, we use  $\tau = 0.2$  as a threshold for statistical significance.<sup>4</sup>

Table 4.1 and Table 4.2 show the results obtained by RoBERTa base models fine-tuned on augmented data when starting with 500 and 2,000 gold examples, respectively. BERT base results are reported in Tables 4.3 and 4.4. While for the setup in which we start with 2,000 annotated examples (Table 4.2) we use both filtering thresholds ( $p > 0.5$  and  $p > 0.7$ ), for the setup in which we start with 500 examples we report the results for models trained on generated data filtered with the  $p > 0.5$  threshold only. The reason for this is that with less data, the confidence of the model is much lower, and not all 10 runs can generate enough examples that are classified with a confidence score higher than 0.7.

**Impact of number of training instances** Overall, it appears that data augmentation is more effective in very low-resource scenarios, such as the

---

<sup>4</sup>This threshold has a Type I error rate comparable to that of a  $p$ -value threshold of 0.05 (Ulmer et al., 2022).



Gold data: 500 examples		Test			
Train: <b>MDA</b>		<b>MDA</b>	<b>Founta</b>	<b>SBIC</b>	<b>SOLID</b>
No augmentation		.655 (.603)	.805 (.743)	.543 (.537)	.807 (.734)
Oversampling		<b>.725 (.662)*</b>	<b>.882 (.846)</b>	.554 (.522)	<b>.825 (.757)*</b>
Filtering: $p > 0.5$	<i>tag-prompt</i>	.700 (.638)*	.859 (.810)*	.547 (.524)	.862 (.804)*
	<i>nl-prompt</i>	.694 (.638)*	.863 (.820)*	.560 (.546)	.863 (.806)*
	<i>cloze-prompt</i>	.692 (.634)*	.860 (.815)*	.545 (.524)	.859 (.803)*
	<i>1/label</i>	.716 (.656)*	.872 (.834)*	<b>.572 (.567)</b>	.874 (.823)*
Train: <b>Founta</b>		<b>MDA</b>	<b>Founta</b>	<b>SBIC</b>	<b>SOLID</b>
No augmentation		.683 (.622)	.904 (.874)	.540 (.504)	.888 (.844)
Oversampling		.637 (.609)	.900 (.871)	<b>.589 (.582)*</b>	.896 (.856)
Filtering: $p > 0.5$	<i>tag-prompt</i>	.679 (.620)	.909 (.881)	.567 (.542)	<b>.897 (.857)</b>
	<i>nl-prompt</i>	.660 (.611)	.909 (.882)	<b>.589 (.575)*</b>	.895 (.854)
	<i>cloze-prompt</i>	<b>.688 (.626)</b>	<b>.913 (.884)*</b>	.559 (.527)	.891 (.850)
	<i>1/label</i>	.683 (.624)	.910 (.882)	.579 (.563)*	.893 (.851)
Train: <b>SBIC</b>		<b>MDA</b>	<b>Founta</b>	<b>SBIC</b>	<b>SOLID</b>
No augmentation		.556 (.413)	.646 (.472)	.746 (.780)	.714 (.570)
Oversampling		<b>.591 (.506)*</b>	<b>.700 (.564)*</b>	<b>.780 (.814)*</b>	<b>.766 (.653)*</b>
Filtering: $p > 0.5$	<i>tag-prompt</i>	.561 (.447)	.679 (.531)	.765 (.805)*	.744 (.618)
	<i>nl-prompt</i>	.578 (.449)	.687 (.540)	.763 (.803)*	.746 (.622)
	<i>cloze-prompt</i>	.574 (.438)	.663 (.497)	.762 (.799)*	.737 (.604)
	<i>1/label</i>	.584 (.477)*	.676 (.524)	.771 (.805)*	.757 (.636)*
Train: <b>OLID</b>		<b>MDA</b>	<b>Founta</b>	<b>SBIC</b>	<b>SOLID</b>
No augmentation		.568 (.515)	.766 (.676)	.585 (.588)	.797 (.707)
Oversampling		.584 (.591)	<b>.838 (.799)*</b>	<b>.637 (.687)*</b>	<b>.865 (.819)*</b>
Filtering: $p > 0.5$	<i>tag-prompt</i>	.578 (.567)	.812 (.755)	.610 (.644)	.845 (.786)
	<i>nl-prompt</i>	.581 (.564)	.811 (.763)	.615 (.652)	.838 (.781)
	<i>cloze-prompt</i>	<b>.586 (.565)</b>	.816 (.763)	.618 (.656)	.843 (.783)
	<i>1/label</i>	.575 (.584)	.831 (.791)	.631 (.697)	.855 (.810)

Table 4.1: Average macro- $F_1$  scores (over 10 runs) obtained by RoBERTa-base fine-tuned on augmented data, starting with 500 gold examples.  $F_1$  scores for the minority class are in parentheses. Grey cells contain within-dataset results, while the others contain cross-dataset results. Asterisks denote statistically significant results (compared to no augmentation). The best result for each train-test dataset combination is in bold.

Gold data: 2,000 examples		Test			
Train: MDA	MDA	Founta	SBIC	SOLID	
No augmentation	.770 (.708)	<b>.900</b> (.861)	.568 (.580)	.895 (.840)	
Oversampling	.761 (.699)	.894 (.860)	.592 (.575)*	.877 (.823)	
Filtering: $p > 0.5$	<i>tag-prompt</i>	<b>.773</b> (.714)	<b>.900</b> (.868)	.582 (.563)*	.890 (.840)
	<i>nl-prompt</i>	.771 (.712)	<b>.900</b> (.867)	.576 (.555)	.895 (.850)
	<i>cloze-prompt</i>	.771 (.713)	<b>.900</b> (.868)	.576 (.555)	<b>.896</b> (.850)
	<i>1/label</i>	.769 (.712)	.893 (.861)	.594 (.585)*	.885 (.837)
Filtering: $p > 0.7$	<i>tag-prompt</i>	.766 (.708)	.895 (.861)	.590 (.580)*	.887 (.840)
	<i>nl-prompt</i>	.771 (.714)	.898 (.866)	.586 (.572)*	.892 (.847)
	<i>cloze-prompt</i>	.769 (.712)	.897 (.864)	.586 (.570)*	.891 (.846)
	<i>1/label</i>	.768 (.713)	.894 (.862)	<b>.596</b> (.586)*	.886 (.838)
Train: Founta	MDA	Founta	SBIC	SOLID	
No augmentation	.635 (.619)	.910 (.883)	.611 (.612)	.904 (.866)	
Oversampling	.628 (.610)	.907 (.880)	.615 (.618)	.901 (.862)	
Filtering: $p > 0.5$	<i>tag-prompt</i>	.645 (.620)	.911 (.883)	.614 (.618)	.901 (.863)
	<i>nl-prompt</i>	.635 (.616)	.911 (.885)	.625 (.633)	.905 (.868)
	<i>cloze-prompt</i>	.644 (.619)	<b>.915</b> (.888)	.607 (.607)	.906 (.870)
	<i>1/label</i>	.633 (.613)	.910 (.881)	.612 (.615)	.902 (.864)
Filtering: $p > 0.7$	<i>tag-prompt</i>	<b>.650</b> (.623)	.913 (.885)	.619 (.624)	.903 (.865)
	<i>nl-prompt</i>	.645 (.619)	.914 (.887)	.615 (.617)	<b>.908</b> (.872)
	<i>cloze-prompt</i>	.640 (.619)	.913 (.885)	<b>.621</b> (.625)	.904 (.866)
	<i>1/label</i>	.647 (.619)	.914 (.886)	.612 (.614)	.907 (.871)
Train: SBIC	MDA	Founta	SBIC	SOLID	
No augmentation	.608 (.555)	<b>.737</b> (.618)	.813 (.844)	.804 (.712)	
Oversampling	.591 (.544)	.722 (.601)	.810 (.842)	.789 (.691)	
Filtering: $p > 0.5$	<i>tag-prompt</i>	.603 (.550)	.725 (.597)	.812 (.840)	.803 (.708)
	<i>nl-prompt</i>	.604 (.547)	.730 (.605)	<b>.814</b> (.844)	.802 (.708)
	<i>cloze-prompt</i>	.608 (.552)	.729 (.607)	<b>.814</b> (.844)	.806 (.714)
	<i>1/label</i>	.606 (.548)	.725 (.598)	.811 (.840)	.800 (.704)
Filtering: $p > 0.7$	<i>tag-prompt</i>	.608 (.560)	.733 (.611)	.811 (.841)	<b>.807</b> (.716)
	<i>nl-prompt</i>	<b>.618</b> (.546)	.724 (.593)	<b>.814</b> (.842)	.801 (.703)
	<i>cloze-prompt</i>	.611 (.555)	.735 (.615)	.813 (.844)	<b>.807</b> (.714)
	<i>1/label</i>	.609 (.558)	.733 (.612)	<b>.814</b> (.844)	.804 (.709)
Train: OLID	MDA	Founta	SBIC	SOLID	
No augmentation	.584 (.599)	.874 (.841)	.633 (.668)	<b>.897</b> (.859)	
Oversampling	.576 (.593)	.858 (.824)	.637 (.678)	.887 (.847)	
Filtering: $p > 0.5$	<i>tag-prompt</i>	.570 (.593)	.867 (.832)	.636 (.681)	.891 (.852)
	<i>nl-prompt</i>	.586 (.598)	.875 (.841)	.641 (.681)	.895 (.856)
	<i>cloze-prompt</i>	<b>.592</b> (.603)	<b>.878</b> (.845)	.638 (.672)	<b>.897</b> (.861)
	<i>1/label</i>	.573 (.594)	.871 (.839)	<b>.644</b> (.687)	.892 (.855)
Filtering: $p > 0.7$	<i>tag-prompt</i>	.578 (.597)	.864 (.831)	.634 (.675)	.892 (.853)
	<i>nl-prompt</i>	.581 (.597)	.873 (.841)	.642 (.681)	.896 (.858)
	<i>cloze-prompt</i>	.582 (.597)	.871 (.839)	.638 (.676)	.895 (.857)
	<i>1/label</i>	.579 (.597)	.872 (.839)	.643 (.684)	.895 (.858)

Table 4.2: Average macro- $F_1$  scores (over 10 runs) obtained by RoBERTa-base fine-tuned on augmented data, starting with 2,000 gold examples.  $F_1$  scores for the minority class are in parentheses. Grey cells contain within-dataset results, while the others contain cross-dataset results. Asterisks denote statistically significant results (compared to no augmentation). The best result for each train-test dataset combination is in bold.

Gold data: 500 examples		Test			
<b>Train: MDA</b>		<b>MDA</b>	<b>Founta</b>	<b>SBIC</b>	<b>SOLID</b>
No augmentation		.630 (.551)	.716 (.620)	.550 (.583)	.700 (.635)
Oversampling		<b>.696 (.623)*</b>	<b>.823 (.768)*</b>	.573 (.566)	<b>.825 (.757)*</b>
Filtering: $p > 0.5$	<i>tag-prompt</i>	.663 (.583)	.775 (.698)*	.562 (.578)	.774 (.704)
	<i>nl-prompt</i>	.654 (.574)	.752 (.664)	<b>.584 (.610)</b>	.767 (.698)
	<i>cloze-prompt</i>	.665 (.589)*	.773 (.693)*	.554 (.562)	.780 (.709)*
	<i>1/label</i>	.688 (.617)*	.798 (.744)*	.575 (.604)	.797 (.728)*
<b>Train: FOUNTA</b>		<b>MDA</b>	<b>Founta</b>	<b>SBIC</b>	<b>SOLID</b>
No augmentation		.619 (.549)	.890 (.856)	.613 (.622)	.847 (.790)
Oversampling		<b>.638 (.585)</b>	<b>.906 (.876)*</b>	.598 (.591)	<b>.885 (.841)*</b>
Filtering: $p > 0.5$	<i>tag-prompt</i>	.636 (.564)	.904 (.874)	.600 (.597)	.876 (.826)*
	<i>nl-prompt</i>	.614 (.567)	.900 (.869)	<b>.641 (.665)</b>	.874 (.828)*
	<i>cloze-prompt</i>	.632 (.564)	.900 (.869)	.606 (.611)	.857 (.807)
	<i>1/label</i>	.629 (.574)	.899 (.870)	.633 (.654)	.878 (.834)*
<b>Train: SBIC</b>		<b>MDA</b>	<b>Founta</b>	<b>SBIC</b>	<b>SOLID</b>
No augmentation		.566 (.425)	.629 (.438)	.747 (.787)	.727 (.600)
Oversampling		<b>.579 (.481)*</b>	<b>.682 (.540)</b>	<b>.766 (.801)*</b>	<b>.756 (.643)</b>
Filtering: $p > 0.5$	<i>tag-prompt</i>	.575 (.426)	.679 (.530)	.754 (.796)	.755 (.640)
	<i>nl-prompt</i>	.576 (.451)	.677 (.523)	.757 (.797)	.754 (.640)
	<i>cloze-prompt</i>	.566 (.426)	.656 (.487)	.754 (.796)	.738 (.614)
	<i>1/label</i>	.574 (.447)	.664 (.500)	.762 (.799)*	.743 (.619)
<b>Train: OLID</b>		<b>MDA</b>	<b>Founta</b>	<b>SBIC</b>	<b>SOLID</b>
No augmentation		.555 (.538)	.757 (.691)	.635 (.712)	.770 (.704)
Oversampling		.555 (.570)	<b>.832 (.792)*</b>	.653 (.717)	<b>.852 (.804)*</b>
Filtering: $p > 0.5$	<i>tag-prompt</i>	.554 (.550)	.795 (.743)	.641 (.723)	.813 (.754)
	<i>nl-prompt</i>	.559 (.559)	.810 (.762)*	<b>.658 (.728)*</b>	.832 (.778)*
	<i>cloze-prompt</i>	<b>.562 (.553)</b>	.803 (.750)*	.648 (.720)	.823 (.766)
	<i>1/label</i>	.537 (.557)	.805 (.764)*	.648 (.750)	.821 (.769)*

Table 4.3: Average macro- $F_1$  scores (over 10 runs) obtained by BERT-base-uncased fine-tuned on augmented data, starting with 500 gold examples.  $F_1$  scores for the minority class are in parentheses. Grey cells contain within-dataset results, while the others contain cross-dataset results. Asterisks denote statistically significant results (compared to no augmentation). The best result for each train-test dataset combination is in bold.

Gold data: 2,000 examples		Test				
Train: MDA	MDA	Founta	SBIC	SOLID		
No augmentation	.756 (.694)	.894 (.860)	.573 (.550)	.891 (.844)		
Oversampling	.746 (.684)	.884 (.848)	.592 (.580)	.880 (.830)		
Filtering: $p > 0.5$	<i>tag-prompt</i>	.759 (.694)	.900 (.867)	.567 (.535)	.893 (.846)	
	<i>nl-prompt</i>	.761 (.697)	.901 (.869)*	.567 (.538)	<b>.900 (.860)*</b>	
	<i>cloze-prompt</i>	.756 (.697)	.901 (.869)*	.572 (.544)	.899 (.855)*	
	<i>1/label</i>	.749 (.689)	.892 (.859)	.584 (.571)	.891 (.843)	
Filtering: $p > 0.7$	<i>tag-prompt</i>	.760 (.695)	.899 (.867)	.578 (.554)	.893 (.844)	
	<i>nl-prompt</i>	.760 (.698)	.899 (.867)	.572 (.545)	.897 (.853)*	
	<i>cloze-prompt</i>	<b>.762 (.699)</b>	<b>.902 (.870)*</b>	.572 (.548)	.898 (.855)*	
	<i>1/label</i>	.753 (.690)	.897 (.863)	<b>.593 (.579)*</b>	.893 (.848)	
Train: FOUNTA	MDA	Founta	SBIC	SOLID		
No augmentation	.616 (.601)	.913 (.887)	.628 (.635)	.905 (.868)		
Oversampling	.635 (.604)*	.911 (.883)	.617 (.618)	.899 (.859)		
Filtering: $p > 0.5$	<i>tag-prompt</i>	.634 (.605)*	.914 (.888)	.621 (.625)	.905 (.867)	
	<i>nl-prompt</i>	.632 (.607)	.914 (.887)	.627 (.632)	.905 (.868)	
	<i>cloze-prompt</i>	.617 (.599)	.914 (.887)	<b>.630 (.641)</b>	.903 (.867)	
	<i>1/label</i>	.629 (.604)	.914 (.888)	.628 (.637)	.904 (.866)	
Filtering: $p > 0.7$	<i>tag-prompt</i>	<b>.636 (.609)*</b>	.913 (.887)	.624 (.629)	.905 (.868)	
	<i>nl-prompt</i>	.634 (.605)	<b>.915 (.888)</b>	.629 (.636)	.904 (.867)	
	<i>cloze-prompt</i>	.633 (.608)	.914 (.888)	<b>.630 (.636)</b>	<b>.907 (.871)</b>	
	<i>1/label</i>	.629 (.605)	.913 (.884)	.627 (.635)	.903 (.864)	
Train: SBIC	MDA	Founta	SBIC	SOLID		
No augmentation	.589 (.539)	<b>.743 (.632)</b>	.806 (.838)	.807 (.718)		
Oversampling	.588 (.526)	.716 (.590)	.799 (.829)	.786 (.683)		
Filtering: $p > 0.5$	<i>tag-prompt</i>	.584 (.538)	.742 (.626)	.806 (.839)	<b>.809 (.717)</b>	
	<i>nl-prompt</i>	.594 (.526)	.734 (.590)	.807 (.829)	.802 (.683)	
	<i>cloze-prompt</i>	.593 (.532)	.735 (.617)	.806 (.838)	.802 (.705)	
	<i>1/label</i>	.586 (.532)	.739 (.628)	.804 (.835)	.800 (.705)	
Filtering: $p > 0.7$	<i>tag-prompt</i>	.582 (.536)	<b>.743 (.629)</b>	<b>.809 (.842)</b>	.806 (.714)	
	<i>nl-prompt</i>	.588 (.538)	.734 (.616)	.806 (.840)	.803 (.712)	
	<i>cloze-prompt</i>	<b>.598 (.539)</b>	.742 (.628)	.807 (.837)	.807 (.716)	
	<i>1/label</i>	.591 (.533)	.732 (.614)	.806 (.835)	.803 (.709)	
Train: OLID	MDA	Founta	SBIC	SOLID		
No augmentation	.562 (.588)	.874 (.843)	.653 (.693)	.897 (.861)		
Oversampling	.549 (.580)	.859 (.824)	.661 (.709)	.885 (.845)		
Filtering: $p > 0.5$	<i>tag-prompt</i>	.555 (.583)	.862 (.829)	.662 (.713)	.890 (.852)	
	<i>nl-prompt</i>	.553 (.582)	.868 (.834)	.668 (.717)*	.893 (.855)	
	<i>cloze-prompt</i>	.568 (.588)	.875 (.845)	.659 (.700)	.897 (.861)	
	<i>1/label</i>	.541 (.578)	.858 (.824)	.669 (.728)*	.885 (.844)	
Filtering: $p > 0.7$	<i>tag-prompt</i>	.555 (.583)	.862 (.826)	.663 (.712)	.892 (.853)	
	<i>nl-prompt</i>	.561 (.586)	.873 (.841)	.665 (.712)*	.896 (.859)	
	<i>cloze-prompt</i>	<b>.575 (.591)</b>	<b>.879 (.846)</b>	.658 (.698)	<b>.898 (.862)</b>	
	<i>1/label</i>	.548 (.581)	.863 (.829)	<b>.671 (.725)</b>	.889 (.851)	

Table 4.4: Average macro- $F_1$  scores (over 10 runs) obtained by BERT-base-uncased fine-tuned on augmented data, starting with 2,000 gold examples.  $F_1$  scores for the minority class are in parentheses. Grey cells contain within-dataset results, while the others contain cross-dataset results. Asterisks denote statistically significant results (compared to no augmentation). The best result for each train-test dataset combination is in bold.

setting with 500 examples. The fact that DA is more useful as the amount of available data lowers is in line with what has been observed for other tasks, as well as in multiclass setups, albeit with a much lower number of examples per class (Anaby-Tavor et al., 2020; Kumar et al., 2020). In the setup in which 2,000 gold examples are available, there are very few significant improvements in performance when using generative data augmentation.

**Impact of prompting and filtering** Interestingly, no prompting type seems to clearly outperform the others across setups. For instance, augmenting the **MDA** dataset starting with 500 gold examples has a positive effect on performance across all prompting types both when tested on the in-domain test data and when tested on **Founta** and **SOLID**, while when tested on **SBIC** none of the setups lead to significant improvements in performance. This seems to indicate that dataset characteristics have a greater impact than the prompting setup on whether generative DA can be effective in this kind of scenario. However, looking at Table 4.2, the situation is reversed: the RoBERTa model trained on **MDA** only significantly benefits from data augmentation when tested on **SBIC** across most setups. A filtering threshold of 0.7 does seem to help improve performance at least marginally, but only on this dataset combination out of all the ones we tested. Overall, it appears that whether DA will have a positive impact on classification might not depend much on the generation setup in our case.

**Overall findings** The most important pattern that emerges from our results is that generative DA using GPT-2 does not appear to reliably improve model performance across setups, both in and out of domain. It apparently *can* significantly improve model performance, especially for

some dataset combinations and with very low amounts of data. However, this improvement is not consistent, so based on our results we would consider this type of DA unreliable as a method for improving offensive language classifiers in similar setups.

Another important aspect that emerges from our results is that oversampling is a very strong baseline, especially for the setup with 500 available annotated examples, even though it is often overlooked. To our knowledge, it was used as a baseline only in Juuti et al. (2020) for generative DA on this task, while most other works report the performance on augmented data only. Interestingly, oversampling does not only improve within-dataset performance, but it also has a significant positive impact on cross-dataset performance, even though intuitively it would be expected to lead to overfitting of the training data. Since it requires a fraction of the computational resources needed for generative DA, it may be preferable when  $\sim 500$  gold examples are available. We hypothesize that one of the reasons why oversampling can perform well is that at least a subset of the datasets share superficial features that might be amplified in the oversampling process, such as specific terms that are associated with offensiveness across datasets. We will explore lexical aspects in Section 4.4.2.

The results for BERT models are in general in line with those for RoBERTa models, although BERT models tend to perform worse regardless of setup. Again, with BERT models, oversampling seems to be just as reliable to improve both within-dataset and cross-dataset performance. In general, although it does not reliably improve model performance, generative DA does not seem to significantly decrease performance either. Wullach et al. (2021) believe that generative DA could improve lexical diversity, leading to better generalization. In Section 4.4, we examine the generated data from a qualitative point of view, to as-

sess whether it could lead to benefits with regards to lexical variety, or changes in performance on the **HateCheck** tests.

## 4.4 Qualitative Analysis

In this section, we examine the generated texts from a qualitative point of view, first through a manual analysis of a small subset of generated texts (Sec. 4.4.1), then through a lexical analysis of the synthetic texts (Sec. 4.4.2), and finally we discuss the results obtained by models trained on augmented data on HateCheck, investigating any changes in performance across both specific adversarial tests and targets of hate (Sec. 4.4.3).

### 4.4.1 Examples of Generated Texts

In order to estimate the quality of the generated examples and the impact of the prompting method, we randomly select a small subset of 10 generated examples for every dataset, setup, and data size combination for manual analysis, for a total of 320 examples. We find that there are some clear differences between the prompting setups, and that the methods that exploit prompting in natural language, i.e. *nl-prompt* (Sec. 4.2.2) and *cloze-prompt* (Sec. 4.2.2), tend to generate the most realistic examples. *Tag-prompt* tends to often generate strings of random special characters, resulting in very low quality data, while the *1/label* setup often results in sequences that appear out of domain. Some examples of the generated texts are shown in Table 4.5 for models trained on 500 examples and Table 4.6 for models trained on 2,000 annotated texts. Some texts have been truncated (signaled by “[...]”) due to space constraints. The labels reported in Tables 4.5 and 4.6 are those assigned by the RoBERTa baseline.

prompting	dataset	generated text	label
<i>tag-prompt</i>	MDA	#SEP #BlackLivesMatter #BIDEN #MAGA URL	0
	Founta	~~~@USER	0
	SBIC	~~~~~ [..]	0
	OLID	*****@USER You're fucking awesome	0
<i>nl-prompt</i>	MDA	@USER It is time to end this corrupt criminal republic #DemocRATting #BlackLivesMatter URL	0
	Founta	????????????????????????????????	1
	SBIC	I hate all of you who call yourself patriots. You are useless.	1
	OLID	@USER @USER you are not a Christian..you are a hypocrite	1
<i>cloze-prompt</i>	MDA	"I love how the media is pushing the idea that the protesters are racist	0
	Founta	USER: hey, u should really try and figure out what is going on in the world of 4chan	0
	SBIC	If you can't get a girl to like you, you're a loser..	1
	OLID	@USER @USER He is the worst	0
<i>1/label</i>	MDA	Coronavirus was first identified in 1984 in the U.S. as a con- taminant of the popular oral flu vaccine. Because of the severity of this illness, the CDC in the early 90s ordered the removal of all Coronaviruses from vaccines and in 1998 [...]	0
	Founta	#6a8a2a color RGB value is (135,18 ,225). #6b98f6 color hex is (13 ,180,,25). the number is on its way to being determined so I don't need to change it [...]	0
	SBIC	"The best way to get ahead in life is to keep learning" - Steve Jobs "As a scientist, the thing I am most excited about is the posi- bility that we will be able to engineer a cure for cancer - by using a synthetic biology-based approach" - Dr. Aubrey de Grey, M.D.[...]	0
	OLID	EVERYBODY, EVERYBODY is talking about the fact that President Trump is being sued by the media. Yes, you read that correctly. Media. Lawsuit. #MAGA #Trump #To- riesOut #ToryOut'sOut#POTUSOut #MAGADump #MAG- AMAG #MAGAZINEAU'S MAGA NEWSLETTER @USER @USER The way he [...]	0

Table 4.5: Examples of sequences generated by GPT-2 large models trained on 500 gold examples for every combination of prompting and dataset.



prompting	dataset	generated text	label
<i>tag-prompt</i>	MDA	URL #Elections2020	0
	Founta	!!!! &#128580;&#127907;&amp;&gt; I fucking hate you URL	1
	SBIC	phone 7, apple iphones 8, iphONES 8, android ips 8, blackberry ips, ixp8 iph, [...]	0
	OLID	URL You guys are pathetic	1
<i>nl-prompt</i>	MDA	@USER Because the Democrats lie, cheat, steal and intentionally create the chaos in our country. They are beyond evil!	0
	Founta	????????? @USER We are listening..if you have any ideas please send us a tweet at URL	0
	SBIC	You’re never too young to dream, you just gotta work it.	0
	OLID	You and your friends are fucking delusional. There are far worse far right wing nut jobs out there than antifa. Try to keep your stupid comment to yourself	1
<i>cloze-prompt</i>	MDA	The most disgusting thing about this is that the person spreading this filth is a member of the media. They are enabling the filth.	0
	Founta	@USER @USER Yeah, that would be a pretty crazy idea! I’m sure that would cause an uproar and a few angry people would object	1
	SBIC	Do you think these fake news articles on vaccines and autism are getting too much play on social media?	0
	OLID	@USER @USER @USER @USER He is just one more nut job in Congress.	1
<i>1/label</i>	MDA	The family of a missing teen has filed a \$5 million lawsuit against the city of Madison, alleging that officials didn’t do enough to find their son when he went missing in September 2010. [...]	0
	Founta	\$4.99 Featuring all the fan favorites including: • Exclusive and classic illustrations from some of comics’ most beloved artists [...]	0
	SBIC	Beef and broccoli are the perfect meal for when you are hungry and not hungry. But I think they would probably turn to dust in an instant if you tried to eat them.Blessings are spread in the name of the Lord, and His name is exalted. [...]	0
	OLID	WWF @USER We are all outraged that this clown is using his access to the office to threaten &amp; bully. You &amp;#Array; you &amp;#Array; yourself should be ashamed of yourself.@USER you are so right on this issue but I have to agree that we conservatives are becoming too emotional. [...]	1

Table 4.6: Examples of sequences generated by GPT-2 large models trained on 2,000 gold examples for every combination of prompting and dataset.

### 4.4.2 Lexical Analysis

To investigate the lexical variation between the gold data and the generated data, we use the Variationist tool (Appendix A). In particular, we calculate a normalized weighted relevance metric based on point-wise mutual information (PMI), following Ramponi and Tonelli (2022). We analyze the most informative tokens for the *offensive* class in each dataset, looking at how certain tokens become less or more informative in the generated data. For this analysis, we choose to use the same sub-word tokens used by the models rather than whitespace-separated tokens, in order to get a better glimpse at the informativeness of tokens for the classifiers we use.

In this section, we present the lists of top-11 informative tokens for the offensive class, both on gold and on generated data. Lists for data in the setup where we start with 500 annotated instances can be found in Table 4.7, and those for the setup with 2,000 gold instances are in Table 4.8.

The first tendency that can be noticed when looking at how the ranking of tokens' informativeness changes between gold and synthetic data is that for some of the datasets the changes are more evident (i.e. for **MDA** and **SBIC**). For example, in the gold **SBIC** data, the word *fucking* is ranked as the 1,203rd most informative word for the *offensive* class. In data augmented using the *tag-prompt* type on the generative model trained on 2,000 instances, however, the same word is ranked 4th. This means that the model has generated a very large amount of offensive messages containing this word, while it was not prominent in the gold data it was fine-tuned on. This happens for both the setup starting with 500 gold examples and the one with 2,000.

While the prominence of a potentially predictive word for offensive

MDA									
gold data		tag-prompt		nl-prompt		cloze-prompt		1/label	
index	token	index	token	index	token	index	token	index	token
0	fuck	0	fuck	2	fucking	2	fucking	0	fuck
1	shit	2	fucking	0	fuck	1	shit	2	fucking
2	fucking	1	shit	1	shit	0	fuck	1	shit
3	ass	31	racist	5	dumb	6	stupid	6	stupid
4	idiot	5	dumb	6	stupid	31	racist	3	ass
5	dumb	6	stupid	31	racist	5	dumb	7	##s
6	stupid	3	ass	7	##s	3	ass	11	guy
7	##s	25	mor	3	ass	302	disgusting	17	piece
8	b*tch	13	trump	4	idiot	4	idiot	5	dumb
9	##er	4	idiot	302	disgusting	18	user	4	idiot
10	bullshit	423	people	17	piece	25	mor	31	racist

FOUNTA									
gold data		tag-prompt		nl-prompt		cloze-prompt		1/label	
index	token	index	token	index	token	index	token	index	token
0	fucking	0	fucking	0	fucking	0	fucking	0	fucking
1	fucked	4	fuck	3	b*tch	4	fuck	4	fuck
2	user	2	user	4	fuck	2	user	6	hate
3	b*tch	6	hate	6	hate	6	hate	11	shit
4	fuck	3	b*tch	11	shit	3	b*tch	1	fucked
5	ass	5	ass	10	stupid	5	ass	10	stupid
6	hate	1	fucked	8	idiot	1	fucked	3	b*tch
7	128	11	shit	5	ass	10	stupid	5	ass
8	idiot	10	stupid	1	fucked	11	shit	8	idiot
9	##gga	8	idiot	43	sick	8	idiot	43	sick
10	stupid	43	sick	41	##tar	43	sick	34	kill

SBIC									
gold data		tag-prompt		nl-prompt		cloze-prompt		1/label	
index	token	index	token	index	token	index	token	index	token
0	black	0	black	0	black	0	black	0	black
1	b*tch	4	white	4	white	3	difference	29	woman
2	##es	9264	[SEP]	38	people	12	girl	5	sex
3	difference	38	people	3	difference	4	white	8	women
4	white	11	##s	12	girl	29	woman	38	people
5	sex	3	difference	31	person	5	sex	4	white
6	ho	5382	fucking	29	woman	80	guy	12	girl
7	##gga	31	person	17	##gger	31	person	57	racist
8	women	29	woman	5382	fucking	38	people	14	gay
9	jew	8	women	7	##gga	8	women	80	guy
10	fuck	10	fuck	8	women	5382	fucking	44	kill

OLID									
gold data		tag-prompt		nl-prompt		cloze-prompt		1/label	
index	token	index	token	index	token	index	token	index	token
0	shit	0	shit	0	shit	19	disgusting	0	shit
1	fuck	16	people	6	liberals	6	liberals	7	stupid
2	ass	19	disgusting	1	fuck	16	people	1	fuck
3	fucking	6	liberals	19	disgusting	0	shit	3	fucking
4	##s	28	hate	52	sick	7	stupid	16	people
5	b*tch	18	racist	7	stupid	9	idiot	52	sick
6	liberals	7	stupid	3	fucking	28	hate	19	disgusting
7	stupid	52	sick	16	people	22	liar	99	wrong
8	control	22	liar	28	hate	1	fuck	29	disgrace
9	idiot	1	fuck	9	idiot	18	racist	31	bad
10	dumb	10	dumb	22	liar	10	dumb	97	women

Table 4.7: Top tokens for the *offensive* class in the gold data and in the generated data when starting with 500 examples, computed using the PMI implementation of Ramponi and Tonelli (2022). The indices refer to the ranking of importance of the tokens in the gold data, while the order of the tokens reflect their informativeness for the offensive class in the generated data.

MDA									
gold data		tag-prompt		nl-prompt		cloze-prompt		1/label	
index	token	index	token	index	token	index	token	index	token
0	fuck	0	fuck	18	##ass	2	fucking	0	fuck
1	shit	2	fucking	52	##est	0	fuck	1	shit
2	fucking	1	shit	16	##on	1	shit	2	fucking
3	ass	23	racist	418	##path	950	user	14	mag
4	##s	7	dumb	4	##s	6	idiot	23	racist
5	stupid	6	idiot	3	ass	5	stupid	6	idiot
6	idiot	89	liar	10	asshole	23	racist	22	##a
7	dumb	5	stupid	12	b*tch	7	dumb	5	stupid
8	piece	135	##trum	9	bullshit	89	liar	7	dumb
9	bullshit	14	mag	105	bunch	8	piece	8	piece
10	asshole	1284	##p	1049	complete	3	ass	13	guy

FOUNTA									
gold data		tag-prompt		nl-prompt		cloze-prompt		1/label	
index	token	index	token	index	token	index	token	index	token
0	fucking	0	fucking	0	fucking	0	fucking	0	fucking
1	fucked	6	hate	1	fucked	1	fucked	4	fuck
2	user	1	fucked	4	fuck	2	user	1	fucked
3	b*tch	4	fuck	3	b*tch	6	hate	6	hate
4	fuck	3	b*tch	9	shit	4	fuck	3	b*tch
5	ass	16339	[SEP]	6	hate	3	b*tch	5	ass
6	hate	5	ass	5	ass	5	ass	9	shit
7	##gga	8	shit	11	stupid	10	idiot	11	stupid
8	128	7	##gga	20	sick	11	stupid	10	idiot
9	shit	2	user	7	##gga	9	shit	20	sick
10	idiot	11	stupid	19	mad	7	##gga	8	128

SBIC									
gold data		tag-prompt		nl-prompt		cloze-prompt		1/label	
index	token	index	token	index	token	index	token	index	token
0	black	0	black	0	black	0	black	0	black
1	b*tch	15008	[SEP]	3	white	3	white	3	white
2	difference	10	##s	1	b*tch	2	difference	5	sex
3	white	10203	fucking	14	##gger	15	woman	2	difference
4	##es	1	b*tch	12	jews	8	women	8	women
5	sex	763	offensive	7	##gga	12	jews	15	woman
6	ho	3	white	2	difference	5	sex	9	fuck
7	##gga	5	sex	10	##s	11	jew	43	racist
8	women	9	fuck	11	jew	19	girl	16	##ist
9	fuck	43	racist	15	woman	1	b*tch	1	b*tch
10	##s	4	##es	8	women	14	##gger	11	jew

OLID									
gold data		tag-prompt		nl-prompt		cloze-prompt		1/label	
index	token	index	token	index	token	index	token	index	token
0	shit	11	liberals	11	liberals	11	liberals	0	shit
1	fuck	12	disgusting	1	fuck	0	shit	12	disgusting
2	ass	7	people	0	shit	12	disgusting	6	stupid
3	fucking	0	shit	12	disgusting	6	stupid	7	people
4	b*tch	13	racist	53	disgrace	18	liar	1	fuck
5	##s	6	stupid	6	stupid	53	disgrace	14	sick
6	stupid	53	disgrace	14	sick	26	##yp	13	racist
7	people	26	##yp	18	liar	14	sick	18	liar
8	idiot	29	##oc	3	fucking	29	##oc	3	fucking
9	dumb	14	sick	7	people	32	lying	26	##yp
10	user	16	fake	5	##s	1	fuck	29	##oc

Table 4.8: Top tokens for the *offensive* class in the gold data and in the generated data when starting with 2,000 examples, computed using the PMI implementation of Ramponi and Tonelli (2022). The indices refer to the ranking of importance of the tokens in the gold data, while the order of the tokens reflect their informativeness for the offensive class in the generated data.

language, like  *fucking* , is potentially a good sign of the quality of the generated data, since it means that the generation process can usefully augment existing data, this happens also with tokens that should not be predictive for this task, defined by Ramponi and Tonelli (2022) as  *spurious lexical artifacts* . For example, across several generation setups, we can notice the increased importance for the offensive label of identity terms, such as  *girl* ,  *gay* ,  *woman* ,  *jews* , but also words that hold no inherent association with abusive language, such as  *people*  or  *difference* . The higher importance of some identity terms could be associated with biases that were already present in the pre-trained generative model, and which were amplified during the generation process. In order to better understand the impact of these differences in token informativeness, we turn to the HateCheck functional tests in Section 4.4.3.

### 4.4.3 HateCheck Analysis

In this section, we explore some of the weaknesses of classifiers trained on gold and augmented data through the HateCheck suite (Sec. 3.6). Since our models in this section are more generally aimed at detecting offensive language and HateCheck is focused on hate speech, i.e. a narrower phenomenon, not all tests are entirely informative in our case, such as test number 11, testing the performance on  *non-hateful profanities* , as these are considered  *offensive*  in the Founta and OLID datasets, while they are not  *hate speech* . In general, however, the labels of HateCheck tests are aligned with our task, so these tests can still be informative.

Table 4.9 and Table 4.10 present the results on HateCheck tests and targets for models in the 500 gold examples setup. Table 4.11 and Table 4.12, on the other hand, present the results on the functional tests and targets for models in the setup in which we start with 2,000 annotated examples.

Overall, data augmentation has little impact on the performance of models on HateCheck, especially on models trained starting with 2,000 gold instances. Similarly to the results reported in Section 4.3, oversampling tends to lead to similar or better improvements than generative DA, again depending on the starting dataset. However, this improvement is focused only on some functional tests, leading to worse performance on other tests. For instance, two interesting tests are those on *non-hateful slurs*, including test 8, non-hateful homonyms of slurs (e.g. “Female canines are called bitches”), and test 9, *reclaimed slurs* (e.g. “The Q in LGBTQ stands for queer”). For models trained on 500 gold instances, the performance consistently drops on these two tests for all augmentation setups and oversampling across all datasets. This suggests that models trained using oversampling or generative data augmentation may start overfitting and relying on the presence of specific terms such as slurs for the classification of offensive language. For models trained on 2,000 gold instances, the performance on these two tests is somewhat stable, with no improvements or decreases in performance.

Another interesting test for examining whether generative DA increases the reliance of classifiers on specific words is test 15, *negated hateful statement*, e.g. “I would never hurt any woman”. The performance for this test when using augmented data tends to be overall stable or lower than the baseline, mostly decreasing with oversampling and in the setup starting from 500 annotated instances. This again indicates that models trained on augmented data in the setups we used show potentially harmful weaknesses, since they tend to misclassify sequences based on simple patterns such as the presence of specific words.

While models trained on augmented data tend to perform worse on non-hateful sequences containing identity terms and slurs, they do show improvements for those tests that benefit from being able to find these

terms, such as test 7, *hate expressed using slur*, or test 10, *hate expressed using profanity*, further confirming that this type of augmentation tends to steer models into overfitting identity terms and slurs.

Functionality	MDA				FOUNTA				SBIC				OLID						
	base	over	tag	nl	cloze	1/I	base	over	tag	nl	cloze	1/I	base	over	tag	nl	cloze	1/I	
1 derog_neg_emote.h	0.22	0.19	0.25	0.26	0.22	0.17	0.53	0.79	0.73	0.77	0.61	0.69	0.71	0.80	0.74	0.71	0.79	0.82	0.77
2 derog_neg_attrib.h	0.44	0.56	0.55	0.50	0.50	0.52	0.74	0.91	0.83	0.89	0.81	0.86	0.89	0.87	0.85	0.83	0.90	0.92	0.82
3 derog_dehum.h	0.39	0.58	0.50	0.46	0.43	0.47	0.50	0.82	0.71	0.82	0.62	0.72	0.90	0.90	0.86	0.83	0.89	0.93	0.75
4 derog_impl.h	0.09	0.15	0.11	0.12	0.08	0.11	0.14	0.47	0.27	0.41	0.19	0.30	0.90	0.90	0.84	0.85	0.91	0.91	0.54
5 threat_dir.h	0.19	0.27	0.21	0.21	0.20	0.17	0.31	0.69	0.56	0.65	0.40	0.54	0.87	0.93	0.82	0.84	0.89	0.92	0.63
6 threat_norm.h	0.18	0.22	0.16	0.20	0.19	0.18	0.31	0.69	0.56	0.65	0.40	0.54	0.87	0.93	0.82	0.84	0.89	0.92	0.63
7 slur.h	0.60	0.74	0.65	0.65	0.65	0.68	0.57	0.74	0.66	0.70	0.69	0.73	0.77	0.82	0.76	0.79	0.79	0.82	0.69
8 slur_homonym_nh	0.74	0.55	0.60	0.59	0.63	0.59	0.64	0.54	0.51	0.51	0.50	0.53	0.53	0.50	0.53	0.47	0.48	0.49	0.67
9 slur_reclaimed_nh	0.28	0.24	0.25	0.25	0.24	0.26	0.22	0.18	0.20	0.19	0.19	0.19	0.33	0.21	0.33	0.29	0.28	0.26	0.34
10 profanity_h	0.88	0.94	0.93	0.93	0.93	0.92	1.00	0.99	1.00	1.00	1.00	1.00	0.90	0.90	0.84	0.85	0.93	0.92	0.89
11 profanity_nh	0.20	0.11	0.12	0.11	0.12	0.14	0.01	0.02	0.00	0.00	0.00	0.01	0.56	0.63	0.59	0.49	0.56	0.70	0.17
12 ref_subs_clause.h	0.34	0.39	0.41	0.46	0.43	0.38	0.49	0.74	0.68	0.73	0.61	0.69	0.85	0.92	0.79	0.80	0.88	0.90	0.79
13 ref_subs_sent.h	0.49	0.49	0.56	0.57	0.57	0.54	0.61	0.82	0.78	0.85	0.70	0.77	0.89	0.93	0.83	0.83	0.91	0.94	0.86
14 negate_pos.h	0.04	0.07	0.09	0.07	0.05	0.04	0.05	0.35	0.26	0.30	0.17	0.26	0.85	0.85	0.77	0.81	0.84	0.83	0.45
15 negate_neg.h	0.90	0.87	0.84	0.88	0.87	0.87	0.76	0.34	0.46	0.42	0.63	0.52	0.17	0.12	0.21	0.19	0.13	0.43	0.30
16 phrase_question.h	0.40	0.34	0.36	0.42	0.41	0.38	0.57	0.80	0.72	0.83	0.62	0.73	0.99	0.94	0.98	0.99	0.99	0.97	0.83
17 phrase_opinion.h	0.37	0.51	0.45	0.47	0.44	0.46	0.56	0.80	0.72	0.80	0.66	0.72	0.81	0.91	0.78	0.78	0.88	0.91	0.78
18 ident_neutral_nh	0.98	0.98	0.99	0.98	0.99	0.98	1.00	0.88	0.96	0.92	0.95	0.94	0.37	0.36	0.44	0.35	0.40	0.39	0.96
19 ident_pos_nh	0.98	0.98	0.98	0.98	0.99	0.97	1.00	0.84	0.94	0.95	0.91	0.91	0.36	0.34	0.42	0.39	0.38	0.40	0.92
20 counter_quote_nh	0.42	0.52	0.47	0.42	0.50	0.45	0.24	0.11	0.17	0.08	0.22	0.16	0.21	0.21	0.13	0.19	0.12	0.14	0.14
21 counter_ref_nh	0.48	0.42	0.45	0.43	0.44	0.43	0.26	0.11	0.19	0.10	0.24	0.17	0.10	0.14	0.16	0.14	0.07	0.11	0.16
22 target_obj_nh	0.75	0.74	0.75	0.76	0.77	0.78	0.61	0.49	0.56	0.47	0.58	0.58	0.52	0.68	0.62	0.60	0.62	0.68	0.39
23 target_indiv_nh	0.62	0.58	0.55	0.60	0.60	0.61	0.42	0.21	0.22	0.14	0.33	0.28	0.52	0.44	0.52	0.44	0.49	0.55	0.33
24 target_group_nh	0.65	0.57	0.60	0.65	0.65	0.64	0.57	0.32	0.39	0.31	0.51	0.45	0.30	0.33	0.36	0.34	0.32	0.36	0.33
25 spell_char_swap.h	0.27	0.28	0.30	0.25	0.25	0.27	0.30	0.56	0.51	0.57	0.50	0.56	0.74	0.79	0.76	0.75	0.80	0.87	0.40

Table 4.9: Accuracy on the first 25 functional HateCheck tests of RoBERTa models trained on 500 gold examples and on the augmented data.

Target	MDA				FOUNTA				SBIC				OLID						
	base	over	tag	nl	cloze	1/I	base	over	tag	nl	cloze	1/I	base	over	tag	nl	cloze	1/I	
1 women	0.44	0.49	0.47	0.47	0.45	0.46	0.50	0.58	0.56	0.59	0.53	0.56	0.50	0.53	0.50	0.48	0.52	0.58	0.57
2 trans people	0.43	0.45	0.44	0.45	0.44	0.41	0.50	0.56	0.56	0.58	0.52	0.57	0.48	0.51	0.49	0.46	0.51	0.52	0.56
3 gay people	0.50	0.53	0.51	0.52	0.50	0.53	0.56	0.57	0.60	0.63	0.56	0.59	0.48	0.50	0.48	0.46	0.50	0.51	0.59
4 black people	0.47	0.49	0.49	0.49	0.48	0.46	0.54	0.61	0.60	0.65	0.56	0.61	0.47	0.48	0.47	0.44	0.49	0.50	0.59
5 disabled people	0.40	0.44	0.44	0.44	0.44	0.40	0.45	0.54	0.54	0.57	0.51	0.52	0.50	0.53	0.51	0.49	0.52	0.55	0.55
6 Muslims	0.43	0.49	0.46	0.45	0.45	0.46	0.50	0.60	0.56	0.61	0.54	0.57	0.49	0.52	0.51	0.47	0.52	0.54	0.55
7 immigrants	0.39	0.45	0.43	0.42	0.42	0.42	0.44	0.54	0.52	0.54	0.50	0.51	0.51	0.55	0.53	0.49	0.53	0.55	0.53

Table 4.10: Accuracy on the different types of targets in the HateCheck tests of RoBERTa models trained on 500 gold examples and on the augmented data.



Functionality	MDA				FOUNTA				SBIC				OLID			
	base	over	tag	1/I	base	over	tag	1/I	base	over	tag	1/I	base	over	tag	1/I
1 derog_neg.emote.h	0.07	0.20	0.11	0.07	0.12	0.91	0.89	0.91	0.91	0.87	0.88	0.89	0.91	0.89	0.91	0.95
2 derog_neg.attrib.h	0.51	0.57	0.57	0.55	0.56	0.94	0.94	0.94	0.95	0.95	0.98	0.98	0.98	0.99	1.00	0.99
3 derog_neg.dehum.h	0.57	0.66	0.67	0.64	0.62	0.66	0.90	0.89	0.89	0.92	0.96	0.98	0.96	0.97	0.95	0.98
4 derog_impl.h	0.11	0.20	0.13	0.13	0.11	0.16	0.59	0.53	0.54	0.50	0.92	0.94	0.89	0.76	0.75	0.68
5 threat_dir.h	0.23	0.46	0.34	0.25	0.28	0.34	0.87	0.86	0.84	0.87	0.95	0.97	0.93	0.87	0.90	0.85
6 threat_norm.h	0.11	0.37	0.18	0.12	0.11	0.22	0.81	0.87	0.81	0.82	0.78	0.95	0.97	0.94	0.90	0.90
7 slur.h	0.76	0.76	0.79	0.77	0.76	0.79	0.77	0.76	0.79	0.77	0.85	0.88	0.84	0.85	0.90	0.86
8 slur.homonym_nh	0.50	0.50	0.50	0.50	0.50	0.46	0.49	0.46	0.47	0.47	0.45	0.45	0.47	0.46	0.42	0.48
9 slur.reclaimed_nh	0.22	0.23	0.22	0.22	0.22	0.23	0.14	0.14	0.14	0.15	0.21	0.24	0.19	0.18	0.19	0.18
10 profanity.h	0.95	0.95	0.92	0.93	0.94	0.93	1.00	1.00	1.00	1.00	0.99	0.98	0.99	1.00	1.00	1.00
11 profanity_nh	0.11	0.11	0.12	0.11	0.10	0.10	0.00	0.01	0.00	0.00	0.65	0.66	0.66	0.65	0.65	0.65
12 ref.subs.clause.h	0.36	0.46	0.39	0.40	0.38	0.45	0.88	0.87	0.87	0.90	0.87	0.96	0.98	0.94	0.97	0.93
13 ref.subs.sent.h	0.51	0.53	0.51	0.52	0.52	0.55	0.93	0.93	0.90	0.93	0.97	0.97	0.97	0.98	0.95	0.98
14 negate_pos.h	0.02	0.05	0.03	0.03	0.02	0.06	0.43	0.52	0.46	0.45	0.46	0.83	0.93	0.80	0.57	0.63
15 negate_neg.h	0.86	0.78	0.86	0.85	0.88	0.83	0.22	0.19	0.21	0.20	0.21	0.25	0.09	0.08	0.12	0.22
16 phrase.question.h	0.32	0.40	0.33	0.33	0.32	0.35	0.87	0.89	0.89	0.90	0.88	0.95	0.96	0.97	0.95	0.91
17 phrase.opinion.h	0.49	0.57	0.51	0.50	0.49	0.54	0.89	0.90	0.87	0.88	0.90	0.85	0.96	0.95	0.97	0.98
18 ident.neutral_nh	0.99	0.96	0.97	0.98	0.99	0.99	0.82	0.78	0.75	0.85	0.84	0.82	0.52	0.36	0.45	0.34
19 ident.pos_nh	0.99	0.97	0.97	0.98	0.98	0.97	0.79	0.73	0.73	0.82	0.80	0.79	0.37	0.22	0.34	0.28
20 counter.quote_nh	0.48	0.48	0.47	0.47	0.50	0.45	0.08	0.08	0.12	0.09	0.06	0.07	0.05	0.06	0.05	0.05
21 counter.ref_nh	0.41	0.41	0.41	0.41	0.43	0.38	0.06	0.07	0.08	0.07	0.08	0.08	0.06	0.06	0.04	0.03
22 target.obj_nh	0.83	0.77	0.80	0.82	0.81	0.81	0.47	0.46	0.45	0.43	0.44	0.74	0.75	0.77	0.77	0.81
23 target.indiv_nh	0.60	0.48	0.51	0.57	0.55	0.60	0.13	0.13	0.14	0.13	0.16	0.38	0.39	0.33	0.37	0.47
24 target.group_nh	0.68	0.56	0.61	0.65	0.65	0.61	0.28	0.27	0.29	0.26	0.25	0.27	0.38	0.35	0.35	0.38
25 spell.char.swap.h	0.24	0.29	0.34	0.34	0.28	0.33	0.71	0.69	0.72	0.72	0.71	0.73	0.87	0.91	0.88	0.84

Table 4.11: Accuracy on the first 25 functional HateCheck tests of RoBERTa models trained on 2,000 gold examples and on the augmented data.

Target	MDA				FOUNTA				SBIC				OLID			
	base	over	tag	1/I	base	over	tag	1/I	base	over	tag	1/I	base	over	tag	1/I
1 women	0.47	0.51	0.50	0.48	0.47	0.51	0.62	0.61	0.62	0.61	0.59	0.57	0.59	0.66	0.67	0.68
2 trans people	0.42	0.46	0.45	0.45	0.44	0.44	0.59	0.57	0.61	0.60	0.55	0.53	0.54	0.63	0.65	0.66
3 gay people	0.50	0.55	0.53	0.51	0.51	0.53	0.58	0.57	0.55	0.61	0.52	0.48	0.51	0.63	0.60	0.63
4 black people	0.46	0.52	0.48	0.48	0.48	0.49	0.61	0.58	0.64	0.62	0.52	0.49	0.47	0.63	0.67	0.68
5 disabled people	0.43	0.45	0.45	0.45	0.44	0.44	0.57	0.57	0.59	0.59	0.57	0.56	0.59	0.66	0.65	0.68
6 Muslims	0.45	0.52	0.48	0.46	0.46	0.49	0.60	0.62	0.62	0.63	0.54	0.48	0.51	0.64	0.66	0.64
7 immigrants	0.43	0.48	0.45	0.44	0.44	0.46	0.58	0.59	0.57	0.58	0.59	0.56	0.57	0.63	0.65	0.66

Table 4.12: Accuracy on the different types of targets in the HateCheck tests of RoBERTa models trained on 2,000 gold examples and on the augmented data.

## 4.5 Conclusions on Robustness

In this chapter, we presented an evaluation of both existing and novel data augmentation setups based on generative large language models for offensive language detection. We investigated the robustness of such models, testing them in within-dataset and cross-dataset scenarios, and performed a qualitative analysis on the augmented data.

Relative to research question **RQ1.1**, we found that while generative DA can positively impact model *performance* in some cases, especially when low amounts of gold data are available, this positive effect is not consistent across setups, making generative DA unreliable in the setups we tested, with no clear improvements with regards to *robustness* (**RQ1.2**). In addition to this, we found that generative DA can potentially introduce lexical bias from the pre-trained generative model into the augmented data, as well as increase the reliance of models on identity terms and slurs, which could have unintended effects on classification.

Overall, although it might improve classification performance in some cases, using generative DA for this task using the setups we experimented with would be inadvisable, as it is computationally intensive and it does not appear to consistently make models perform better or be more robust.

# Chapter 5

## Fairness

In this chapter, we aim at investigating the effect of data augmentation on *fairness* (research question **RQ1.3**). More specifically, we explore the representation of target identity groups in synthetic data, as well as the possibility of exploiting synthetic data to make the identity groups that are represented in the data more evenly distributed, under the assumption that this might make systems trained on such data fairer. Indeed, no in-depth analysis of the effects of data augmentation for less-represented hate speech targets has been carried out in previous work, while it could potentially be beneficial not only to make systems more accurate and robust, but also fairer.

In order to conduct this analysis, we carry out a comparison between recent generative language models and more traditional approaches to data augmentation, such as synonym replacement or token deletion. Such a comparison has, in fact, not been carried out before to our knowledge, although increasing the amount of training data with synthetic examples has been successfully exploited well before the advent of generative large language models (Chen et al., 2023).

More specifically, we address the following research questions related to *fairness* (**RQ1.3**):

- **(RQ1.3.1)** Does data augmentation impact the performance of hate speech detection classifiers differently depending on specific target identities?
- **(RQ1.3.2)** Can information about identity groups in the data augmentation process help the creation of better and more representative synthetic examples?
- **(RQ1.3.3)** Do different data augmentation setups and approaches have distinct effects on the performance of models on underrepresented targets?

We answer the above questions through a set of experiments in which we focus on the performance of models by target identity group. In addition, we introduce two novel elements compared to previous work on generative DA: *(i)* we experiment with setups in which we exploit target identity information during generation, attempting to increase the relative representation of scarcely represented targets, and *(ii)* we experiment with instruction-finetuned LLMs, which have been shown to be able to improve downstream task performances (Wei et al., 2022). We carry out generation-based data augmentation using 4 different generative models, both with and without access to target identity information. We also further investigate potential fairness-related weaknesses of models using the HateCheck test suite (Sec. 3.6) combined with a manual analysis of generated examples.

## 5.1 Fairness Experiments Data

For our experiments, we use the **MHS** corpus (Sec. 3.4). The main characteristic of the MHS dataset that makes it ideal for our study is that it includes labels signaling the presence of identity groups and sub-groups

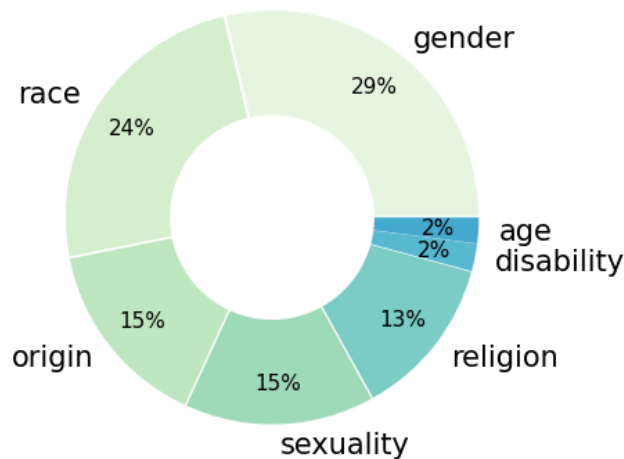


Figure 5.1: Identity group distribution in the MHS corpus.

in texts. Importantly, this annotation is present regardless of hatefulness, resulting in target annotations even for posts containing supportive or counter-speech.

In the entire corpus, only 48 examples are not associated with any target identity group. Two examples of texts and their annotations from the processed corpus we use are shown in Table 5.1.<sup>1</sup>

In the freely available version of the MHS dataset<sup>2</sup> we find annotations for seven target identity groups: *race*, *religion*, *origin*, *gender*, *sexuality*, *age*, and *disability*. Their distribution in the data can be seen in Figure 5.1, which shows how the most widely studied targets of hate speech, *race* and *gender*, are also the most widely represented in the MHS corpus, while some targets such as *age*, *disability*, or *religion* are less frequent.

<sup>1</sup>The original messages from the dataset were changed slightly to avoid using user content. Additionally, slurs are manually obfuscated by us, following the guidelines by Nozza and Hovy (2023).

<sup>2</sup><https://huggingface.co/datasets/ucberkeley-dlab/measuring-hate-speech>

Text	Hate speech	Target identity groups
I think it was a joke, no black girls eever dream of such things.	0	Gender, Race
no one gives a shit f*g	1	Sexuality
People would rather stand up on the bus than sit next to me, hijabi life is awesome	0	Religion

Table 5.1: Examples from the dataset after our aggregation operations, with a hate speech label and a list of target identity groups mentioned or referred to in the text.

## 5.2 Methodology

In our experiments, we implement a data augmentation pipeline inspired by Anaby-Tavor et al. (2020), which has also been used in other work (Wullach et al., 2021), similarly to that of Section 4.1. The pipeline is displayed in Figure 5.2. Starting from a small set of *Gold data* from the MHS corpus, a *Generator* is employed to augment them by generating synthetic examples with the corresponding label (hateful or not) using either *finetuning* or *few-shot prompting* (see Section 5.2.2). Since the labels associated with the *Generated data* may not be accurate, given that generative models cannot always preserve the desired labels (Kumar et al., 2020), a subsequent filtering step is used in order to maximize the chances of label correctness, similarly to that of Section 4.2.3. In order to create a model for filtering generated texts, the same gold data is used also to fine-tune a binary classifier that assigns a *hateful/non hateful* label to the generated data. However, differently from the filtering step of Section 4.2.3, in this case we preserve the synthetic examples for which there is a match between the label assigned during generation and by the *Classification model*, with no specific classifier confidence threshold, as our results of Chapter 4 showed small differences between the filtering thresholds. The *Filtered synthetic data* is then used to train a hate speech classifier that we evaluate for the task performance in general and then on specific hate targets. We detail below the variants we test for each step of the pipeline.

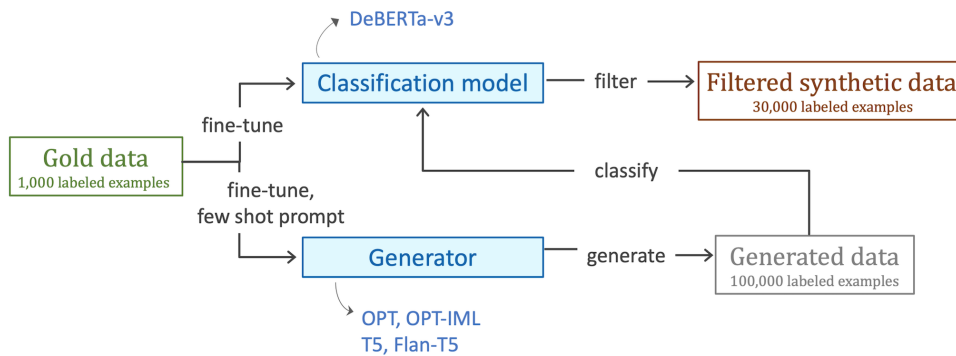


Figure 5.2: The generative DA pipeline, from gold data to filtered synthetic data.

### 5.2.1 Generative Models

For the **Generator** step, we experiment with four different transformer models: OPT (Zhang et al., 2022) and T5 (Raffel et al., 2020) and their instruction-finetuned counterparts: OPT-IML (Iyer et al., 2022) and Flan-T5 (Chung et al., 2024). We choose to only use openly available models for our experiments to favor reproducibility. The selected models allow us also to compare a decoder-only model (OPT) with an encoder-decoder model (T5), which to our knowledge has not been done in previous studies on this type of data augmentation (Azam et al., 2022). Another aspect we want to investigate is the performance of instruction-finetuned models compared to their standard version, since recent works showed that instruction-tuning can improve generalization to unseen tasks (Chung et al., 2024). For this reason, we include OPT-IML and Flan-T5 beside OPT and T5. We use the 1.3B parameter version of OPT and OPT-IML and the Large version of T5 and Flan-T5 (770M).<sup>3</sup>

<sup>3</sup>While this means there is a disparity between the decoder-only and the encoder-decoder model sizes we use, 1.3B is the smallest available model size for OPT-IML, and finetuning Flan-T5 3B, the next model size available for the encoder-decoder architecture, was beyond our computing capacity at the time the experiments were carried out.

## 5.2.2 Finetuning, Few-Shot Prompting, and Identity Group Information

A large number of works on data augmentation based on generative models rely on finetuning a model on a small set of gold data, and then generating new data with the finetuned model, encoding the label information within the text sequences in some form (e.g. Anaby-Tavor et al. (2020); Kumar et al. (2020)), as seen in Section 2.1.2. Other works use few-shot demonstration-based prompting, in which the pre-trained model is prompted with one or more sequences similar to what the model is expected to generate, with no finetuning (e.g. Hartvigsen et al. (2022); Azam et al. (2022); Ashida and Komachi (2022)). We experiment with both strategies for each transformer model.

Since our research questions revolve around the impact that information regarding target identity groups can have on data augmentation, finetuning and few-shot demonstration-based prompting are further tested in two variants: with and without mentioning the **target identity group information**. Our hypothesis is that the inclusion of this kind of information might help in generating more varied data with regards to identity group mentions for both hateful and non-hateful messages. By generating target-specific examples also for the non-hateful class, we ideally aim at implicitly contrasting identity term bias (Zhou et al., 2021). In order to do this, we encode target identity information into the prompts given to the models using different methods, as shown below.

### Finetuning (FT)

For finetuning, we follow an approach similar to that of Anaby-Tavor et al. (2020), in which a generative LLM is fine-tuned on annotated se-



quences that are concatenated with labels. At generation time, the desired label information is fed into the model, and the model is expected to generate a sequence belonging to the specified class. We discuss the details of the formatting of the label information in Section 5.2.2.

This method has the upside of theoretically being more likely to generate examples that are closer to the original distribution of the data to be augmented. However, this can also be a downside, if the desired effect is increasing the variety of the data. In addition, finetuning is more computationally expensive than few-shot prompting. For models fine-tuned with target identity information, given that each sequence can be associated with more than one target (in cases of intersectional hate speech for instance), the label-encoding sequence will include the list of all targets mentioned or referred to in that post.

### **Few-shot prompting (FS)**

Following the large amount of works focusing on few-shot demonstration-based instructions, especially with instruction-finetuned models (Iyer et al., 2022; Chung et al., 2024), we also experiment with demonstration-based prompting, in which the models are shown 3 examples belonging to the desired label (and target identity, if available to that particular model), and then asked to produce a new one.

With models exploiting target identity information for few-shot prompting, we associate the desired label and target with 3 sequences. For instance, if the model is expected to generate a non-hateful post about the identity group *gender*, we select 3 sequences that are annotated in the gold data as non-hateful and about *gender* (although they might be annotated as being *also* about other identity groups).

## Formatting

As regards prompt formatting, we aim at using the same type of prompting layout across experiments. We choose to use prompting sequences in natural language, given that we found them to lead to generally more realistic generated examples in Chapter 4. In order to find prompts in natural language that could be leveraged by our models, we consulted the FLAN corpus (Wei et al., 2022), which is part of the finetuning data of both FLAN-T5 and OPT-IML. Among the instruction templates, we find one of the CommonGen templates (Lin et al., 2020) to fit with our aims:

*‘Write a sentence about the following things: [concepts], [target]’.*

We reformulate it to obtain a prompting sequence that reflects our application, and can be exploited by instruction-finetuned models:

*‘Write a [∅/ hateful] social media post [∅/ about t],’*

where  $t$  is one of the 7 target identity groups in the MHS corpus.

Table 5.2 presents the sequences and prompts used for training and prompting our models. For the encoder-decoder models, the prompting sequence is the input and the desired text is the output at finetuning time.

Target	Write a [∅ — hateful] social media post about $\{t\}$ : $\{\text{text}\}$
No target	Write a [∅ — hateful] social media post: $\{\text{text}\}$

Table 5.2: Templates used for fine-tuning and prompting generative models during the generation step.

To summarize, four transformer models (OPT, OPT-IML, T5 and Flan-T5) generate synthetic data using finetuning or few-shot prompting, either having access to target information or not, for a total of sixteen different strategies for generative DA that we evaluate in our experiments.

### 5.3 Experimental Setup

For all experiments, following previous work on DA, we simulate a setup in which we have a small amount of gold data available prior to augmentation, similarly to the experiments in Chapter 4 (see Figure 5.2 for an overview of the data augmentation pipeline for our *fairness* experiments). We randomly select 1,000 gold examples from the MHS corpus, as we deem it a realistically small dataset size for a hate speech detection corpus based on Vidgen and Derczynski (2020). Our goal is to create a larger dataset out of the starting 1,000 examples. Given that the ‘natural’ size of the Measuring HS dataset is around 35k examples, we aim for 30k new annotated examples to use in augmentation, which will result in a 31k example dataset for each setup.

Wullach et al. (2021), using a similar DA approach, preserve around 1/3 of the generated examples after filtering. We therefore generate around 3 times as many examples as we need, similarly to the setup presented in Section 4.1, setting the total number of generated examples for each setup to 100,000, equally divided into *hateful* and *not hateful*. For each setup, we thus prompt models to generate 50k examples for each class.

Given that our focus is on different targets of hate, we aim at investigating the impact of their representation in the data on model performance. Specifically, since their distribution in the MHS corpus is highly imbalanced, as seen in Figure 5.1, we hypothesize that their representation might influence the performance of models. Because of this, we choose to equally augment each target identity category (gender, race, origin, sexuality, religion, disability, and age). Indeed, for models that rely on target identity information, out of the 50k generated instances for each class, we generate 1/7 for each target identity category (7,140

newly generated sequences for each target identity group).

Once we generate 100,000 examples, we filter them by predicting their hate speech label using a DeBERTa-v3 Large (He et al., 2023) classifier, finetuned on the initial 1,000 gold examples. We only preserve the examples for which the classifier label assignment matches the desired label that was in the model input at generation time, in line with previous work that used this kind of filtering for synthetic sequences. If less than 15k generated sequences per class pass filtering, we preserve the examples that did pass filtering for that class, and proceed with the rest of the pipeline. In some setups, this means that we end up with fewer synthetic examples than initially expected. We discuss this more in depth in Section 5.4. We then test the quality of the synthetic data extrinsically, by using it in addition to the initial available gold data for training classifiers aimed at detecting the presence of hate speech for specific targets.

### 5.3.1 Implementation details with Generative DA

For all of our experiments, we employ the HuggingFace library (Wolf et al., 2020). All the hyperparameters we use that are not specified in this section are the default ones from their `TrainingArguments` class.

We fine-tune T5 Flan-T5, OPT and OPT-IML with batch 16 and  $LR = 1e - 3$ . For generation, we use  $top-p=0.9$  decoding and set min and max lengths of generated sequences to 5 and 150 tokens, respectively. The DeBERTa classifiers we use as baselines and for filtering are trained for 5 epochs. All the classifiers that are trained on augmented data are trained for 3 epochs (given that they are trained on more data, they require less epochs to converge) with batch size 16 and  $LR=5e - 6$ . In this case, at the end of training, we preserve the model from the epoch with the lowest evaluation cross-entropy loss.

The random seeds we used for shuffling, subsampling the gold data, and initializing both generative and classification models are 522, 97, 709, 16, and 42. Finetuning of all classifiers and generative models, including baselines and models trained on augmented data, took 50 hours, of which 45 on a Nvidia V100 GPU and 5 on a Nvidia A40. Inference time for generating all of the sequences (a total of 8 million generated texts) took  $\sim$ 500 hours total.

### 5.3.2 System comparison

We compare all of our models with **Easy Data Augmentation (EDA)** (Wei and Zou, 2019) in its implementation by Marivate and Sefara (2020). EDA consists of four operations: synonym replacement using WordNet (Miller, 1992), random insertion, random swap, and random deletion of tokens. Similarly to our other setups, we produce 30k new sequences with EDA, of which 7,500 with each operation, on the initial 1,000 examples in each fold. We then also experiment with the mixture of EDA and generative DA, in which instead of augmenting the initial gold data with 30k synthetic sequences obtained with EDA or generative DA, we randomly select 15k examples of LLM-generated texts and 15k examples of EDA-perturbed examples and concatenate them.

We also implement **two baselines** using DeBERTa: *i*) the classifier finetuned on the starting 1k gold examples, and *ii*) the same classifier finetuned on an oversampled version of the training data (repeating the initial 1k sequences until we get to 31k, the size of the augmented setups), which we already found to be a competitive baseline even in cross-dataset scenarios in Chapter 4.

			M-F <sub>1</sub>	Hate-F <sub>1</sub>	Hate-F <sub>1</sub>							n(h)
					Gender	Race	Origin	Sexuality	Religion	Disability	Age	
<b>No augmentation</b>			.773 <sup>.02</sup>	.652 <sup>.03</sup>	.635 <sup>.02</sup>	.696 <sup>.04</sup>	.497 <sup>.05</sup>	.756 <sup>.03</sup>	.485 <sup>.12</sup>	.698 <sup>.03</sup>	.545 <sup>.04</sup>	
<b>Oversampling</b>			.773 <sup>.02</sup>	.653 <sup>.04</sup>	.652 <sup>.05</sup>	.740 <sup>.02</sup> *	.568 <sup>.05</sup> *	.787 <sup>.02</sup> *	.571 <sup>.03</sup> *	.732 <sup>.04</sup> ◇	.555 <sup>.06</sup>	
<b>EDA</b>			.799 <sup>.01</sup> *	.714 <sup>.01</sup> *	.687 <sup>.01</sup> *	.771 <sup>.02</sup> *	.582 <sup>.03</sup> *	.806 <sup>.01</sup> *	.601 <sup>.02</sup> *	.799 <sup>.02</sup> *	.589 <sup>.06</sup> *	15k
Model	Target											
OPT	FT	Y	.783 <sup>.00</sup> *	.683 <sup>.01</sup> *	.653 <sup>.03</sup> ◇	.740 <sup>.02</sup> *	.556 <sup>.05</sup> *	.779 <sup>.02</sup> ◇	.535 <sup>.07</sup>	.777 <sup>.02</sup> *	.587 <sup>.06</sup> ◇	0.5k
		N	.774 <sup>.04</sup>	.652 <sup>.07</sup>	.634 <sup>.05</sup>	.707 <sup>.06</sup>	.505 <sup>.06</sup>	.738 <sup>.10</sup>	.461 <sup>.07</sup>	.690 <sup>.11</sup>	.590 <sup>.10</sup> ◇	15k
	FS	Y	.782 <sup>.01</sup> ◇	.691 <sup>.02</sup> *	.667 <sup>.02</sup> *	.750 <sup>.02</sup> *	.553 <sup>.04</sup> *	.790 <sup>.01</sup> *	.546 <sup>.05</sup> ◇	.791 <sup>.02</sup> ◇	.582 <sup>.07</sup> ◇	11k
		N	.791 <sup>.01</sup> *	.700 <sup>.01</sup> *	.675 <sup>.02</sup> *	.758 <sup>.02</sup> *	.561 <sup>.02</sup> *	.791 <sup>.02</sup> *	.555 <sup>.07</sup> ◇	.776 <sup>.03</sup> *	.597 <sup>.05</sup> *	15k
OPT IML	FT	Y	.789 <sup>.01</sup> *	.681 <sup>.02</sup> *	.661 <sup>.02</sup> *	.720 <sup>.05</sup>	.516 <sup>.09</sup>	.789 <sup>.01</sup> *	.493 <sup>.05</sup>	.735 <sup>.04</sup> ◇	.579 <sup>.06</sup> ◇	15k
		N	.796 <sup>.01</sup> *	.690 <sup>.02</sup> *	.674 <sup>.03</sup> *	.738 <sup>.02</sup> *	.500 <sup>.07</sup>	.791 <sup>.02</sup> *	.488 <sup>.10</sup>	.723 <sup>.09</sup>	.593 <sup>.10</sup> ◇	15k
	FS	Y	.789 <sup>.01</sup> ◇	.698 <sup>.01</sup> *	.672 <sup>.02</sup> *	.757 <sup>.02</sup> *	.563 <sup>.03</sup> *	.798 <sup>.02</sup> *	.552 <sup>.07</sup> ◇	.780 <sup>.03</sup> *	.577 <sup>.07</sup>	11k
		N	.792 <sup>.01</sup> *	.699 <sup>.01</sup> *	.673 <sup>.02</sup> *	.755 <sup>.02</sup> *	.564 <sup>.03</sup> *	.795 <sup>.01</sup> *	.558 <sup>.06</sup> ◇	.772 <sup>.04</sup> *	.604 <sup>.05</sup> *	15k
T5	FT	Y	.792 <sup>.01</sup> *	.696 <sup>.02</sup> *	.667 <sup>.02</sup> *	.753 <sup>.02</sup> *	.567 <sup>.04</sup> *	.795 <sup>.02</sup> *	.566 <sup>.05</sup> ◇	.771 <sup>.03</sup> *	.584 <sup>.09</sup>	12k
		N	.789 <sup>.01</sup> *	.684 <sup>.01</sup> *	.660 <sup>.02</sup> *	.731 <sup>.03</sup> ◇	.536 <sup>.02</sup> *	.784 <sup>.01</sup> *	.523 <sup>.08</sup>	.748 <sup>.04</sup> *	.592 <sup>.07</sup> ◇	10k
	FS	Y	.786 <sup>.01</sup> ◇	.682 <sup>.02</sup> *	.674 <sup>.03</sup> *	.738 <sup>.02</sup> *	.500 <sup>.07</sup>	.791 <sup>.02</sup> *	.488 <sup>.10</sup>	.723 <sup>.09</sup>	.593 <sup>.10</sup> *	11k
		N	.798 <sup>.01</sup> *	.700 <sup>.02</sup> *	.666 <sup>.02</sup> *	.756 <sup>.02</sup> *	.559 <sup>.07</sup> ◇	.793 <sup>.01</sup> *	.573 <sup>.05</sup> ◇	.774 <sup>.03</sup> *	.596 <sup>.04</sup> *	15k
FLAN T5	FT	Y	.792 <sup>.01</sup> *	.696 <sup>.01</sup> *	.669 <sup>.01</sup> *	.752 <sup>.01</sup> *	.559 <sup>.03</sup> *	.792 <sup>.02</sup> *	.574 <sup>.05</sup> ◇	.767 <sup>.03</sup> *	.600 <sup>.07</sup> ◇	14k
		N	.793 <sup>.01</sup> *	.691 <sup>.01</sup> *	.672 <sup>.02</sup> *	.737 <sup>.03</sup> *	.544 <sup>.05</sup>	.790 <sup>.01</sup> *	.520 <sup>.08</sup>	.750 <sup>.04</sup> *	.597 <sup>.08</sup>	10k
	FS	Y	.786 <sup>.00</sup> *	.684 <sup>.01</sup> *	.651 <sup>.02</sup>	.743 <sup>.02</sup> *	.558 <sup>.04</sup> *	.778 <sup>.01</sup> *	.536 <sup>.04</sup>	.744 <sup>.04</sup> *	.590 <sup>.10</sup>	0.3k
		N	.774 <sup>.02</sup>	.662 <sup>.04</sup>	.637 <sup>.04</sup>	.709 <sup>.06</sup>	.509 <sup>.09</sup>	.765 <sup>.03</sup>	.490 <sup>.09</sup>	.724 <sup>.06</sup>	.583 <sup>.09</sup>	0.3k

Table 5.3: DeBERTa results (macro-F<sub>1</sub> and hate-class F<sub>1</sub>) with generative DA, averaged over 5 runs  $\pm std_{dev}$ , overall and by target (Gender, Race, Origin, Sexuality, Religion, Disability, and Age). Statistical significance is calculated against the *no augmentation* baseline. \*: highly statistically significant ( $\tau = 0.2$ ), ◇: statistically significant ( $\tau = 0.5$ ).  $n(h)$  = number of *hateful* synthetic examples preserved after filtering.

## 5.4 Results and Discussion

In this section we report the results of our experiments averaged across 5 data folds. We test statistical significance using Almost Stochastic Order (ASO) (Dror et al., 2019; Del Barrio et al., 2018), as implemented by Ulmer et al. (2022).

### 5.4.1 Generative DA

We report in Table 5.3 the results of our experiments using generative DA compared with EDA and the two baselines described above. The classification performance is evaluated globally in terms of macro-F<sub>1</sub>

and minority (*hate*) class  $F_1$  and for each target identity category as *hate* class  $F_1$ , so that the impact of synthetic data can be examined on a per-target basis.

Even from the *no augmentation* baseline, it is clear that performance can vary greatly across targets, with up to 27% hate- $F_1$  differences between them. In particular, the model appears to struggle with posts about *origin*, *religion*, and *age*, while, although underrepresented, posts about *disability* tend to be classified more accurately. This suggests that performance might also be influenced by factors other than the representation of targets in the dataset, such as how broad a target category is in terms of sub-groups. For instance, *origin* can include any type of discrimination based on geographical origin, including specific countries, and *religion* can encompass any type of religious discourse, although religions have been found to often be targeted through specific offense types (Ramponi et al., 2022). This makes classification challenging, especially for systems mostly relying on lexicon. This shows also how relevant it is to assess performance on targets separately, as examples referring to different target identity groups might pose different challenges for classification.

Most of the models trained on generation-augmented data outperform the *no augmentation* baseline across targets, with different improvements based on target identity group (*origin*, *religion*, and *age* in particular). Strikingly, however, EDA performs better than all generation-based DA configurations, regardless of prompting type or access to target information, for all targets but *age*. While performance gains are similar between EDA and the best generation-based setup compared to the baseline (+.026 and +.025 M- $F_1$  respectively), EDA appears to lead to slightly better performance in terms of minority class  $F_1$  (+.062 against +.048), at a small fraction of the computational cost of the generation-

based approaches. This is reflected on the performance per target identity, in which EDA outperforms generative DA across all but one target, *age*, the least represented one in the data. We hypothesize EDA is effective because small perturbations can make models more robust, especially with regards to the *hateful* class, while generative models might increase performance, but they might also be more likely to inject noise.

The impact of finetuning compared to few-shot prompting appears to be model-dependent, with differences across models also regarding the impact of target information. For all but OPT-IML, finetuning approaches tend to favor the inclusion of target information, albeit with relatively minor differences. Interestingly, the amount of synthetic examples labeled as *hateful* (reported in Table 5.3 as  $n(h)$ ) that pass filtering does not appear to strongly impact the performance of models trained on synthetic data, indicating that potentially even just a few hundred synthetic examples can positively impact generalization. This could also indicate that even just the addition of non-hateful synthetic examples might help models to generalize.

#### 5.4.2 Mixture of Generative DA and EDA

Since models trained on EDA-augmented data outperform models trained only on generation-augmented sequences, we also experiment with the mixture of the two methods, with 15k synthetic examples created using each of them. In Table 5.4 we report the results of these experiments.

Overall, it appears that the combination of EDA and generative DA can outperform each of the two methods separately, with some differences across models, augmentation setups, and target groups. The setup with EDA and the T5 model finetuned with target information leads to statistically significant hate- $F_1$  gains over EDA both overall and on the *gender*, *sexuality*, and *religion* targets. In addition, the classification of *ori-*



*gin*, *religion*, and *disability* improves by around or over 10% M-F<sub>1</sub> over the *no augmentation* baseline, showing the potential of this DA setup. However, the impact of finetuning vs. few-shot prompting still appears to be model-dependent, similarly to the impact of target identity information in the prompts.

In general, the high computational cost of generative approaches might not always justify their use against simpler yet effective DA approaches such as EDA in low-resource scenarios. Nevertheless, the combination of the two methods can outperform each method alone, and we hypothesize that it may be due to the fact that the gains are complementary: while EDA can make models robust to small perturbations such as word order changes, generative DA could be better at increasing lexical variety.

## 5.5 Qualitative Analysis

In this section, we look into the synthetically generated texts and the models trained on them from a qualitative point of view. First, we carry out a manual annotation on the generated texts to compare the different settings in terms of realism, target identity group assignment correctness and label consistency. Our goal is to assess whether these three dimensions in the generated data correlate with classifier performance. Then, we turn to the HateCheck tests (Sec. 3.6). Given the focus of our work, we analyse the out-of-distribution performance by target with HateCheck. More specifically, HateCheck targets do not exactly overlap with the target identity categories of the MHS dataset, thus providing a complementary view on our models' performance.

			M-F <sub>1</sub>	Hate-F <sub>1</sub>	Hate-F <sub>1</sub>						
					Gender	Race	Origin	Sexuality	Religion	Disability	Age
<b>No augmentation</b>			.773 <sup>.02</sup>	.652 <sup>.03</sup>	.635 <sup>.02</sup>	.696 <sup>.04</sup>	.497 <sup>.05</sup>	.756 <sup>.03</sup>	.485 <sup>.12</sup>	.698 <sup>.03</sup>	.545 <sup>.04</sup>
<b>EDA</b>			.799 <sup>.01</sup>	.714 <sup>.01</sup>	.687 <sup>.01</sup>	.771 <sup>.02</sup>	.582 <sup>.03</sup>	.806 <sup>.01</sup>	.601 <sup>.02</sup>	.799 <sup>.02</sup>	.589 <sup>.06</sup>
Model	Tar										
OPT + EDA	FT	Y	.777 <sup>.02</sup>	.698 <sup>.02</sup>	.679 <sup>.02</sup>	.759 <sup>.01</sup>	.567 <sup>.05</sup>	.795 <sup>.01</sup>	.593 <sup>.03</sup>	.801 <sup>.03</sup>	.578 <sup>.07</sup>
		N	.792 <sup>.02</sup>	.711 <sup>.02</sup>	.687 <sup>.02</sup>	.768 <sup>.02</sup>	.586 <sup>.06</sup>	.806 <sup>.01</sup>	.599 <sup>.03</sup>	<b>.812<sup>.02</sup>◇</b>	.588 <sup>.06</sup>
	FS	Y	.788 <sup>.02</sup>	.707 <sup>.01</sup>	.684 <sup>.01</sup>	.767 <sup>.01</sup>	.579 <sup>.04</sup>	.803 <sup>.01</sup>	.588 <sup>.03</sup>	.803 <sup>.02</sup>	.588 <sup>.06</sup>
		N	.795 <sup>.03</sup>	.715 <sup>.03</sup>	.689 <sup>.03</sup>	.774 <sup>.02</sup>	.595 <sup>.04</sup>	.808 <sup>.03</sup>	.617 <sup>.03</sup> ◇	.802 <sup>.04</sup>	<b>.631<sup>.07</sup>◇</b>
OPT-IML + EDA	FT	Y	.788 <sup>.01</sup>	.709 <sup>.01</sup>	.687 <sup>.02</sup>	.767 <sup>.01</sup>	.577 <sup>.04</sup>	.808 <sup>.02</sup>	.611 <sup>.01</sup> ◇	.797 <sup>.03</sup>	.584 <sup>.04</sup>
		N	.791 <sup>.01</sup>	.710 <sup>.01</sup>	.687 <sup>.01</sup>	.766 <sup>.01</sup>	.580 <sup>.05</sup>	.805 <sup>.02</sup>	.597 <sup>.03</sup>	.800 <sup>.01</sup>	.597 <sup>.05</sup>
	FS	Y	.789 <sup>.02</sup>	.709 <sup>.02</sup>	.685 <sup>.02</sup>	.769 <sup>.01</sup>	.584 <sup>.05</sup>	.805 <sup>.02</sup>	.615 <sup>.03</sup> ◇	.795 <sup>.04</sup>	.597 <sup>.06</sup>
		N	.791 <sup>.01</sup>	.711 <sup>.01</sup>	.689 <sup>.02</sup>	.766 <sup>.01</sup>	.588 <sup>.04</sup>	.809 <sup>.01</sup>	.601 <sup>.02</sup>	.807 <sup>.02</sup>	.596 <sup>.06</sup>
T5 + EDA	FT	Y	<b>.805<sup>.01</sup></b>	<b>.722<sup>.01</sup>◇</b>	.695 <sup>.01</sup> ◇	<b>.778<sup>.01</sup></b>	<b>.596<sup>.03</sup></b>	.815 <sup>.01</sup> ◇	<b>.628<sup>.03</sup>◇</b>	.808 <sup>.03</sup>	.588 <sup>.08</sup>
		N	.799 <sup>.00</sup>	.716 <sup>.01</sup>	<b>.696<sup>.01</sup>◇</b>	.772 <sup>.01</sup>	.589 <sup>.03</sup>	.810 <sup>.02</sup>	.610 <sup>.03</sup>	.800 <sup>.02</sup>	.617 <sup>.07</sup>
	FS	Y	.796 <sup>.01</sup>	.715 <sup>.01</sup>	.689 <sup>.02</sup>	.776 <sup>.01</sup>	.582 <sup>.04</sup>	.807 <sup>.02</sup>	.611 <sup>.03</sup>	.809 <sup>.02</sup> ◇	.618 <sup>.06</sup>
		N	.793 <sup>.01</sup>	.712 <sup>.01</sup>	.691 <sup>.02</sup>	.771 <sup>.01</sup>	.586 <sup>.04</sup>	.809 <sup>.01</sup>	.602 <sup>.01</sup>	.803 <sup>.02</sup>	<b>.619<sup>.03</sup>◇</b>
Flan-T5 + EDA	FT	Y	.803 <sup>.01</sup>	.718 <sup>.01</sup>	.690 <sup>.01</sup>	.774 <sup>.02</sup>	.586 <sup>.04</sup>	.813 <sup>.01</sup> ◇	.609 <sup>.04</sup>	.801 <sup>.03</sup>	.597 <sup>.07</sup>
		N	.794 <sup>.01</sup>	.712 <sup>.01</sup>	.690 <sup>.02</sup>	.766 <sup>.01</sup>	.585 <sup>.04</sup>	.811 <sup>.01</sup>	.600 <sup>.02</sup>	.794 <sup>.03</sup>	.582 <sup>.06</sup>
	FS	Y	.788 <sup>.02</sup>	.707 <sup>.02</sup>	.685 <sup>.03</sup>	.767 <sup>.02</sup>	.574 <sup>.05</sup>	.807 <sup>.01</sup>	.598 <sup>.03</sup>	.797 <sup>.03</sup>	.590 <sup>.06</sup>
		N	.799 <sup>.02</sup>	.718 <sup>.02</sup>	.691 <sup>.03</sup>	.777 <sup>.02</sup>	.582 <sup>.03</sup>	<b>.816<sup>.02</sup>◇</b>	.619 <sup>.03</sup> ◇	.810 <sup>.03</sup>	.615 <sup>.08</sup>

Table 5.4: DeBERTa results of generative DA + EDA overall and by target, averaged over 5 runs  $\pm stdev$ . Statistical significance is calculated against the results obtained with EDA. ◇: statistically significant against EDA alone ( $\tau = 0.5$ ).

### 5.5.1 Manual Annotation

A total of 1,120 generated texts filtered with DeBERTa were annotated by two annotators with a background in linguistics and experience in hate speech research. For each combination of finetuning/prompting/target presence (16 setups), they annotated 70 examples, evenly distributed across labels and, in the experiments that used them, targets.

In particular, for each setting for generative DA *with target information*, annotators were asked to annotate synthetic examples by specifying the following dimensions:

- *Label*: whether the content of the text is hateful or not,
- *Target correctness*: whether the target mentioned in the text matches with the target identity category given in input to the generative model,
- *Realism*: whether the message appears realistic and could have been plausibly written by a human.

For the examples generated *without access to target information*, the *Target correctness* was not annotated.

Consider for example the following sentence, generated giving ‘age’ as target information:

*‘Fuckin white men are trashy like a muthafucker’.*

In this case, *Label* would be ‘hateful’, *Realism* would be ‘Yes’ but *Target correctness* would be ‘No’, because the target identity category of the generated example should arguably be ‘race’ and not ‘age’. Inter-annotator agreement was calculated using Krippendorff’s alpha on 10% of the manually analyzed data. The annotators showed moderate agreement with regards to label correctness ( $\alpha = 0.76$ ), while the scores were

higher for target identity group matches ( $\alpha = 0.83$ ) and realism ( $\alpha = 0.82$ ).

Model	Target	Label	Target Match	Realism	M-F <sub>1</sub>	Hate F <sub>1</sub>	
OPT	FT	Y	93%	63%	66%	.783	.683
		N	N/A	/	0%	.774	.652
	FS	Y	90%	39%	83%	.782	.691
		N	81%	/	70%	.791	.700
OPT-IML	FT	Y	96%	53%	66%	.789	.681
		N	N/A	/	0%	.796	.690
	FS	Y	90%	57%	79%	.789	.698
		N	81%	/	73%	.792	.699
T5	FT	Y	83%	59%	80%	.792	.696
		N	74%	/	30%	.789	.684
	FS	Y	N/A	N/A	0%	.786	.682
		N	N/A	/	0%	.798	.700
Flan-T5	FT	Y	94%	66%	81%	.792	.696
		N	74%	/	41%	.793	.691
	FS	Y	89%	36%	84%	.786	.684
		N	87%	/	86%	.774	.662

Table 5.5: Generated texts labeled as correct by human annotators in terms of labels, target categories, and realism. N/A refers to cases in which all of the generated texts were nonsensical (0% realistic), with impossible assignment of labels or categories. We also report the model performance from Table 5.3 in terms of Macro-F<sub>1</sub> and Hate F<sub>1</sub>, in order to make comparisons between model performance and manual annotation results easier.

An overview of the manual annotations is reported in Table 5.5. In most cases, the addition of target information results in more realistic texts and, in general, more accurate label assignment by the generation model. However, this is not directly associated with the augmented data improving model performance when used for training. For instance, the setting that yields the best results with data generated by T5 (0.798 M-F<sub>1</sub> and 0.700 Hate-F<sub>1</sub>, see Table 5.3) is the one with few-shot prompting without target information. The texts generated by this model are, in fact, deemed as never realistic by the human annotators. On the other hand, the worst classification setting performance-wise is obtained with

examples generated by OPT using finetuning and no target information (0.774 M-F<sub>1</sub> and 0.652 Hate-F<sub>1</sub>), which led the model to generate nonsensical texts.

If we compare the behavior of the different generative models, we observe that Flan-T5 is the most consistent in terms of realistic generated text, being able to produce some realistic sentences in every setting, and obtaining the highest *Realism* score overall. OPT and OPT-IML, on the other hand, generate nonsensical texts when finetuned without target information, while T5 does not generate any realistic sentence when few-shot prompting is used, both with and without target information.

Overall, the rate of realistic texts and the accuracy of the identity categories are still somewhat low compared to the correctness of label assignment, showing that the generative models we tested might have difficulties dealing with more than one type of constraint/instruction. Indeed, while few-shot (FS) approaches tend to lead to more realistic generated sequences (aside from T5), this typically entails lower label correctness or target match, and vice-versa.

### 5.5.2 HateCheck Analysis

We perform a second qualitative analysis using the HateCheck test suite (Sec 3.6). We focus on the models trained with augmented data using generative DA + EDA for this analysis (Table 5.4), since they yield the best classification performance. Again, each generative model + EDA is used in four settings to generate new data: with finetuning or few-shot prompting, each one with or without target information.

All HateCheck test cases mention a specific target identity, to allow the exploration of unintended biases against different target groups. However, the target groups used in HateCheck do not fully overlap with the target identity groups in the MHS corpus (Figure 5.1). The target iden-

			Women	Trans p.	Gay p.	Black p.	Disabled p.	Muslims	Immigrants
No Augmentation			.142 <sup>.05</sup>	.101 <sup>.03</sup>	.252 <sup>.06</sup>	.216 <sup>.07</sup>	.113 <sup>.04</sup>	.147 <sup>.04</sup>	.109 <sup>.01</sup>
EDA			.400 <sup>.04</sup>	.485 <sup>.09</sup>	.590 <sup>.06</sup>	.643 <sup>.09</sup>	.463 <sup>.11</sup>	.546 <sup>.13</sup>	.420 <sup>.06</sup>
Model	Target								
OPT	FT	Y	<b>.458</b> <sup>.12</sup>	<b>.526</b> <sup>.12</sup>	<b>.646</b> <sup>.10</sup> ◇	<b>.671</b> <sup>.09</sup>	<b>.533</b> <sup>.13</sup> ◇	<b>.608</b> <sup>.16</sup>	<b>.529</b> <sup>.18</sup> ◇
		N	.354 <sup>.10</sup>	.394 <sup>.13</sup>	.537 <sup>.14</sup>	.581 <sup>.15</sup>	.372 <sup>.10</sup>	.538 <sup>.17</sup>	.402 <sup>.13</sup>
+ EDA	FS	Y	.384 <sup>.15</sup>	.412 <sup>.15</sup>	.552 <sup>.10</sup>	.605 <sup>.10</sup>	.408 <sup>.15</sup>	.511 <sup>.19</sup>	.411 <sup>.21</sup>
		N	.313 <sup>.10</sup>	.316 <sup>.10</sup>	.464 <sup>.16</sup>	.497 <sup>.13</sup>	.324 <sup>.10</sup>	.456 <sup>.16</sup>	.350 <sup>.14</sup>
OPT-IML	FT	Y	.409 <sup>.11</sup>	.468 <sup>.18</sup>	.583 <sup>.16</sup>	.612 <sup>.14</sup>	.493 <sup>.15</sup>	.572 <sup>.19</sup>	.488 <sup>.19</sup>
		N	.337 <sup>.09</sup>	.369 <sup>.14</sup>	.517 <sup>.16</sup>	.531 <sup>.19</sup>	.355 <sup>.14</sup>	.525 <sup>.20</sup>	.370 <sup>.16</sup>
+ EDA	FS	Y	.396 <sup>.14</sup>	.415 <sup>.12</sup>	.565 <sup>.06</sup>	.632 <sup>.06</sup>	.403 <sup>.14</sup>	.545 <sup>.15</sup>	.452 <sup>.18</sup>
		N	.324 <sup>.05</sup>	.315 <sup>.06</sup>	.436 <sup>.12</sup>	.527 <sup>.12</sup>	.321 <sup>.11</sup>	.415 <sup>.14</sup>	.308 <sup>.09</sup>
T5	FT	Y	.305 <sup>.05</sup>	.299 <sup>.12</sup>	.470 <sup>.13</sup>	.472 <sup>.11</sup>	.323 <sup>.11</sup>	.412 <sup>.06</sup>	.318 <sup>.08</sup>
		N	.273 <sup>.07</sup>	.273 <sup>.07</sup>	.502 <sup>.08</sup>	.518 <sup>.10</sup>	.309 <sup>.06</sup>	.417 <sup>.12</sup>	.303 <sup>.08</sup>
+ EDA	FS	Y	.357 <sup>.08</sup>	.382 <sup>.13</sup>	.518 <sup>.16</sup>	.547 <sup>.16</sup>	.341 <sup>.11</sup>	.527 <sup>.18</sup>	.388 <sup>.15</sup>
		N	.402 <sup>.13</sup>	.457 <sup>.16</sup>	.594 <sup>.14</sup>	.620 <sup>.14</sup>	.436 <sup>.18</sup>	.580 <sup>.18</sup>	.478 <sup>.18</sup>
Flan-T5	FT	Y	.287 <sup>.06</sup>	.257 <sup>.08</sup>	.447 <sup>.12</sup>	.454 <sup>.10</sup>	.254 <sup>.08</sup>	.436 <sup>.11</sup>	.294 <sup>.09</sup>
		N	.300 <sup>.05</sup>	.301 <sup>.08</sup>	.449 <sup>.13</sup>	.456 <sup>.12</sup>	.307 <sup>.09</sup>	.475 <sup>.16</sup>	.337 <sup>.11</sup>
+ EDA	FS	Y	.371 <sup>.13</sup>	.428 <sup>.09</sup>	.576 <sup>.10</sup>	.613 <sup>.11</sup>	.404 <sup>.10</sup>	.567 <sup>.15</sup>	.441 <sup>.13</sup>
		N	.388 <sup>.13</sup>	.411 <sup>.16</sup>	.556 <sup>.14</sup>	.593 <sup>.15</sup>	.399 <sup>.16</sup>	.543 <sup>.20</sup>	.422 <sup>.17</sup>

Table 5.6: DeBERTa results on HateCheck (hate-F<sub>1</sub>) by target identity, averaged across 5 runs. *p.* is an abbreviation for *people*. Statistical significance is calculated against the results obtained with EDA. ◇: statistically significant ( $\tau = 0.5$ ).

tities that are present in HateCheck are: *women* (which would fall under *gender* in MHS), *trans people* (*gender* in MHS), *gay people* (*sexuality* in MHS), *black people* (*race* in MHS), *disabled people* (*disability* in MHS), *Muslims* (*religion* in MHS) and *immigrants* (*origin* in MHS). The *age* category is present in MHS corpus and entirely missing in HateCheck.

We test the models trained on generation and EDA-augmented data (see setup in Section 5.4.2) on HateCheck targets, reporting the performance on a per-target basis in terms of hateful-class F<sub>1</sub>. The results are reported in Table 5.6, divided by the target identity categories used in HateCheck. We also report, as baselines, the results of the classification model using no augmentation and EDA alone.

Overall, we notice significant improvements over the *no augmenta-*

tion baseline across setups and target identities, although these improvements are variable. Indeed, EDA + OPT finetuned (FT) with target information lead to performance increases of up to .109 hate- $F_1$  over EDA alone, also resulting in more evenly distributed scores across targets, potentially showing less fairness-related weaknesses, paired with an improvement in performance across *all targets*. Interestingly, however, even overlapping targets between the MHS dataset and HateCheck show different improvements across generative DA setups. For instance, the HateCheck target *disabled people* apparently benefits from target information inclusion more than the *disability* category in the MHS corpus, suggesting that the inclusion of identity group information could affect performances differently in and out of domain.

Furthermore, HateCheck test instances are manually created starting from templates that include specific mentions of identity-group terms, which could lead to sentences that are very different from ‘real’ examples. For instance, one of the hateful templates used in HateCheck is ‘*I hate [IDENTITY]*’, in which *[IDENTITY]* is substituted with *women*, *disabled people*, and so on for all of the targets reported above. This can result in somewhat stereotypical examples, while hateful posts in the MHS corpus often contain slurs and offensive terms instead of neutral names to refer to people belonging to a certain group.

From the distribution of HateCheck scores, it appears that the OPT models tend to benefit from both finetuning and the addition of target identity group information, while T5 and Flan-T5 tend to produce texts that lead to the best gains when prompted few-shot. The influence of target information insertion has different effects on distinct target identity groups. The effect of finetuning, prompting, and target information therefore seems to depend more on the type or architecture of the generative model rather than whether it is instruction-tuned or not, as

OPT-IML and Flan-T5, which are both instruction-tuned, exhibit different trends.

Looking specifically at the HateCheck counterparts of the targets that are least represented in the MHS corpus (*disabled people* for *disability* and *Muslims* for *religion*), it is clear that data augmentation with target information tends to lead to consistently better results than the non-target-aware data augmentation approach with OPT and OPT-IML, while for both T5 models this is not always the case. This might indicate that, as previously discussed, models might struggle with multiple constraints when generating new examples.

Finally, in terms of fairness, it appears that data augmentation using generative models and EDA can work towards improving the performance of models on all targets included in HateCheck, even if there is no 1:1 mapping with the original targets. This indicates that this approach can potentially be effective in improving the performance of models across different targets of hate.

## 5.6 Conclusions on Fairness

In this chapter, we have investigated the impact of data augmentation with generative models on specific targets of hate, experimenting with instruction-finetuned models and the addition of target information when generating new sequences, with the goal of exploring the *fairness* implications of data augmentation approaches for hate speech detection.

It appears that DA methods have different types of impact on different targets, but they can improve performance for scarcely represented identity categories (**RQ1.3.1**). We observed that generative data augmentation alone is not as strong as simpler methods such as EDA, both globally and on a per-target basis, especially given that generative DA



is highly computationally expensive. However, their combination can lead to models that are more robust, especially for more scarcely represented target identities, highlighting the potential of this type of approach (**RQ1.3.3**). Through a qualitative analysis, we also emphasized the fact that including target information when generating synthetic examples can facilitate the creation of examples that are more realistic and exhibit more correct label assignments (**RQ1.3.2**), although these characteristics do not directly correlate with downstream task performance. One hypothesis to explain this is that the generation process could produce sequences that are so different from the distributions of the datasets we test our models on, that the test-set performance of models is negatively affected in spite of the examples being more realistic and correct.

Overall, our analysis shows that there is potential in data augmentation with regards to *fairness*, intended as group fairness, implying independence between model classification output and sensitive attributes (Anthis et al., 2024). However, although potentially useful, this type of DA can still lead to unpredictable results, and it is not guaranteed to always improve the performance of models across all identity groups with regards to hate speech.



# Chapter 6

## Privacy

While in Chapters 4 and 5 we experimented with synthetic data setups in which we started with small amounts of gold data and augmented it in order to obtain *more data* for various reasons, in this Chapter we move onto a setup in which generative data augmentation is desirable and potentially useful for *privacy* reasons, with the goal of substituting the gold data entirely (**RQ2.1**). Privacy, as a matter of fact, is one of the main reasons why synthetic data is used in a variety of sensitive applications of machine learning, such as healthcare and law (Jordon et al., 2022; Whitney and Norman, 2024). Datasets created to train abusive language detection systems deserve particular attention with regards to privacy, as they could be maliciously employed to profile users and target them. As discussed in Jahan and Oussalah (2023), even when such datasets do not contain user information, a search engine could be straightforwardly used to trace back the person who posted a certain message, nullifying traditional anonymization efforts such as the removal of user mentions.

More and more restrictions now being set that limit resharing of social media data even for research purposes (e.g. the 2023 changes in X/Twitter terms of use). For instance, in several jurisdictions around the world, social media users should be granted the so-called ‘right to be forgotten’

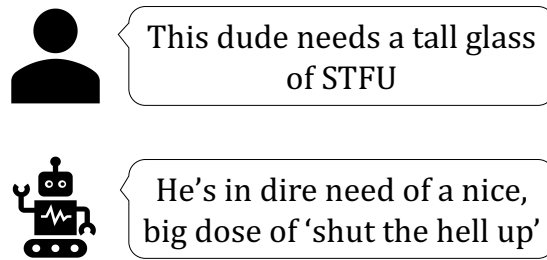


Figure 6.1: Original example and its corresponding synthetic rewriting.

or ‘right to erasure’.<sup>1</sup>

In the light of these restrictions, the availability of data to pursue research on abusive language detection and content moderation might represent an issue in the future. We therefore address the following question in this chapter: *would it be possible to replace existing datasets for abusive language detection with synthetic ones by maintaining the same classification performance?* Performing this task with a good accuracy would present several advantages. For example, it would potentially be possible to freely share datasets without the risk of disclosing user information or infringing terms of use and regulations if the synthetic data is created carefully in order to guarantee privacy.<sup>2</sup> Furthermore, being able to share datasets that do not directly contain user-created content could mitigate the problem of data degradation, which makes social media datasets unusable after few years from release, since hateful content is frequently deleted, as we discussed in Section 2.1.3. However, as it is not guaranteed that the synthetic data cannot actually be traced to the original posts that were used to create it, it is also important to investigate

<sup>1</sup><https://gdpr.eu/right-to-be-forgotten/>

<sup>2</sup>As noted by Jordon et al. (2022), synthetic data is not necessarily inherently private. While the focus of this chapter is not on how models can leak personal data from their pretraining into the synthetic data, that is a scenario that can actually occur in similar setups. In this respect, it would be interesting to investigate in the future a possible integration between our approach and differentially private models (Yu et al., 2022; Matzken et al., 2023) to mitigate this kind of risk.

potential failures of this type of approach.

In this chapter, we present a first set of experiments in this direction by using a generative large language model to rewrite two different abusive language datasets, comparing two prompt types. An example of a *rewriting* is shown in Figure 6.1. We then evaluate the quality of the generated data by using it for training abusive language detection classifiers, as well as by manually inspecting it and trying to reverse-search the original social media posts our synthetic data derives from.

## 6.1 Methods

As seen in Section 2.2.3, previous works experimenting with generative LLMs to augment abusive language datasets mostly exploit approaches such as fine-tuning of generative models on existing gold data (Anaby-Tavor et al., 2020; Kumar et al., 2020), trainable components for task-specific decoding (Hartvigsen et al., 2022), or humans in the loop to evaluate generated sequences (Fantón et al., 2021; Chung et al., 2023). However, with the growing size of generative LLMs, making them more expensive to fine-tune, and their zero-shot capabilities thanks to instruction tuning, these models can often carry out numerous tasks without requiring any further fine-tuning (Wei et al., 2022). Because of this, for the experiments of this chapter we use a freely available instruction-tuned model, Llama-2 chat 7B (Touvron et al., 2023), through the HuggingFace library (Wolf et al., 2020).<sup>3</sup>

The now widespread use of instruction-tuned generative large language models has also led to numerous efforts towards *alignment*, ideally in order to minimize inappropriate, offensive or unethical uses (Rao et al., 2023). While this is often preferable for many applications, it can

---

<sup>3</sup><https://huggingface.co/meta-llama/Llama-2-7b-hf>

make the creation of synthetic abusive language detection datasets complex, as these models are tuned to avoid producing abusive content due to its potentially harmful uses. Because of this, we frame the task not as the *generation* of new, unseen data, but rather as the *rewriting* of existing gold sequences, so that *i*) the synthetic sequences are semantically close to existing data, inspired by the simple changes applied by Easy Data Augmentation (Wei and Zou, 2019), and *ii*) the synthetic sequences cannot in principle be traced back to any existing social media posts or their original posters.

In this chapter, we experiment with the creation of synthetic data starting from two different English datasets: the **MDA** dataset (Sec. 3.5) and the **MHS** corpus (Sec. 3.4). We then test models on both datasets to evaluate cross-dataset robustness (**RQ1.2**). Furthermore, we again test our models with the **HateCheck** test suite (Sec. 3.6).

We use the default train-dev-test splits of the MDA dataset. We instead use a subset of the MHS dataset, in order to have a comparable size with the other dataset we use, selecting 10% of the dataset as the test set (3,524), 5% of the remaining examples as development data (1,586 examples) and finally 10% of the remaining texts as the training data (3,013 posts). The random seed we use for splitting<sup>4</sup> and shuffling the data for all datasets is 2023.

### 6.1.1 Rewriting Original Examples

We frame our rewriting task as two established NLP tasks: **paraphrasing** and **formality style transfer**, with the aim of forcing rewriting to be informal, so that we counteract the tendency of aligned models to use language that is as neutral as possible.

---

<sup>4</sup>With the `train_test_split` function from `sklearn.model_selection`.

In order to maximize our chances of producing at least one ‘valid’ sequence corresponding to each gold example, we produce multiple synthetic sequences. In particular, starting from each original text, we prompt Llama-2 with three different prompts for each task type, giving the model the original text and instructing it to reword it.

For the **paraphrasing** task framing, we use the following prompting templates:

1. *Paraphrase this text: “text”*  
*Paraphrased text: “*
2. *Reword this text, preserving meaning and tone: “text”*  
*Reworded text: “*
3. *Rewrite this message keeping the same meaning: “text”*  
*Reworded text: “*

For the **formality style transfer** task framing we use the following templates for prompting:

1. *Paraphrase this text in a more informal way: “text”*  
*Paraphrased text: “*
2. *Reword this text, preserving meaning and tone but using more informal language: “text”*  
*Reworded text: “*
3. *Rewrite this message more informally, keeping the same meaning: “text”*  
*Reworded text: “*

In addition to using 3 different prompts for each task framing, we run each prompt 3 times for each corresponding gold example, resulting in at least 9 synthetic texts for each source text for each prompt type.<sup>5</sup>

---

<sup>5</sup>In some cases the model will continue producing paraphrases until it hits our maximum length, often resulting in more than 9 synthetic sequences being produced.

For generation, we use  $top-p = 0.9$ , a temperature of 1.0, and we set the minimum and maximum lengths of the generated sequences to 3 and 500, respectively.<sup>6</sup>

### 6.1.2 Filtering

Since we aim at obtaining synthetic data that *i*) cannot be reconnected to its source text but *ii*) preserves the original labels of the source data, we perform two filtering steps. First, we discard the synthetic sequences that are *verbatim* or extremely similar repetitions of the original gold texts using the TheFuzz library<sup>7</sup>, a Levenshtein distance-based tool to calculate string similarity. Then, we further filter the synthetic sequences using a classifier, similarly to the filtering step shown in Section 5.3, to minimize issues with data preservation (Kumar et al., 2020). In particular, we train a Roberta large classifier (Liu et al., 2019) on the original gold data, and then use this classifier to infer the class of the synthetic instances. We discard all sequences for which the predicted label of the synthetic text does not match the label of the original text used to create it. Finally, out of all the remaining sequences, we pick a random one to use as the synthetic equivalent of the original text. If for a given original example no synthetic texts pass the filtering stage, we move onto the next example, *de facto* discarding that text sequence. As a result, the size of the synthetic dataset tends to be smaller than the original gold one. The total number of synthetic texts that pass filtering for each prompting type is reported in Table 6.1, in the  $n(train)$  column.

Out of the synthetic texts that do not pass the filtering step, an overwhelming majority of them (between 95 and 98% across both datasets

---

<sup>6</sup>The remaining hyperparameters we use are the default ones of the GenerationConfig HuggingFace class.

<sup>7</sup><https://github.com/seatgeek/thefuzz>. We discard all sequences scoring over 75 in terms of similarity with their original counterparts.



and prompting strategies) does not make it due to inconsistent label assignment, i.e. the classifier predicted a different label for the synthetic text than the label of the original. The remaining texts that are discarded during filtering are mostly almost-exact matches with the original data, and instances that do not pass either filter are extremely rare.

## 6.2 Evaluation

Since our main focus in this chapter is on the possibility to replace abusive language datasets with synthetic data by maintaining the same performance level, we train classifiers on synthetic data derived from MDA and MHS, and evaluate them on the gold test sets of both, including cross-dataset testing to assess model robustness. Additionally, we test our models with the HateCheck test suite and perform a manual analysis. For our classification experiments, we fine-tune a Roberta large classifier (355M parameters) (Liu et al., 2019) on the original and synthetic data for both gold datasets and for both prompting types. We select this model because it was the best performing one on the MHS dataset, as reported in Kennedy et al. (2020), and it was reported to outperform BERT on the MDA dataset (Leonardelli et al., 2021). Furthermore, in Chapter 4, we found no relevant differences among BERT-like models with respect to the impact of synthetic data on model performance for abusive language detection.

Similarly to the previous chapters, we use the HuggingFace library for all model implementations. For the RoBERTa classifiers, we use the default hyperparameters of the TrainingArgs class, setting batch size to 16, the maximum sequence length to 150, and the learning rate to 5e-6. For the models trained on the original data, we run training for 10 epochs, selecting the epoch with the lowest validation loss. After a

manual analysis of the best epochs in most of the runs with the original data, we pick 3 epochs for training the models on synthetic data, as using synthetic development data would be misleading, and we aimed at not using any gold data during the training phase. Generation took about 180 hours and training of classifiers took about 10 hours on a Nvidia A40 GPU.

### 6.2.1 Classification Results

The classification results of models trained on synthetic data are reported in Table 6.1. We report the mean macro- $F_1$  and abusive-class  $F_1$  across 5 runs with different data shuffles and different model initializations, as well as the standard deviation across runs.

We observe that models trained on synthetic data *tend to perform similarly to the models trained on the original gold data, in some cases even with mild improvements*. This is in contrast with previous findings showing that synthetic data are generally not very helpful for subjective tasks such as this one (Li et al., 2023). The difference with Li et al. (2023) might be due to the fact that they frame the creation of synthetic data as generation, not rewriting of existing examples, and they do not carry out any filtering on the artificial texts. Furthermore, our models based on synthetic data perform surprisingly well even if the training set size is smaller than the gold one. In addition, we observe improvements with regards to cross-dataset performance, especially in the case of the synthetic data produced starting from the MHS corpus. Indeed, the model trained on data rewritten starting from MHS data yields an improvement over training using gold data when tested both on MDA and HateCheck, with up to 16 abusive-class  $F_1$  points on the former.

These results suggest that synthetic data can potentially improve robustness in out-of-distribution scenarios (**RQ2.1**), probably because lex-

Test data →		MDA		MHS		HateCheck		
Training data ↓	$n(\text{train})$	M- $F_1$	Ab- $F_1$	M- $F_1$	Ab- $F_1$	M- $F_1$	Ab- $F_1$	
MDA	Original gold	2,161	0.779 $\pm$ .009	<b>0.720</b> $\pm$ .008	0.661 $\pm$ .011	0.595 $\pm$ .007	<b>0.519</b> $\pm$ .021	<b>0.573</b> $\pm$ .033
	Synth: Paraphrase	1,444	0.779 $\pm$ .013	0.706 $\pm$ .022	0.680 $\pm$ .005	0.607 $\pm$ .004	0.508 $\pm$ .009	0.552 $\pm$ .016
	Synth: Formality	1,557	<b>0.783</b> $\pm$ .003	0.713 $\pm$ .004	<b>0.684</b> $\pm$ .007	<b>0.611</b> $\pm$ .005	0.470 $\pm$ .015	0.490 $\pm$ .029
MHS	Original gold	3,013	0.540 $\pm$ .034	0.260 $\pm$ .063	0.791 $\pm$ .006	0.688 $\pm$ .010	0.338 $\pm$ .029	0.206 $\pm$ .049
	Synth: Paraphrase	2,435	<b>0.629</b> $\pm$ .014	<b>0.423</b> $\pm$ .025	0.787 $\pm$ .005	0.694 $\pm$ .008	0.351 $\pm$ .025	0.236 $\pm$ .051
	Synth: Formality	2,587	0.606 $\pm$ .019	0.381 $\pm$ .035	<b>0.793</b> $\pm$ .003	<b>0.697</b> $\pm$ .003	<b>0.359</b> $\pm$ .008	<b>0.255</b> $\pm$ .017

Table 6.1: Average results over 5 runs in terms of Macro- $F_1$  ( $M-F_1$ ) and Abusive-class  $F_1$  ( $Ab-F_1$ )  $\pm$  stdev. Grey cells denote out of distribution / cross-dataset performance. The  $n(\text{train})$  column indicates the number of initial examples for gold data and the number of synthetic instances that passed filtering and are therefore used for training.

ical cues specific to the training data may be removed through rewriting, allowing models to achieve better generalization capabilities. This would explain why cross-dataset improvement is particularly evident when testing on the MDA dataset, which deals with three specific topics, while MHS has a broader coverage in terms of domains and hate targets. Similarly, with regards to Hatecheck, we observe improvements for the MHS-derived synthetic data over the original data, while the synthetic data derived from MDA appears to not generalize as well. Overall, neither prompting type appears to clearly outperform the other.

## 6.2.2 Qualitative Analysis

In this section, we examine the generated data from a qualitative point of view. First, we carry out an analysis focused on lexicon. Then, we manually inspect the synthetic texts to check if they can be traced back to the original posts they derive from.

To analyze differences between original and synthetic data, we compare their lexical diversity using Type Token Ratio (TTR) and Measure of Textual Lexical Diversity (MTLD) (McCarthy, 2005), which are cal-

		TTR	MTLD
MDA	Original Gold	0.86	52.37
	Synth: Paraphrase	0.88	59.86
	Synth: Formality	0.88	69.70
MHS	Original Gold	0.88	52.08
	Synth: Paraphrase	0.87	65.18
	Synth: Formality	0.87	66.46

Table 6.2: Lexical diversity measures on the original and synthetic data for both datasets.

culated using Variationist (TTR; Appendix A) and the TAALED library<sup>8</sup> (MTLD) on texts tokenized with Spacy.<sup>9</sup> Results are reported in Table 6.2. While TTR is comparable on all datasets, generally indicating high lexical variability, MTLD shows a difference between gold and synthetic data. Indeed, synthetic data exhibits a higher degree of lexical diversity, especially if generated with the *formality* prompt type. The different output between TTR and MTLD may be due to the fact that the latter is more robust with regards to text length variations (Fergadiotis et al., 2015).

In addition to quantitatively measuring lexical diversity, we also inspect the data for any lexical cues that might influence generalization. Using the Variationist tool, we calculate `npw_relevance`, a normalized class relevance metric based on PMI as seen in Ramponi and Tonelli (2022). The 10 most relevant tokens for the *abusive* class in gold data and in the synthetic data created using both prompt formats is shown in Table 6.3.

The first aspect that emerges from the analysis of Table 6.3 is that original gold data for both datasets contains more profanities, while synthetic data overall appears to contain less swear words, with terms such

<sup>8</sup><https://pypi.org/project/taaled/>

<sup>9</sup><https://spacy.io/>

Top 10 most relevant tokens for the abusive class					
MHS			MDA		
Original Gold	Synth: Paraphrase	Synth: Formality	Original Gold	Synth: Paraphrase	Synth: Formality
b*tch	people	dude	CHEATER	people	Hey
fucking	f*ggot	Ugh	shit	Trump	Yo
ass	ass	b*tch	Fuck	COVID	😂
fuck	individual	Hey	ass	individual	Dude
f*ggot	term	total	Trump	person	dude
cock	worthless	gonna	people	19	people
n*gga	individuals	Dude	covid	individuals	Trump
cum	language	dudes	fucking	Biden	total
black	time	Yo	stupid	Joe	gonna
shit	person	real	piece	actions	Covid

Table 6.3: Top 10 most relevant tokens for the abusive class in the original gold data and in the synthetic data for both datasets, as calculated using the `npw.relevance` metric in Variationist.

as *individuals* or *people* being highly relevant for the *abusive* class instead. This seems to support the hypothesis that synthetic data might change the level of reliance of models on certain lexical cues, decreasing the relevance of some terms and increasing the relevance of others. However, strikingly, some slurs, such as *f\*ggot* and *b\*tch* still remain relevant for the *abusive* class even in the synthetic data for the MHS corpus. A possible explanation for this is that the alignment process of models can interfere with the rewriting process, possibly because models are fine-tuned to avoid using certain terms. For instance, the very common swear word *fucking* stops being one of the most relevant ones for the *abusive* class in the synthetic data, as well as any references to skin color. Our hypothesis regarding this is that the alignment of models was focused on the most common profanities and targets of hate. We will discuss qualitative aspects regarding the presence of certain terms and identity groups more in depth in Chapter 7.

We finally perform a manual inspection of the generated data, to assess whether the synthetic data can be traced back to the original post through online search. We provide an annotator with 200 synthetic examples created starting from data from the MDA dataset, divided equally between the prompting types.<sup>10</sup> Out of the original posts that are still online (around 50%), 60% were found by the annotator through a search engine by typing the text of the message into a search engine. In this percentage, we consider as found all posts for which the annotator was unambiguously able to find the original post. In some cases, for example with very generic posts such as ‘*Wear A Freakin’ Mask!!*’, they would find too many results and not be able to tell which one was the original post. Strikingly, however, *none of the original posts could be found starting from their synthetic counterparts*, showing the potential of this type of approach for data anonymization.

### 6.3 Conclusions on Privacy

In this chapter, we carried out an exploration of abusive language detection using synthetic data generated through rewriting, with the goal of investigating whether classification models fine-tuned entirely on synthetic data can be effective for the task of abusive language detection when datasets cannot be freely shared due to *privacy* reasons (**RQ2.1**). We show that this is a promising research direction, since models trained on synthetic data can achieve classification performance on par with models based on gold data, and even show better robustness in some cross-dataset settings. Furthermore, it was not possible to trace back the original data starting from the synthetic examples, even through a man-

---

<sup>10</sup>We only manually analyze MDA-derived data since we have the original Tweet IDs of the messages, to effectively check whether the original messages still exist online.

ual search online. Rewriting of original texts seems to be a promising strategy both through paraphrasing and formality style transfer. We believe this approach to be a step forward for the development of datasets and systems for subjective tasks that are more privacy-aware and compliant with existing regulations on personal data sharing, anonymization, and right to be forgotten, although further work should be carried out on the topic, to understand the role of large language models in potentially mitigating privacy issues. In the next Chapter, we will move on to a more qualitative exploration of whether synthetic data can actually emulate the characteristics of real data.





# Chapter 7

## Realism and Quality

In this chapter, we move on to a more qualitative look at LLM-augmented data, focusing on how *realistic* and representative of real gold data synthetic texts can be (**RQ2.2**), considering also the hidden *risks* it may present (**RQ2.3**). Indeed, beside extrinsic evaluations, little attention has been paid in previous work to the risks and qualitative implications of employing synthetic data in sensitive tasks like hate speech detection, which as we have seen in the previous chapters can lead to mixed results.

In our *realism* experiments of this chapter, we address a scenario in which *one may need to perform hate speech detection on unseen data, and they would like to exploit the potential of generative LLMs and existing hate speech datasets*. What potential advantages can synthetic data offer in this respect? What are the risks associated with using LLMs for the task? Could generated data amplify bias or harm? As a first exploration in this direction, similarly to the experimental setup of Chapter 6, we focus on hate speech detection assuming enough several LLMs and hate speech datasets are already available (Poletto et al., 2021).

In particular, we aim at evaluating whether training on synthetic data created through paraphrasing can lead to better performance than using

existing hate speech data on out of distribution setups, similarly to the experiments of Chapter 6. However, in this chapter, we experiment with different generative models rather than with different types of prompts, as we saw in the previous chapter that there were no stark differences between prompting layouts.

The computational experiments are then paired with the main contribution of this chapter: a thorough manual analysis of the generated data, assessing fluency, grammaticality and ‘artificiality’. Given that biases may affect specific targets of hate differently (Sap et al., 2019, 2020), we devote particular attention to a per-target analysis, showing the effects of the usage of LLMs to produce synthetic data on target identity distribution, and subsequently its impact on fairness (Q1.3), further investigating the aspects we already discussed in Section 5 related to this topic from a qualitative point of view.

Since generated data is increasingly being used even for extremely sensitive applications (Ghanadian et al., 2024, for instance), it is important that the NLP community critically addresses the impact of synthetic data including ethical risks, along the line of similar discussions in other research communities (Whitney and Norman, 2024). In this chapter we propose an initial contribution in this direction.

## 7.1 Synthetic Data Generation

To be able to analyze the extrinsic impact on performance and the intrinsic characteristics of synthetic data for hate speech detection, as with the previous chapters, we first artificially create training data. For this, we use a similar generation approach to to the experiments of Chapter 6, since in this case our focus is more on the analysis of the synthetic data rather than on devising a better generation method. We therefore again

use an experimental setup in which we leverage existing hate speech resources by casting data augmentation as *paraphrasing* rather than as zero-shot generation, which allows us also to potentially mitigate effects related to model *alignment*, with LLMs often being programmatically blocked in generating hateful messages.

For investigating the effects that synthetic data can have on hate speech detection, we choose the Measuring Hate Speech (MHS, Sec. 3.4) corpus as the starting point for the augmentation, since it covers different target identity categories. We then test models on the Multi-Domain Agreement dataset (MDA, Sec. 3.5) and on HateCheck (Sec. 3.6). We expect the output text to *i)* be similar to the original social media post, *ii)* reflect the same hate speech label, and *iii)* preserve roughly the same meaning and topic. We analyze these aspects in our human evaluation in Section 7.3.

Our synthetic data creation pipeline consists of two steps. First, we prompt the models (Sec. 7.1.1) to obtain synthetic versions of the real data in the MHS corpus, creating one artificial counterpart for each example in the dataset, similarly to the experiments of Chapter 6. After extracting the paraphrased text from the model output, we perform two additional filtering steps on the synthetic sequences (Sec. 7.1.2).

We use three freely available and widely used generative models for our experiments, to favor comparability and reproducibility, through the HuggingFace library (Wolf et al., 2020): Llama-2 Chat 7B<sup>1</sup>, Mistral 7B Instruct v0.2<sup>2</sup>, and Mixtral 8x7B Instruct v0.1.<sup>3</sup> We load all the models in 4 bits, and as hyperparameters for generation we use *top-p* decoding = 0.9 and we set the minimum and maximum lengths of the

---

<sup>1</sup>[huggingface.co/meta-llama/Llama-2-7b-chat-hf](https://huggingface.co/meta-llama/Llama-2-7b-chat-hf)

<sup>2</sup>[huggingface.co/mistralai/Mistral-7B-Instruct-v0.2](https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2)

<sup>3</sup>[huggingface.co/mistralai/Mixtral-8x7B-Instruct-v0.1](https://huggingface.co/mistralai/Mixtral-8x7B-Instruct-v0.1)

generated sequences to 5 and 300, respectively.<sup>4</sup>

### 7.1.1 Prompting

We frame synthetic data creation as paraphrasing, similarly to one of the prompts we experimented with in Chapter 6, as it is a common task in instruction tuning datasets that are widely used for training LLMs (Wang et al., 2022; Wei et al., 2022) and thus it does not necessarily require fine-tuning or detailed prompting. Given a text, we prompt the models with the following template:

Paraphrase this text: “{text}”

Paraphrased text: “

For Mistral and Mixtral, the template is preceded and followed by the [INST] and [/INST] tags. We then extract, using a regular expression, the first text sequence after ‘Paraphrased text:’ that is between inverted commas in the model output.

### 7.1.2 Filtering

As we have already observed in Chapter 6, in a limited number of cases synthetic examples are almost identical to the original text they (should) paraphrase. We thus carry out fuzzy matching using the `thefuzz` library<sup>5</sup> to discard sequences that are verbatim copies of the original gold data. After some manual checks, following the same process as Section 6.1.2, we again set the similarity threshold for discarding sequences that are too similar to 75.

---

<sup>4</sup>The remaining hyperparameters we use are the default ones of the `GenerationConfig` HuggingFace class.

<sup>5</sup>[pypi.org/project/thefuzz](https://pypi.org/project/thefuzz)

In addition, as we have already seen and experimented with in the previous chapters, in previous work a further *filtering* step is typically employed, in which the generated sequences are re-labeled using a classifier to increase the chance that the label assignment of the synthetic texts is correct.

We aim at exploring the actual impact of this step, so we divide our experimental setups into:

- **No classifier filtering**, in which we preserve all synthetically created texts that passed the fuzzy matching step;
- **Classifier filtering**, in which we discard all the synthetic examples for which a classifier trained on gold data predicts a different label from the one that was assigned to the gold example the synthetic text derives from.

## 7.2 Extrinsic Evaluation

Before moving on to the qualitative analysis, we analyze the *extrinsic* impact of synthetic data by fine-tuning classifiers on both artificial and original data. This analysis contextualizes the main contribution of this chapter, namely the *intrinsic* evaluation of synthetic data (Section 7.3). Following the same thread as the previous chapters but with multiple state-of-the-art generative models, this analysis is aimed at addressing the following question: *What is the usefulness of synthetic data for the downstream task of hate speech detection with regards to performance?*

For the classification experiments, we use three pre-trained classifiers: RoBERTa Large<sup>6</sup> (355M parameters), RoBERTa Base<sup>7</sup> (125M parameters),

---

<sup>6</sup>[huggingface.co/FacebookAI/roberta-large](https://huggingface.co/FacebookAI/roberta-large)

<sup>7</sup>[huggingface.co/FacebookAI/roberta-base](https://huggingface.co/FacebookAI/roberta-base)

and DeBERTa Base<sup>8</sup> (140M parameters), for which we use the default hyperparameters of the `TrainingArgs` class, setting batch size to 64, the maximum sequence length to 150, and the learning rate to 5e-6. We train models for 3 epochs, similarly to the experiments of Chapter 6.

We again compare the performance of a model trained on original gold data with the performance of the same model trained on synthetic data only, in order to assess how effectively the synthetic data can mimic the gold training data.

In addition, we also test models trained only on data augmented using EDA (Wei and Zou, 2019), since we found it to be a competitive DA technique in Chapter 5, as well as to facilitate comparisons of the results shown in this chapter with those of Chapter 5. However, EDA is used in these experiments only for comparison purposes, as the fundamental assumption behind the perturbations used by EDA is that they do not alter the original text enough to be considered fully different examples. Hence, the usefulness of EDA-based approaches is heavily reduced in privacy-focused applications of synthetic data, in which it is important that the original sequences cannot be retrieved. The EDA setup reported in this section consists of perturbing each of the original examples from the MHS corpus with one of *synonym replacement*, *random insertion*, *random swap*, or *random deletion*, chosen randomly for each example. We do not perform any filtering on the EDA-generated sequences, as they would not pass our fuzzy match filter due to their extreme similarity to the original sequences.

While the classifiers are always trained on data (original or synthetic) from MHS, they are tested on the test splits of all datasets (MHS, MDA, and HateCheck), in order to assess both their in-distribution and their out-of-distribution performance. The metrics we use for evaluating clas-

---

<sup>8</sup>[huggingface.co/microsoft/deberta-v3-base](https://huggingface.co/microsoft/deberta-v3-base)

sifiers are macro- $F_1$  and minority class (*hate*)  $F_1$ .

Table 7.1 reports our experimental results with the Roberta Large model, averaged across 5 runs with different data shuffles and model initializations. The results of RoBERTa Base models are reported in Table 7.2 and those of DeBERTa Base in Table 7.3. Overall, the three models do not show remarkable differences among them in terms of performance, so we will mostly comment on the Roberta Large results for brevity.

				Test data			
				MHS		MDA	HateCheck
		$n(\text{train})$	% <i>hateful</i>	M- $F_1$	Hate $F_1$	M- $F_1$	M- $F_1$
Original gold data (MHS)		30,132	26%	<b>.811</b> $\pm$ .004	<b>.718</b> $\pm$ .008	.507 $\pm$ .027	.386 $\pm$ .026
EDA		30,132	26%	<b>.813</b> $\pm$ .003	<b>.723</b> $\pm$ .005	.531 $\pm$ .010	.405 $\pm$ .027
Gen. Model	Filter						
Llama-2 Chat 7B	No	28,289	26%	.769 $\pm$ .004	.680 $\pm$ .003	.675 $\pm$ .009	.603 $\pm$ .021
	Yes	20,187	2%	.805 $\pm$ .002	.715 $\pm$ .002	.539 $\pm$ .008	.346 $\pm$ .009
Mistral 7B Instruct	No	29,344	26%	.772 $\pm$ .004	.686 $\pm$ .003	.684 $\pm$ .007	<b>.665</b> $\pm$ .017
	Yes	22,483	4%	.808 $\pm$ .003	.716 $\pm$ .004	.526 $\pm$ .011	.371 $\pm$ .012
Mixtral 8x7B Instruct	No	29,351	26%	.754 $\pm$ .004	.670 $\pm$ .003	<b>.687</b> $\pm$ .005	<b>.665</b> $\pm$ .005
	Yes	22,370	3%	.802 $\pm$ .002	.706 $\pm$ .003	.525 $\pm$ .016	.364 $\pm$ .012

Table 7.1: Results of Roberta Large models trained on synthetic data only (average of 5 runs  $\pm$  stdev). Grey cells indicate out-of-distribution performance. *Filter:No* means that only paraphrased sequences too similar to the original ones and ill-formatted texts were discarded. *Filter:Yes* means that *classifier filtering* was applied.

The amount of training data for synthetic setups is lower than the amount of gold data due to the filtering step being applied to all synthetic sequences (Sec. 7.1.2). Specifically, in the ‘*no classifier filtering*’ setups (*Filter: No* in Tables 7.1, 7.2, and 7.3), we discard texts for which the output of the model was ill-formatted (i.e., no sequence between inverted commas was in the model output) or sequences were too similar to the original text. The number of training texts further decreases in the ‘*classifier filtering*’ setups (*Filter: Yes* in Tables 7.1, 7.2, and 7.3), in which we also discard the sequences that did not pass *classifier filtering* (Sec-

		Test data				
		MHS		MDA	HateCheck	
		$n(\text{train})$	M-F <sub>1</sub>	Hate F <sub>1</sub>	M-F <sub>1</sub>	M-F <sub>1</sub>
Original gold data (MHS)		30,132	<b>.805</b> $\pm$ .003	<b>.708</b> $\pm$ .006	.546 $\pm$ .022	.314 $\pm$ .012
EDA		30,132	<b>.807</b> $\pm$ .003	<b>.714</b> $\pm$ .005	.566 $\pm$ .012	.315 $\pm$ .012
Gen. Model	Filter					
Llama-2 Chat 7B	No	28,289	.742 $\pm$ .004	.643 $\pm$ .004	.661 $\pm$ .007	.490 $\pm$ .016
	Yes	21,132	.786 $\pm$ .004	.686 $\pm$ .005	.595 $\pm$ .012	.326 $\pm$ .007
Mistral 7B Instruct	No	29,344	.743 $\pm$ .007	.654 $\pm$ .005	.686 $\pm$ .003	<b>.551</b> $\pm$ .009
	Yes	22,453	.784 $\pm$ .005	.684 $\pm$ .007	.595 $\pm$ .013	.337 $\pm$ .009
Mixtral 8x7B Instruct	No	29,351	.718 $\pm$ .007	.632 $\pm$ .006	<b>.696</b> $\pm$ .005	.541 $\pm$ .008
	Yes	22,325	.783 $\pm$ .003	.687 $\pm$ .004	.619 $\pm$ .007	.328 $\pm$ .004

Table 7.2: Results of RobertaBase models trained on synthetic data only (average of 5 runs  $\pm$  stdev). Grey cells indicate out-of-distribution performance. *Filter:No* means that only paraphrased sequences too similar to the original ones and ill-formatted texts were discarded. *Filter:Yes* means that *classifier filtering* was applied.

		Test data				
		MHS		MDA	HateCheck	
		$n(\text{train})$	M-F <sub>1</sub>	Hate F <sub>1</sub>	M-F <sub>1</sub>	M-F <sub>1</sub>
Original gold data (MHS)		30,132	<b>.809</b> $\pm$ .002	<b>.717</b> $\pm$ .005	.522 $\pm$ .018	.347 $\pm$ .008
EDA		30,132	<b>.809</b> $\pm$ .004	<b>.718</b> $\pm$ .005	.537 $\pm$ .005	.354 $\pm$ .012
Gen. Model	Filter					
Llama-2 Chat 7B	No	28,289	.736 $\pm$ .004	.642 $\pm$ .005	.670 $\pm$ .014	.597 $\pm$ .019
	Yes	21,116	.785 $\pm$ .0066	.684 $\pm$ .012	.569 $\pm$ .019	.332 $\pm$ .016
Mistral 7B Instruct	No	29,344	.732 $\pm$ .010	.643 $\pm$ .007	.672 $\pm$ .006	.636 $\pm$ .017
	Yes	22,445	.782 $\pm$ .005	.678 $\pm$ .006	.564 $\pm$ .020	.387 $\pm$ .008
Mixtral 8x7B Instruct	No	29,351	.710 $\pm$ .007	.626 $\pm$ .004	<b>.697</b> $\pm$ .007	<b>.638</b> $\pm$ .014
	Yes	22,292	.781 $\pm$ .007	.679 $\pm$ .013	.579 $\pm$ .028	.390 $\pm$ .021

Table 7.3: Results of DeBERTa Base models trained on synthetic data only (average of 5 runs  $\pm$  stdev). Grey cells indicate out-of-distribution performance. *Filter:No* means that only paraphrased sequences too similar to the original ones and ill-formatted texts were discarded. *Filter:Yes* means that *classifier filtering* was applied.



tion 7.1.2). For these setups, models are on average trained on around two thirds of the amount of data available to the other models, with a different class balance: a large majority of examples that are discarded during this phase are *hateful*, so in the *classifier filtering* setups the synthetic data is composed of very few *hateful* examples. Surprisingly, however, these setups achieve comparable performance with models trained on the original gold data.

Our experimental results show that synthetic data can get close to the performance of classifiers trained on gold data, indicating the potential utility of the approach and confirming the preliminary findings of Chapter 6. However, there is a clear difference between the setups in which *classifier filtering* is employed and those in which it is not. In particular, filtering leads to better performance on the same data distribution (i.e., when testing on the MHS dataset), which could be attributed to the classifier overfitting the original data and misclassifying texts that drift too far from it. Conversely, not filtering typically leads to losses of around .04  $F_1$  over using actual gold data in in-distribution scenarios, but it can heavily boost performance in out-of-distribution scenarios, with improvements of up to .18  $F_1$  for the MDA dataset and up to .30  $F_1$  on HateCheck. This might be due to potential injection of more lexical variety by the LLMs during the paraphrasing process, positively affecting models trained on synthetic data to generalize out-of-distribution.

While EDA can lead to some mild improvements in performance, especially in out-of-distribution scenarios, this impact appears to be reduced compared with that of LLM-paraphrased examples in the *no classifier filtering* setups, which clearly outperform EDA when tested cross-dataset. While in Chapter 5 we found EDA to be stronger than generative LLM-based DA (*generative DA*) in a scenario in which we aimed at augmenting existing gold data to increase the amount and variety of

existing training data, EDA might not be an equally competitive choice when *i*) it is necessary to use synthetic data only and *ii*) cross-distribution generalization is deemed important.

### 7.3 Intrinsic Evaluation

Our experiments suggest that synthetic data can be useful in making models more robust to out-of-distribution scenarios (cf. Tables 7.1, 7.2, and 7.3). This would ideally make them advisable for use cases in which hate speech detection has to be performed on target data from a different domain (e.g., genre, topic), as they appear to be more robust from a performance standpoint. However, no in-depth investigation has been carried out so far to highlight what would be the *qualitative* differences between synthetic and gold data for this task. We therefore conduct a qualitative analysis in order to understand what aspects actually play a role in this shift in model performance, to discover what this data contains, if it is realistic enough to mimic real training data and, ultimately, if it is truly advisable to use it in real application scenarios.

The qualitative analysis was carried out by two annotators, one male and one female, both with expertise in online language use, hate speech, and LLM-generated text.

The human evaluation focuses on three aspects:

- The *realism* of the synthetic data, i.e., whether a specific message could realistically be found as a social media post;
- To what extent synthetic data creation ensures *hateful content preservation*, i.e., if after paraphrasing the *hateful* messages remain *hateful* (and vice versa for *non hateful* ones);

- Whether the *representation of target identities* is different in the synthetic data compared with the gold data (e.g., if, after paraphrasing, a text that was originally about black women is still about black women, or whether the identity representation was erased).

These aspects can, in fact, have a number of implications on real-world usage of synthetic data for hate speech detection. For instance, if synthetic data is not realistic, it may introduce spurious correlations between certain tokens and labels, making models overfit to lexical items that rarely occur in real-world data (Ramponi and Tonelli, 2022). On the other hand, label preservation is important because the data augmentation process assumes that the label of the original text will be preserved. Indeed, data augmentation gives the opportunity to modify existing data in order to obtain more training samples *without further manual annotation*. However, if a large fraction of the labels changes after augmentation, it might not always be worth it, as classifiers trained on wrongly-labeled synthetic data could have unpredictable performance. Finally, in the frequent cases in which the targets of hate represented in a dataset have been carefully balanced to ensure a fair representation of different groups, changing this distribution through the augmentation process may not be desirable. Moreover, training a classifier on synthetic data in which specific targets of hate have been neglected would potentially affect classifier fairness, hurting already marginalized communities (Xu et al., 2021).

We conduct the human annotation in two steps:

- Annotators are provided with a sample of 500 texts (both gold and synthetic) and asked whether each example appears to be written by a human or an LLM, to estimate how easy it is to spot LLM-written text;

- Annotators are provided with an additional sample of 3,000 synthetic-only examples, i.e., 1,000 texts created by each of the three generative models we employ in our experiments, equally split between the labels. These examples are annotated along a variety of axes, including grammaticality, presence of hate speech, and presence of identity mentions.

Annotation details are reported in the sections below, and the full annotation guidelines are reported in Appendix B.

### 7.3.1 Realism of Synthetic Texts

The first aspect we investigate is how easy it is to spot synthetic data for a human annotator, as a proxy for how *realistic* the synthetic texts are (RQ2.2). While realism is not fundamental for models to recognize hate speech, the ability (or lack thereof) of a human to recognize a text as produced by an LLM might indicate that synthetic texts do exhibit characteristics that cannot fully mimic those of human-written texts. This might, in turn, result in models learning spurious correlations from LLM-written texts, i.e., relying on some expressions or unusual words as shortcuts for classifying posts as hateful.

**Human or LLM?** In order to assess how real-passing the synthetic texts are, we provide annotators with 500 examples that are a mix of gold texts and texts generated using the three different LLMs that we use in our experiments. To avoid biasing the responses, annotators were not aware of the ratio of real and synthetic examples during the annotation, which is 25% gold and 75% synthetic (i.e., 125 gold examples and 125 synthetic examples for each of the 3 models, for a total of 375 synthetic examples).

The annotators had an accuracy of 88% in correctly identifying LLM-authored texts, with a precision of 0.83 and a recall of 0.90. The differences across models were small: they achieved 87%, 90%, and 92% accuracy in correctly identifying synthetic texts generated with Llama-2 Chat, Mistral, and Mixtral, respectively.

Inter-annotator agreement was calculated on 20% of the annotated examples, selected randomly. The annotators agreed 89% of the time, with a Krippendorff's alpha coefficient of 0.73. We believe that the high accuracy might be due to the annotators' expertise and familiarity with LLM-generated text. However, this shows that, to an expert eye, synthetic texts might not be quite as realistic as expected. For instance, texts with convoluted constructions and unusual (but polite) lexical choices were often easily recognized as synthetic, such as *'kindly halt this conduct characterized by the blending of unconventional gender identities and feminist ideologies'* (paraphrase of: *'please stop this queer feminist bullshit'*).

**Prompt Failures and Grammar** Annotators were asked to label 3,000 synthetic examples (1,000 per model) and report whether *a)* the output did not correctly fulfill the prompt (e.g., the model refused to answer or it answered with a description of the gold text), which we deem a *prompt failure*, *b)* the grammar was deemed correct / realistic, *c)* the 'world knowledge' exhibited by the model was considered acceptable.

Table 7.4 reports the percentage of synthetic texts created with each model and annotated according to these three aspects. Overall, there are no large differences across models: all the models produce sequences that are acceptable with regards to grammar and world knowledge in most cases. Prompt failures are more common with Llama-2 Chat, while they are much less common with Mixtral 8x7B. For prompt failures, the IAA among our annotators was fairly high, with a Krippendorff's alpha

	Llama	Mistral	Mixtral
Prompt failure	14%	11%	5%
Grammar incorrect	1%	2%	1%
World knowledge incorrect	4%	5%	4%

Table 7.4: Synthetic text realism annotations.

of 0.76. While Llama is more prone to prompt failures (14% of produced texts), it actually produces texts that appear slightly more realistic to human eyes when they are not prompt failures. This hypothesis is supported by the lower accuracy of humans in identifying Llama authored texts compared with the other models, as we have observed.

Overall, while this is often taken for granted, we find that synthetic texts are not necessarily human-like, even when they appear grammatically correct and plausible, as expert eyes can still tell them apart from human-written texts.

### 7.3.2 Redistribution of Hateful Texts

The second aspect we investigate in our intrinsic analysis is whether models maintain hatefulness during the synthetic data creation process. Ideally, paraphrasing a text classified as hateful should output another text of the same class. We therefore ask annotators to label the same 3,000 synthetic examples following the guidelines for hate speech annotation that were adopted for building the MHS corpus, and then compare the labels with those originally assigned to the gold texts. The difficulty of preserving labels in LLM-based data augmentation has already been attested in the past (e.g., Kumar et al. (2020)), as we have seen in the previous chapters, but to our knowledge it has never been qualitatively assessed for subjective tasks such as hate speech detection.

While our aggregation process for the *hate speech* label in the MHS

corpus (Sec. 3.4) removed the *unclear* label, our annotators could label texts as *hateful*, *non hateful*, and *unclear*, since they were asked to follow the original guidelines used for creating the corpus, which we report in detail in Appendix B. For the *hate speech* label, the inter-annotator agreement between our annotators was moderately high, with a Krippendorff’s alpha of 0.70.

Figures 7.1, 7.2, and 7.3 show the redistribution of hateful content for the Llama-2 Chat model, the Mistral 7B Instruct model, and the Mixtral 8x7B Instruct model, respectively.

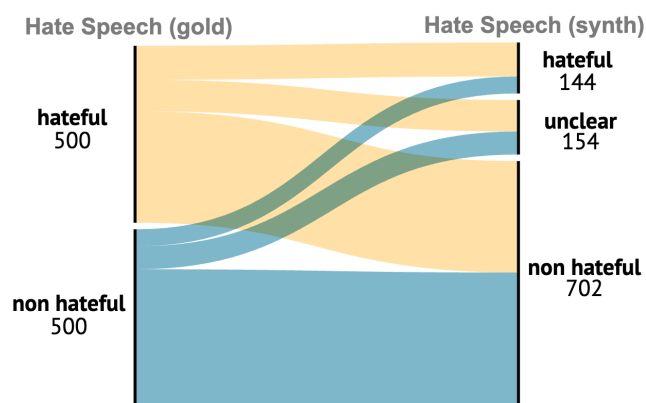


Figure 7.1: Distribution of hateful and non hateful texts in the subset of gold and synthetic data created using the Llama 2 Chat 7B model.

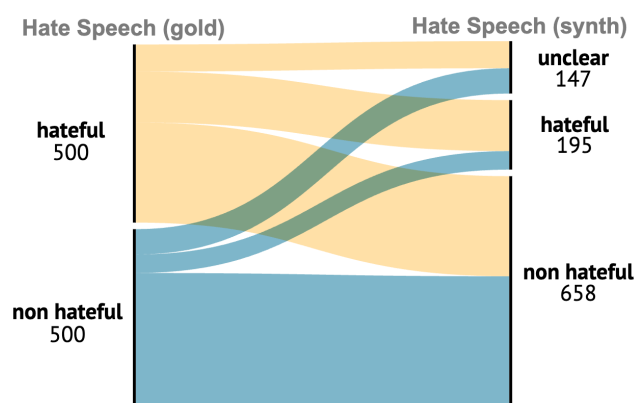


Figure 7.2: Distribution of hateful and non hateful texts in the subset of gold and synthetic data created using the Mistral 7B Instruct model.

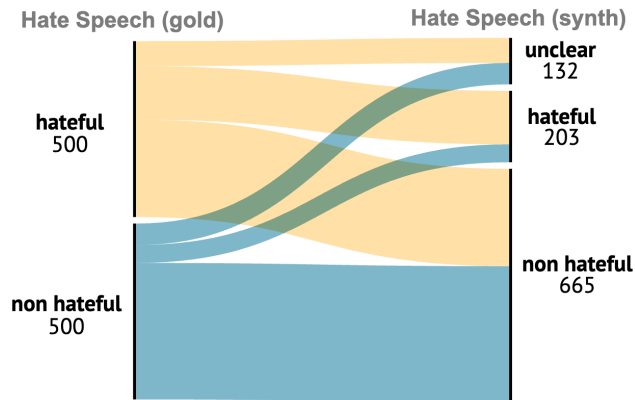


Figure 7.3: Distribution of hateful and non hateful texts in the manually labeled subset of gold and synthetic data created using the Mixtral 8x7B Instruct model.

Overall, tendencies to produce synthetic examples with a different *hate speech* label than their original version are similar across models. For all models, almost half of the examples go through a change of label, with most of these changes regarding texts that are originally *hateful*, which are rendered *non hateful* through the LLM paraphrasing process. We hypothesize this change in label distribution is in part due to the alignment of models, which tends to avoid generating toxic language as they are trained to minimize inappropriate, offensive or unethical uses (Rao et al., 2023). Another small portion of these includes *prompt failures*. Interestingly, there also are a number of examples that transition from being *non hateful* to being *hateful*. In particular, through manually looking at these examples, we note that there are several potential reasons for these changes. Many are cases of clearly sarcastic texts that, through the paraphrasing process, are turned into texts that might sound serious (e.g. *I like that brown people defending their home is 'barbaric'* being turned into *It's savage, in my view, when brown people resist invaders and protect their homes*). Others can be attributed to genuine disagreements between annotators or annotation errors.

Globally, we find that changes in hateful content are quite common,



showing that synthetic data should not always be trusted to maintain the same class distribution as the original gold data when used for hate speech or abusive language detection.

### 7.3.3 Redistribution of Target Identities

Given that the representation of different target identities can lead to discrepancies in classification performance across identity groups, as we have already discussed in Chapter 5, we also analyze the redistribution of identity categories in the synthetic data. Since the findings of our analysis generalize across models, in this section we mostly discuss the statistics for Mixtral 8x7B Instruct, while we do report results across all three generative models.

Annotators are provided the same guidelines as the annotators of the MHS corpus, with 7 categories of identity groups to annotate for both *hateful* and *non hateful* examples: *age*, *disability*, *gender*, *origin*, *race*, *religion*, and *sexuality*. The redistribution of identity group mentions is shown in Figure 7.6. For this, our annotators are again provided the same guidelines as the annotators of the MHS corpus, with 7 different identity groups to annotate for both *hateful* and *non hateful* examples: *age*, *disability*, *gender*, *origin*, *race*, *religion*, and *sexuality*. The redistribution of the presence of identity group mentions is shown in Figure 7.4 for Llama Chat 7b, Figure 7.5 for Mistral 7b Instruct, and Figure 7.6 for Mixtral 8x7B Instruct.

The analysis shows that over one third of the examples lose the reference to the original identity group(s) when paraphrased (cf. Figure 7.6; from any category on the left to *no target* on the right). In particular, the representation of the *gender*, *race*, and *sexuality* categories is heavily reduced, while this reduction is less noticeable for other categories such as *religion* or *disability*. We hypothesize this may also be due to the

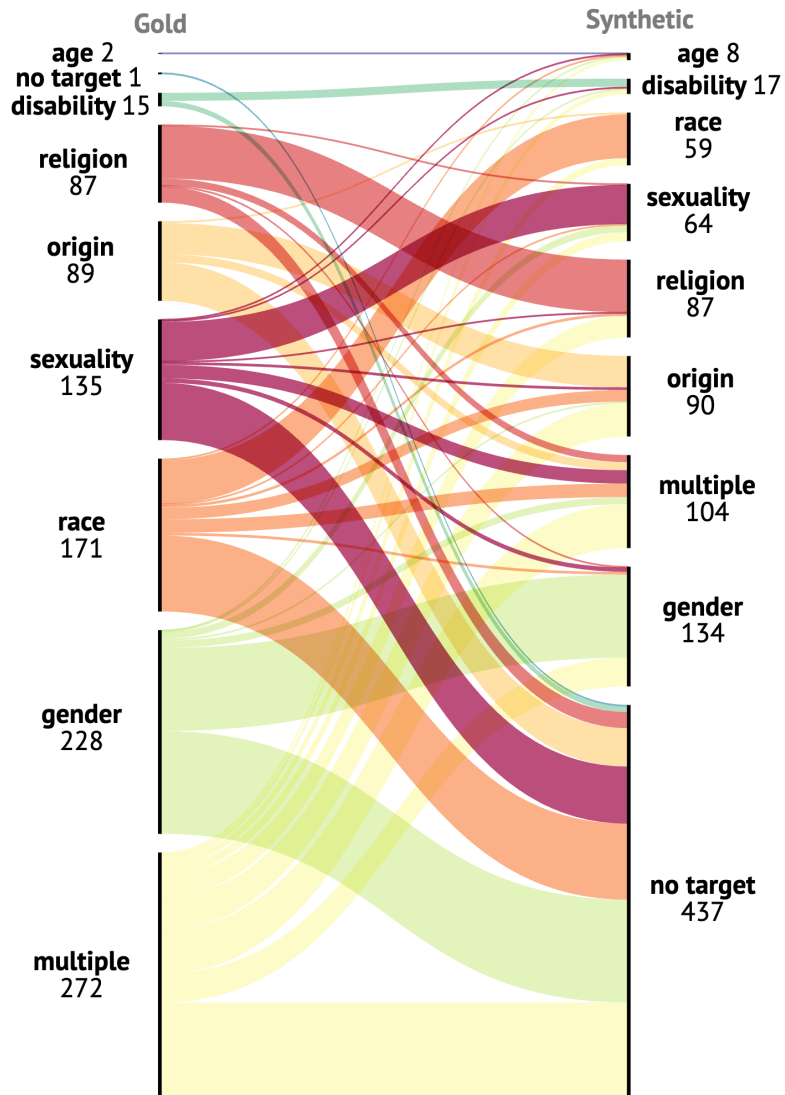


Figure 7.4: Target identity redistribution in synthetic texts created with Llama 2 Chat 7B.

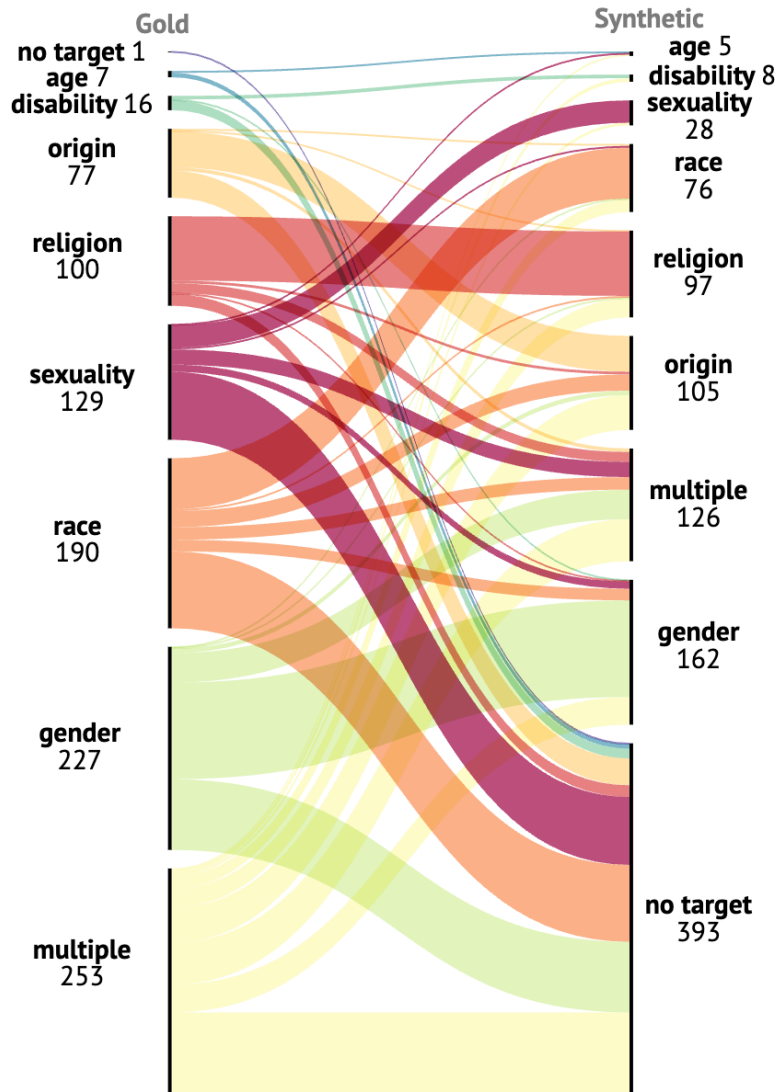


Figure 7.5: Target identity redistribution in synthetic texts created with Mistral 7B Instruct.

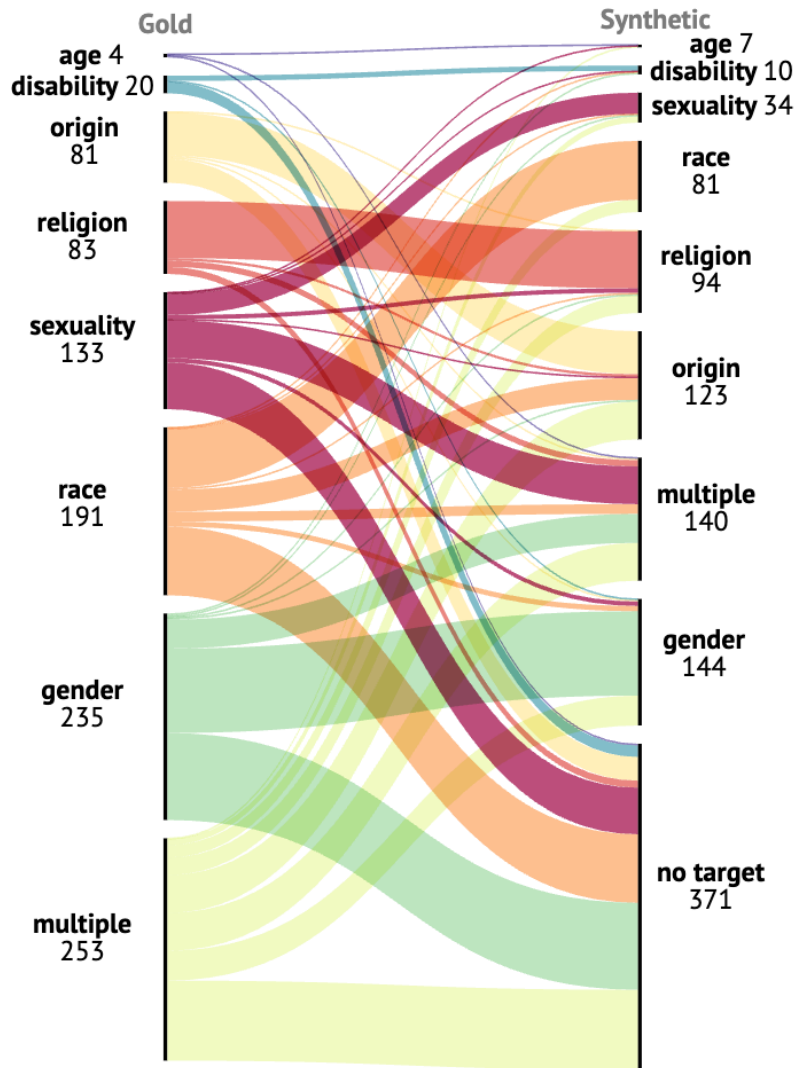


Figure 7.6: Target identity redistribution with the Mixtral 8x7B Instruct model.

## CHAPTER 7. REALISM AND QUALITY

Target	Subset	Top- $k$ tokens
AGE	ORIGINAL	fuck, ass, b*tch, fucking, 🍑, shit, pussy, racist, cunt, kids
	LLAMA-2	person, language, individuals, people, offensive, individual, sexual, children, mother, life
	MISTRAL	female, woman, children, individuals, anus, person, mother, tiny, outdated, life
	MIXTRAL	individuals, individual, woman, children, mother, person, people, sexual, child, women
DISABILITY	GOLD	r*tarded, r*tard, fucking, fuck, shit, ass, b*tch, r*tards, people, kill
	LLAMA-2	language, person, offensive, individuals, people, derogatory, respectful, disabilities, respect, intellectual
	MISTRAL	person, individuals, individual, woman, foolish, intellectual, girl, intellectually, anonymous, intelligence
	MIXTRAL	individuals, person, foolish, individual, intellectually, impaired, intelligence, mentally, lack, ignorant
GENDER	GOLD	b*tch, fuck, ass, fucking, cunt, b*tches, shit, pussy, wh*re, sl*t
	LLAMA-2	person, language, offensive, sexual, individuals, people, derogatory, respectful, respect, women
	MISTRAL	woman, women, female, person, females, individual, individuals, penis, behavior, foolish
	MIXTRAL	woman, women, person, individuals, individual, promiscuous, ignorant, sex, foolish, sexual
ORIGIN	GOLD	fuck, fucking, country, shit, people, america, ass, white, american, b*tch
	LLAMA-2	individuals, people, country, language, person, derogatory, offensive, america, immigrants, beliefs
	MISTRAL	individuals, america, country, people, return, americans, iran, person, white, american
	MIXTRAL	individuals, country, people, america, person, individual, return, american, nation, immigrants
RACE	GOLD	n*ggas, n*ggas, fuck, ass, fucking, white, shit, b*tch, n*gger, 🤡
	LLAMA-2	people, individuals, language, person, offensive, derogatory, respectful, respect, race, white
	MISTRAL	individuals, person, people, white, individual, woman, black, racist, behavior, despicable
	MIXTRAL	individuals, people, person, white, individual, racist, black, african, woman, women
RELIGION	GOLD	fuck, jews, fucking, shit, people, muslim, jew, muslims, white, god
	LLAMA-2	people, individuals, beliefs, language, offensive, person, respect, including, religion, action
	MISTRAL	individuals, jews, jewish, muslim, person, individual, despicable, muslims, white, islam
	MIXTRAL	individuals, people, jewish, individual, jews, muslim, muslims, person, islam, white
SEXUALITY	GOLD	f*ggot, fuck, fucking, ass, f*g, shit, f*ggots, gay, b*tch, dick
	LLAMA-2	language, offensive, sexual, derogatory, person, individuals, people, respect, respectful, lgbtq
	MISTRAL	person, effeminate, homosexual, gay, individual, woman, individuals, penis, derogatory, term
	MIXTRAL	homosexual, person, individuals, gay, individual, term, behavior, derogatory, effeminate, people

Table 7.5: Top- $k$  = 10 most informative tokens for the *hateful* class, according to the PMI metric across targets of hate in GOLD and SYNTHETIC posts paraphrased using Llama-2 Chat 7B, Mistral 7B Instruct, and Mixtral 8x7B Instruct.

*alignment* process for these models, which is likely to prevent models from generating hateful messages against the most common targets of hate. Instead, for other categories such as *religion*, *origin* or *disability*, the model may not have been exposed to them during training, as they are more scarcely represented in widely-used hate speech datasets. Furthermore, creating synthetic paraphrases of texts also appears to reduce the representation of intersectionality, with over half of the gold texts that represent multiple identity categories being either turned into synthetic texts that mention one single identity category or none at all.

To investigate this further, we extract the most informative tokens for the *hateful* class from both the original gold data and the synthetic data

by computing `npw_relevance`, a normalized class relevance metric based on PMI, using the Variationist tool (Appendix A). We report the statistics regarding the most informative tokens in Table 7.5.<sup>9</sup>

From this analysis, it is clear that LLMs tend to turn any potentially harmful input into its ‘safer’ counterpart, with all slurs completely disappearing from the list of the most informative tokens for the *hateful* class for each target category. While the synthetic data we analyze actually *is* still useful as training data for classifiers, as we saw in Section 7.2, it is clear from this analysis that the *content* of this data is largely different from that of the original gold dataset. This might lead to models learning ‘shortcuts’ for classification, and wrongly assuming that certain commonly used words, such as *woman* or *homosexual*, are to be associated with hateful texts. This could have unpredictable consequences if models trained on synthetic data are actually deployed for the identification of hate speech.

## 7.4 Conclusions on Realism and Quality

In this chapter, we have carried out an assessment of synthetic data beyond the mainstream classifier performance evaluation, with the goal of linking classifier performance with an intrinsic qualitative analysis focused on *realism* (RQ2.2). In addition, we aimed to understand the potential risks and drawbacks of using synthetic data for a delicate task such as hate speech detection (RQ2.3). While from a classifier performance and robustness standpoint (Q1.1 and Q1.2), synthetic data shows to be helpful in out-of-distribution scenarios, our qualitative analysis proves that we should not take for granted the preservation of key features of gold data in synthetic data. First, synthetic data might introduce

---

<sup>9</sup>Given the large number of slurs in these lists, we obfuscate profanities as discussed in Section 2.1.1.

spurious correlations due to the language used by models, as it is easily spotted by expert humans. In addition, we show that the preservation of *hate speech* labels during the augmentation process should not be automatically assumed, even when the data still appears to be useful for training a classifier. Finally, LLM-generated paraphrases of gold data show a drastically different identity group distribution compared with the original data, making synthetic data unreliable with regards to identity representation.

Overall, our analysis shows that while classifier performance might show synthetic data to be potentially useful, it can actually hide potential risks we may often be unaware of, hidden behind our assumptions regarding realism, label, and identity preservation.





## Chapter 8

### Conclusions

Synthetic data has been proposed in past research work as a way to mitigate many of the issues with existing abusive language detection datasets and models, especially thanks to the supposedly human-like capabilities of LLMs (Brown et al., 2020). In theory, the use of synthetic data – both as a way to artificially increase the amount of available training data for abusive language classifiers and as a way to substitute existing datasets containing user-written posts – could reduce the impact of problems such as negative impact on annotators, privacy issues, dataset decay, and representation biases (Chapter 2). However, the use of synthetic data for abusive language detection has not been explored much beyond sheer model performance in previous work.

In this dissertation, we have investigated the role of synthetic data for training abusive language detection models, paying attention to both its potential and its risks. More specifically, in contrast with a large body of previous research on this topic, we focused on aspects that are often neglected in machine learning research (Birhane et al., 2022), in addition to *performance* (**RQ1.1**). In particular, we started with an analysis focused on *robustness*, with the goal of understanding whether synthetic data can help models generalize to out-of-distribution scenarios (**RQ1.2**), finding

that this is actually possible, although not always guaranteed to work well (Chapter 4). Subsequently, we carried out an evaluation of the influence of synthetic data on *fairness*, with the aim of exploring the different impact of data augmentation on specific target identities (**RQ1.3**), finding that synthetic data created using LLMs can improve the performance of models on scarcely represented targets of abuse, although this improvement can be inconsistent. In fact, we found that sometimes synthetic data created using generative LLMs works best when paired with less computationally expensive and more traditional DA methods, such as word substitution or deletion. We also observed that including identity group information in the prompts to the generative models led to more realistic examples (Chapter 5). We then moved on to a more qualitative look at synthetic data, first investigating setups in which it can be used to *substitute* real-world training data for *privacy* reasons (**RQ2.1**), finding that models trained on fully synthetic data can achieve classification performance on par with models trained on gold data, in some cases showing better cross-dataset robustness as well. In addition, the synthetic data could not be traced back to the original social media posts it derived from, showing the potential of this kind of application (Chapter 6). Finally, we carried out a manual validation on synthetic data for abusive language detection, with the aim of assessing its *realism* and investigating potential *risks* related to it (**RQ2.2** and **RQ2.3**). We found that, while synthetic data is very promising from a quantitative standpoint, a more qualitative look shows that synthetic data can hide some pitfalls and risks: above all, synthetic data created through rewriting is not guaranteed to mirror the characteristics of the data it is supposed to emulate (Chapter 7).

Aside from the work we carried out in this thesis, there still are a multitude of aspects of synthetic data usage in NLP that would be worth

exploring, as our findings can only be generalizable to a certain extent. First of all, while we have focused only on abusive language detection, it would be interesting to explore if our qualitative insights could generalize to other NLP tasks, especially in light of how widespread the use of synthetic data has become recently in the field. Furthermore, we dealt only with English data in this work, mostly because of how accessible data and models are for this language, and comparisons across models, as well as cross-dataset testing, were important for our experiments. Arguably, however, the real potential benefits of synthetic data would emerge when used in scenarios in which there is actual data scarcity, as is the case for under-resourced languages. An initial exploration of similar approaches for Italian was carried out in Leonardelli and Casula (2023), which we plan to expand on in the future.

Future work on this topic could also include more detailed studies of some of the dimensions we focused on: for instance, *privacy* implications of synthetic data could be explored further by combining our proposed approach with differential privacy methods, typically used to prevent models from leaking private information into their outputs (Yu et al., 2022). Furthermore, *fairness* aspects could be investigated through the point of view of *real* targets of hate through participatory methods rather than through target identity categories defined *a-priori*, which are limited in their representativeness (Caselli et al., 2021b). Another potential research direction that could follow-up on the present work is a more in-depth linguistic exploration of synthetic data beyond lexicon, for instance by focusing on *syntax*, or more thorough explorations of the impact of *model alignment* on the synthetic data. Finally, exploring the creation of synthetic data in a ‘zero-shot’ setup, not starting from any task-specific data in an augmentation setup, could complement our findings, and offer a more comprehensive view of the benefits and pit-

falls of the usage of synthetic data.

While practically useful for getting a small glimpse at how Internet mega-corporations ‘automatically’ decide what constitutes abuse and what does not, a large portion of research work on the topic of abusive language detection - including this thesis - is somewhat sheltered from the complexities of real-life content moderation online, up in the ivory towers of readily-available benchmark datasets and high  $F_1$  scores. However, we believe in the importance of research work that is aware of its limits and that is deliberate in shedding light on even tiny corners of knowledge, regardless.

Although the present work is far from exhaustive with respect to the potential and risks of using synthetic data for abusive language detection, we have offered an exploration of its advantages and drawbacks that we hope will be useful as a stepping stone for future work on the topic, as well as to serve as a reminder that quantitative metrics alone might not always allow us to see the full picture with regards to what *works* and what does not.

# Bibliography

- Anaby-Tavor, A., Carmeli, B., Goldbraich, E., Kantor, A., Kour, G., Shlomov, S., Tepper, N., and Zwerdling, N. (2020). Do Not Have Enough Data? Deep Learning to the Rescue! In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7383–7390.
- Anderson, L. and Barnes, M. (2023). Hate Speech. In Zalta, E. N. and Nodelman, U., editors, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Fall 2023 edition.
- Anthis, J., Lum, K., Ekstrand, M., Feller, A., D’Amour, A., and Tan, C. (2024). The Impossibility of Fair LLMs. arXiv:2406.03198 [cs, stat].
- Anthony, L. (2013). A critical look at software tools in corpus linguistics. *Linguistic Research*, 30(2):141–161.
- Ashida, M. and Komachi, M. (2022). Towards automatic generation of messages countering online hate speech and microaggressions. In Narang, K., Mostafazadeh Davani, A., Mathias, L., Vidgen, B., and Tatal, Z., editors, *Proceedings of the Sixth Workshop on Online Abuse and Harms (WOAH)*, pages 11–23, Seattle, Washington (Hybrid). Association for Computational Linguistics.
- Ashraf, N., Zubiaga, A., and Gelbukh, A. (2021). Abusive language detection in youtube comments leveraging replies as conversational context. *PeerJ Computer Science*, 7:e742.

- Attanasio, G., Pastor, E., Di Bonaventura, C., and Nozza, D. (2023). ferret: a framework for benchmarking explainers on transformers. In Croce, D. and Soldaini, L., editors, *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 256–266, Dubrovnik, Croatia. Association for Computational Linguistics.
- Azam, U., Rizwan, H., and Karim, A. (2022). Exploring data augmentation strategies for hate speech detection in Roman Urdu. In Calzolari, N., Béchet, F., Blache, P., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Odijk, J., and Piperidis, S., editors, *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4523–4531, Marseille, France. European Language Resources Association.
- Basile, V. (2020). It’s the end of the gold standard as we know it. On the impact of pre-aggregation on the evaluation of highly subjective tasks. *CEUR Workshop Proceedings*, 2776:31–40.
- Basile, V., Bosco, C., Fersini, E., Nozza, D., Patti, V., Rangel Pardo, F. M., Rosso, P., and Sanguinetti, M. (2019). SemEval-2019 Task 5: Multilingual Detection of Hate Speech Against Immigrants and Women in Twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 54–63, Minneapolis, Minnesota, USA. Association for Computational Linguistics.
- Bayer, M., Kaufhold, M.-A., and Reuter, C. (2022). A Survey on Data Augmentation for Text Classification. *ACM Computing Surveys*, 55(7):146:1–146:39.
- Bender, E. M., Gebru, T., McMillan-Major, A., and Shmitchell, S. (2021). On the Dangers of Stochastic Parrots: Can Language Models Be Too

- Big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, pages 610–623, New York, NY, USA. Association for Computing Machinery.
- Bhattacharya, S., Singh, S., Kumar, R., Bansal, A., Bhagat, A., Dawer, Y., Lahiri, B., and Ojha, A. K. (2020). Developing a multilingual annotated corpus of misogyny and aggression. In Kumar, R., Ojha, A. K., Lahiri, B., Zampieri, M., Malmasi, S., Murdock, V., and Kadar, D., editors, *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pages 158–168, Marseille, France. European Language Resources Association (ELRA).
- Birhane, A., Kalluri, P., Card, D., Agnew, W., Dotan, R., and Bao, M. (2022). The Values Encoded in Machine Learning Research. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 173–184, Seoul Republic of Korea. ACM.
- Bosco, C., Patti, V., Frenda, S., Cignarella, A. T., Paciello, M., and D’Errico, F. (2023). Detecting racial stereotypes: An Italian social media corpus where psychology meets NLP. *Information Processing and Management*, 60(1):103118.
- Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language Models are Few-Shot Learners. *arXiv:2005.14165 [cs]*.
- Cabitzza, F., Campagner, A., and Basile, V. (2023). Toward a perspectivist

turn in ground truthing for predictive computing. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(6):6860–6868.

Caselli, T., Basile, V., Mitrović, J., and Granitzer, M. (2021a). HateBERT: Retraining BERT for abusive language detection in English. In Mostafazadeh Davani, A., Kiela, D., Lambert, M., Vidgen, B., Prabhakaran, V., and Waseem, Z., editors, *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, pages 17–25, Online. Association for Computational Linguistics.

Caselli, T., Basile, V., Mitrović, J., Kartoziya, I., and Granitzer, M. (2020). I feel offended, don't be abusive! implicit/explicit messages in offensive and abusive language. In Calzolari, N., Béchet, F., Blache, P., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6193–6202, Marseille, France. European Language Resources Association.

Caselli, T., Cibin, R., Conforti, C., Encinas, E., and Teli, M. (2021b). Guiding principles for participatory design-inspired natural language processing. In Field, A., Prabhumoye, S., Sap, M., Jin, Z., Zhao, J., and Brockett, C., editors, *Proceedings of the 1st Workshop on NLP for Positive Impact*, pages 27–35, Online. Association for Computational Linguistics.

Chakravarthi, B. R., Priyadharshini, R., Ponnusamy, R., Kumaresan, P. K., Sampath, K., Thenmozhi, D., Thangasamy, S., Nallathambi, R., and McCrae, J. P. (2021). Dataset for identification of homophobia and transophobia in multilingual youtube comments.

Chapelle, O., Weston, J., Bottou, L., and Vapnik, V. (2000). Vicinal risk



- minimization. In Leen, T., Dietterich, T., and Tresp, V., editors, *Advances in Neural Information Processing Systems*, volume 13. MIT Press.
- Chen, J., Tam, D., Raffel, C., Bansal, M., and Yang, D. (2023). An Empirical Survey of Data Augmentation for Limited Data Learning in NLP. *Transactions of the Association for Computational Linguistics*, 11:191–211.
- Chung, H., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S., Webson, A., Gu, S. S., Dai, Z., Suzgun, M., Chen, X., Chowdhery, A., Castro-Ros, A., Pellat, M., Robinson, K., Valter, D., Narang, S., Mishra, G., Yu, A., Zhao, V., Huang, Y., Dai, A., Yu, H., Petrov, S., Chi, E. H., Dean, J., Devlin, J., Roberts, A., Zhou, D., Le, Q. V., and Wei, J. (2024). Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.
- Chung, J., Kamar, E., and Amershi, S. (2023). Increasing diversity while maintaining accuracy: Text data generation with large language models and human interventions. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 575–593, Toronto, Canada. Association for Computational Linguistics.
- Cortes, C. and Vapnik, V. N. (2004). Support-Vector Networks. *Machine Learning*, 20:273–297.
- 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. In Bouamor, H., Pino, J., and Bali, K., editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 4069–4085. Association for Computational Linguistics.

- Del Barrio, E., Cuesta-Albertos, J. A., and Matrán, C. (2018). An optimal transportation approach for assessing almost stochastic order. In *The Mathematics of the Uncertain*, pages 33–44. Springer.
- Del Vigna, F., Cimino, A., Dell’Orletta, F., Petrocchi, M., and Tesconi, M. (2017). Hate me, hate me not: Hate speech detection on Facebook.
- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T., editors, *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.
- Dror, R., Shlomov, S., and Reichart, R. (2019). Deep dominance - how to properly compare deep neural models. In Korhonen, A., Traum, D., and Màrquez, L., editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2773–2785, Florence, Italy. Association for Computational Linguistics.
- D’Sa, A. G., Illina, I., Fohr, D., Klakow, D., and Ruiter, D. (2021). Exploring Conditional Language Model Based Data Augmentation Approaches for Hate Speech Classification. In *Text, Speech, and Dialogue: 24th International Conference, TSD 2021, Olomouc, Czech Republic, September 6–9, 2021, Proceedings*, pages 135–146, Berlin, Heidelberg. Springer-Verlag.
- Edwards, A. and Camacho-Collados, J. (2024). Language models for text classification: Is in-context learning enough? In Calzolari, N., Kan, M.-Y., Hoste, V., Lenci, A., Sakti, S., and Xue, N., editors, *Proceedings of the 2024 Joint International Conference on Computational Lin-*

*guistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 10058–10072, Torino, Italia. ELRA and ICCL.

European Parliament and Council of the European Union (2022). Regulation (EU) 2022/2065 of the European Parliament and of the Council of 19 October 2022 on a Single Market for Digital Services and amending Directive 2000/31/EC (Digital Services Act). *Official Journal L277*, pages 1–102.

Fan, A., Lewis, M., and Dauphin, Y. (2018). Hierarchical neural story generation. In Gurevych, I. and Miyao, Y., editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Fano, R. M. (1961). *Transmission of information: A statistical theory of communications*. MIT Press, New York, USA.

Fanton, M., Bonaldi, H., Tekiroğlu, S. S., and Guerini, M. (2021). Human-in-the-loop for data collection: a multi-target counter narrative dataset to fight online hate speech. In Zong, C., Xia, F., Li, W., and Navigli, R., editors, *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 3226–3240, Online. Association for Computational Linguistics.

Feng, S. Y., Gangal, V., Wei, J., Chandar, S., Vosoughi, S., Mitamura, T., and Hovy, E. (2021). A survey of data augmentation approaches for NLP. In Zong, C., Xia, F., Li, W., and Navigli, R., editors, *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988, Online. Association for Computational Linguistics.

- Fergadiotis, G., Wright, H. H., and Green, S. B. (2015). Psychometric evaluation of lexical diversity indices: Assessing length effects. *Journal of Speech, Language, and Hearing Research*, 58(3):840–852.
- Florio, K., Basile, V., Polignano, M., Basile, P., and Patti, V. (2020). Time of Your Hate: The Challenge of Time in Hate Speech Detection on Social Media. *Applied Sciences*, 10(12):4180.
- Founta, A., Djouvas, C., Chatzakou, D., Leontiadis, I., Blackburn, J., Stringhini, G., Vakali, A., Sirivianos, M., and Kourtellis, N. (2018). Large Scale Crowdsourcing and Characterization of Twitter Abusive Behavior. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 12.
- Ghanadian, H., Nejadgholi, I., and Osman, H. A. (2024). Socially aware synthetic data generation for suicidal ideation detection using large language models. *IEEE Access*, 12:14350–14363.
- Gomez, R., Gibert, J., Gomez, L., and Karatzas, D. (2020). Exploring hate speech detection in multimodal publications. *Proceedings - 2020 IEEE Winter Conference on Applications of Computer Vision, WACV 2020*, pages 1459–1467.
- Graham, M. and Ferrari, F. (2022). *Digital Work in the Planetary Market*. MIT Press, Cambridge, Massachusetts.
- Guest, E., Vidgen, B., Mittos, A., Sastry, N., Tyson, G., and Margetts, H. (2021). An Expert Annotated Dataset for the Detection of Online Misogyny. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1336–1350, Online. Association for Computational Linguistics.

- Guiraud, P. (1960). *Problèmes et méthodes de la statistique linguistique*. Synthese library. Springer Dordrecht, Dordrecht, Netherlands, first edition.
- Hartvigsen, T., Gabriel, S., Palangi, H., Sap, M., Ray, D., and Kamar, E. (2022). ToxiGen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. In Muresan, S., Nakov, P., and Villavicencio, A., editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3309–3326, Dublin, Ireland. Association for Computational Linguistics.
- He, P., Gao, J., and Chen, W. (2023). DeBERTav3: Improving deBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations*.
- Herdan, G. (1960). *Type-token mathematics: A textbook of mathematical linguistics*. Janua linguarum. Mouton, The Hague, Netherlands.
- Holtzman, A., Buys, J., Du, L., Forbes, M., and Choi, Y. (2020). The curious case of neural text degeneration. In *ICLR 2020 : Eighth International Conference on Learning Representations*.
- Ibañez, M., Sapinit, R., Reyes, L. A., Hussien, M., Imperial, J. M., and Rodriguez, R. (2021). Audio-based hate speech classification from online short-form videos. In *2021 International Conference on Asian Language Processing (IALP)*, pages 72–77.
- Iyer, S., Lin, X. V., Pasunuru, R., Mihaylov, T., Simig, D., Yu, P., Shuster, K., Wang, T., Liu, Q., Koura, P. S., et al. (2022). Opt-impl: Scaling language model instruction meta learning through the lens of generalization.

- Jackson, J. (2024). What is content moderation? *TBIJ*.
- Jahan, M. S. and Oussalah, M. (2023). A systematic review of hate speech automatic detection using natural language processing. *Neurocomputing*, 546:126232.
- Johnson, W. (1944). Studies in language behavior: A program of research. *Psychological Monographs*, 56(2):1–15.
- Jordon, J., Szpruch, L., Houssiau, F., Bottarelli, M., Cherubin, G., Maple, C., Cohen, S. N., and Weller, A. (2022). Synthetic data – what, why and how?
- Juuti, M., Gröndahl, T., Flanagan, A., and Asokan, N. (2020). A little goes a long way: Improving toxic language classification despite data scarcity. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2991–3009, Online. Association for Computational Linguistics.
- Kennedy, B., Atari, M., Davani, A. M., Yeh, L., Omrani, A., Kim, Y., Coombs, K., Havaladar, S., Portillo-Wightman, G., Gonzalez, E., Hoover, J., Azatian, A., Hussain, A., Lara, A., Cardenas, G., Omary, A., Park, C., Wang, X., Wijaya, C., Zhang, Y., Meyerowitz, B., and Dehghani, M. (2022). Introducing the Gab Hate Corpus: defining and applying hate-based rhetoric to social media posts at scale. *Language Resources and Evaluation*, 56(1):79–108.
- Kennedy, C. J., Bacon, G., Sahn, A., and von Vacano, C. (2020). Constructing interval variables via faceted Rasch measurement and multi-task deep learning: a hate speech application. arXiv:2009.10277 [cs].
- Kirk, H., Birhane, A., Vidgen, B., and Derczynski, L. (2022a). Handling and presenting harmful text in NLP research. In Goldberg, Y.,

- Kozareva, Z., and Zhang, Y., editors, *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 497–510, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Kirk, H., Vidgen, B., and Hale, S. (2022b). Is more data better? rethinking the importance of efficiency in abusive language detection with transformers-based active learning. In Kumar, R., Ojha, A. K., Zampieri, M., Malmasi, S., and Kadar, D., editors, *Proceedings of the Third Workshop on Threat, Aggression and Cyberbullying (TRAC 2022)*, pages 52–61, Gyeongju, Republic of Korea. Association for Computational Linguistics.
- Kumar, R., Reganti, A. N., Bhatia, A., and Maheshwari, T. (2018). Aggression-annotated corpus of Hindi-English code-mixed data. In Calzolari, N., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Hasida, K., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., Piperidis, S., and Tokunaga, T., editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).
- Kumar, V., Choudhary, A., and Cho, E. (2020). Data augmentation using pre-trained transformer models. In Campbell, W. M., Waibel, A., Hakkani-Tur, D., Hazen, T. J., Kilgour, K., Cho, E., Kumar, V., and Glaude, H., editors, *Proceedings of the 2nd Workshop on Life-long Learning for Spoken Language Systems*, pages 18–26, Suzhou, China. Association for Computational Linguistics.
- Leonardelli, E. and Casula, C. (2023). Dh-fbk at hodi: Multi-task learning with classifier ensemble agreement, oversampling and synthetic data. In *Proceedings of EVALITA 2023: 8th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian*, volume 3473.

- Leonardelli, E., Menini, S., Palmero Aprosio, A., Guerini, M., and Tonelli, S. (2021). Agreeing to disagree: Annotating offensive language datasets with annotators' disagreement. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t., editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10528–10539, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Lhoest, Q., Villanova del Moral, A., Jernite, Y., Thakur, A., von Platen, P., Patil, S., Chaumond, J., Drame, M., Plu, J., Tunstall, L., Davison, J., Šaško, M., Chhablani, G., Malik, B., Brandeis, S., Le Scao, T., Sanh, V., Xu, C., Patry, N., McMillan-Major, A., Schmid, P., Gugger, S., Delangue, C., Matussière, T., Debut, L., Bekman, S., Cistac, P., Goehringer, T., Mustar, V., Lagunas, F., Rush, A., and Wolf, T. (2021). Datasets: A community library for natural language processing. In Adel, H. and Shi, S., editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Li, Z., Zhu, H., Lu, Z., and Yin, M. (2023). Synthetic data generation with large language models for text classification: Potential and limitations. In Bouamor, H., Pino, J., and Bali, K., editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10443–10461, Singapore. Association for Computational Linguistics.
- Lin, B. Y., Zhou, W., Shen, M., Zhou, P., Bhagavatula, C., Choi, Y., and Ren, X. (2020). CommonGen: A constrained text generation challenge for generative commonsense reasoning. In Cohn, T., He, Y., and Liu, Y., editors, *Findings of the Association for Computational Linguistics: EMNLP*



- 2020, pages 1823–1840, Online. Association for Computational Linguistics.
- Liu, A., Swayamdipta, S., Smith, N. A., and Choi, Y. (2022). WANLI: Worker and AI Collaboration for Natural Language Inference Dataset Creation. *arXiv:2201.05955 [cs]*.
- Liu, R., Xu, G., Jia, C., Ma, W., Wang, L., and Vosoughi, S. (2020). Data Boost: Text Data Augmentation Through Reinforcement Learning Guided Conditional Generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9031–9041, Online. Association for Computational Linguistics.
- Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. *ArXiv*.
- Locatelli, D., Damo, G., and Nozza, D. (2023). A cross-lingual study of homotransphobia on twitter. In *Proceedings of the First Workshop on Cross-Cultural Considerations in NLP (C3NLP)*, pages 16–24.
- Maas, H.-D. (1972). Über den zusammenhang zwischen wortschatzumfang und länge eines textes. *Zeitschrift für Literaturwissenschaft und Linguistik*, 2(8):73.
- Madukwe, K. J., Gao, X., and Xue, B. (2022). Token replacement-based data augmentation methods for hate speech detection. *World Wide Web*.
- Marivate, V. and Sefara, T. (2020). Improving short text classification through global augmentation methods. In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, pages 385–399. Springer.

- Matzken, C., Eger, S., and Habernal, I. (2023). Trade-offs between fairness and privacy in language modeling. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6948–6969, Toronto, Canada. Association for Computational Linguistics.
- McCarthy, P. M. (2005). *An assessment of the range and usefulness of lexical diversity measures and the potential of the measure of textual, lexical diversity (MTLD)*. PhD thesis, The University of Memphis.
- McIntyre, N. (2023). Toxicity and trauma: a day in the life of a dating app moderator. *TBIJ*.
- McIntyre, N., Bradbury, R., and Perrigo, B. (2022). Behind TikTok’s boom: A legion of traumatised, \$10-a-day content moderators. *TIME*.
- Miller, G. A. (1992). WordNet: A lexical database for English. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*.
- Nouri, N. (2022). Data augmentation with dual training for offensive span detection. In Carpuat, M., de Marneffe, M.-C., and Meza Ruiz, I. V., editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2569–2575, Seattle, United States. Association for Computational Linguistics.
- Nozza, D. (2021). Exposing the limits of zero-shot cross-lingual hate speech detection. In Zong, C., Xia, F., Li, W., and Navigli, R., editors, *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.

- Nozza, D. and Hovy, D. (2023). The state of profanity obfuscation in natural language processing scientific publications. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3897–3909, Toronto, Canada. Association for Computational Linguistics.
- Nozza, D., Volpetti, C., and Fersini, E. (2019). Unintended bias in misogyny detection. In *IEEE/WIC/ACM International Conference on Web Intelligence, WI '19*, page 149–155, New York, NY, USA. Association for Computing Machinery.
- Ocampo, N. B., Sviridova, E., Cabrio, E., and Villata, S. (2023). An in-depth analysis of implicit and subtle hate speech messages. In Vlachos, A. and Augenstein, I., editors, *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1997–2013, Dubrovnik, Croatia. Association for Computational Linguistics.
- Parvaresh, V. (2023). Covertly communicated hate speech: A corpus-assisted pragmatic study. *Journal of Pragmatics*, 205:63–77.
- Pellicer, L. F. A. O., Ferreira, T. M., and Costa, A. H. R. (2023). Data augmentation techniques in natural language processing. *Applied Soft Computing*, 132:109803.
- Perez, L. and Wang, J. (2017). The effectiveness of data augmentation in image classification using deep learning.
- Perrigo, B. (2022). Meta Accused Of Human Trafficking and Union-Busting in Kenya. *TIME*.
- Perrigo, B. (2023). Former TikTok Moderator Threatens Lawsuit in Kenya. *TIME*.

- Plaza-del arco, F. M., Nozza, D., and Hovy, D. (2023). Respectful or toxic? using zero-shot learning with language models to detect hate speech. In Chung, Y.-l., Röttger, P., Nozza, D., Talat, Z., and Mostafazadeh Davani, A., editors, *The 7th Workshop on Online Abuse and Harms (WOAH)*, pages 60–68, Toronto, Canada. Association for Computational Linguistics.
- Poletto, F., Basile, V., Sanguinetti, M., Bosco, C., and Patti, V. (2021). Resources and benchmark corpora for hate speech detection: A systematic review. *Language Resources and Evaluation*, 55(2):477–523.
- Post, R. (2009). 123Hate Speech. In *Extreme Speech and Democracy*. Oxford University Press.
- Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving Language Understanding by Generative Pre-Training.
- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language Models are Unsupervised Multitask Learners.
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.
- Ramponi, A., Testa, B., Tonelli, S., and Jezek, E. (2022). Addressing religious hate online: from taxonomy creation to automated detection. *PeerJ Computer Science*, 8:e1128.
- Ramponi, A. and Tonelli, S. (2022). Features or spurious artifacts? data-centric baselines for fair and robust hate speech detection. In Carpuat, M., de Marneffe, M.-C., and Meza Ruiz, I. V., editors, *Proceedings of the*

- 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 3027–3040, Seattle, United States. Association for Computational Linguistics.
- Rao, A., Khandelwal, A., Tanmay, K., Agarwal, U., and Choudhury, M. (2023). Ethical reasoning over moral alignment: A case and framework for in-context ethical policies in LLMs. In Bouamor, H., Pino, J., and Bali, K., editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13370–13388, Singapore. Association for Computational Linguistics.
- Rawat, C., Sarkar, A., Singh, S., Alvarado, R., and Rasberry, L. (2019). Automatic detection of online abuse and analysis of problematic users in wikipedia. In *2019 Systems and Information Engineering Design Symposium (SIEDS)*, pages 1–6. IEEE.
- Roberts, S. T. (2019). *Behind the Screen: Content Moderation in the Shadows of Social Media*. Yale University Press.
- Rosenthal, S., Atanasova, P., Karadzhov, G., Zampieri, M., and Nakov, P. (2021). SOLID: A Large-Scale Semi-Supervised Dataset for Offensive Language Identification. *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 915–928.
- Röttger, P., Vidgen, B., Nguyen, D., Waseem, Z., Margetts, H., and Pierrehumbert, J. (2021). HateCheck: Functional tests for hate speech detection models. In Zong, C., Xia, F., Li, W., and Navigli, R., editors, *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 41–58, Online. Association for Computational Linguistics.

- Sachdeva, P., Barreto, R., Bacon, G., Sahn, A., von Vacano, C., and Kennedy, C. (2022). The measuring hate speech corpus: Leveraging rasch measurement theory for data perspectivism. In Abercrombie, G., Basile, V., Tonelli, S., Rieser, V., and Uma, A., editors, *Proceedings of the 1st Workshop on Perspectivist Approaches to NLP @LREC2022*, pages 83–94, Marseille, France. European Language Resources Association.
- Sap, M., Card, D., Gabriel, S., Choi, Y., and Smith, N. A. (2019). The Risk of Racial Bias in Hate Speech Detection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1668–1678, Florence, Italy. Association for Computational Linguistics.
- Sap, M., Gabriel, S., Qin, L., Jurafsky, D., Smith, N. A., and Choi, Y. (2020). Social Bias Frames: Reasoning about Social and Power Implications of Language. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5477–5490, Online. Association for Computational Linguistics.
- Sarti, G., Feldhus, N., Sickert, L., and van der Wal, O. (2023). Inseq: An interpretability toolkit for sequence generation models. In Bollegala, D., Huang, R., and Ritter, A., editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 421–435, Toronto, Canada. Association for Computational Linguistics.
- Schick, T. and Schütze, H. (2021). Exploiting cloze-questions for few-shot text classification and natural language inference. In Merlo, P., Tiedemann, J., and Tsarfaty, R., editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.

- Schmidt, A. and Wiegand, M. (2017). A Survey on Hate Speech Detection using Natural Language Processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, pages 1–10, Valencia, Spain. Association for Computational Linguistics.
- Seaver, N. (2018). What Should an Anthropology of Algorithms Do? *Cultural Anthropology* 33 (3), pages 375—385.
- Sennrich, R., Haddow, B., and Birch, A. (2016). Improving neural machine translation models with monolingual data. In Erk, K. and Smith, N. A., editors, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.
- Steiger, M., Bharucha, T. J., Venkatagiri, S., Riedl, M. J., and Lease, M. (2021). The psychological well-being of content moderators: The emotional labor of commercial moderation and avenues for improving support. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, CHI '21*, New York, NY, USA. Association for Computing Machinery.
- Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.
- Talat, Z., Bingel, J., and Augenstein, I. (2021). Disembodied machine learning: On the illusion of objectivity in nlp. *ArXiv*, abs/2101.11974.
- Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu,

- J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P. S., Lachaux, M.-A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E. M., Subramanian, R., Tan, X. E., Tang, B., Taylor, R., Williams, A., Kuan, J. X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., and Scialom, T. (2023). Llama 2: Open foundation and fine-tuned chat models.
- Ulmer, D., Hardmeier, C., and Frellsen, J. (2022). deep-significance-easy and meaningful statistical significance testing in the age of neural networks. *arXiv preprint arXiv:2204.06815*.
- VanderPlas, J., Granger, B., Heer, J., Moritz, D., Wongsuphasawat, K., Satyanarayan, A., Lees, E., Timofeev, I., Welsh, B., and Sievert, S. (2018). Altair: Interactive statistical visualizations for python. *Journal of Open Source Software*, 3(32):1057.
- Vargas, F. A., Carvalho, I., de Góes, F. R., Benevenuto, F., and Pardo, T. A. S. (2021). Building an Expert Annotated Corpus of Brazilian Instagram Comments for Hate Speech and Offensive Language Detection. *arXiv:2103.14972 [cs]*. arXiv: 2103.14972.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is All you Need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Vidgen, B. and Derczynski, L. (2020). Directions in abusive language



- training data, a systematic review: Garbage in, garbage out. *PLOS ONE*, 15(12):e0243300.
- Vidgen, B., Harris, A., Nguyen, D., Tromble, R., Hale, S., and Margetts, H. (2019). Challenges and frontiers in abusive content detection. In Roberts, S. T., Tetreault, J., Prabhakaran, V., and Waseem, Z., editors, *Proceedings of the Third Workshop on Abusive Language Online*, pages 80–93, Florence, Italy. Association for Computational Linguistics.
- Vijayakumar, A. K., Cogswell, M., Selvaraju, R. R., Sun, Q., Lee, S., Crandall, D. J., and Batra, D. (2016). Diverse beam search: Decoding diverse solutions from neural sequence models. *Salon des Refusés*, 1.
- Wang, Y., Mishra, S., Alipoormolabashi, P., Kordi, Y., Mirzaei, A., Naik, A., Ashok, A., Dhanasekaran, A. S., Arunkumar, A., Stap, D., Pathak, E., Karamanolakis, G., Lai, H., Purohit, I., Mondal, I., Anderson, J., Kuznia, K., Doshi, K., Pal, K. K., Patel, M., Moradshahi, M., Parmar, M., Purohit, M., Varshney, N., Kaza, P. R., Verma, P., Puri, R. S., Karia, R., Doshi, S., Sampat, S. K., Mishra, S., Reddy A, S., Patro, S., Dixit, T., and Shen, X. (2022). Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In Goldberg, Y., Kozareva, Z., and Zhang, Y., editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Waqas, A., Salminen, J., Jung, S.-g., Almerexhi, H., and Jansen, B. J. (2019). Mapping online hate: A scientometric analysis on research trends and hotspots in research on online hate. *PLOS ONE*, 14(9):e0222194. Publisher: Public Library of Science.
- Waseem, Z. and Hovy, D. (2016). Hateful symbols or hateful people?

- predictive features for hate speech detection on Twitter. In Andreas, J., Choi, E., and Lazaridou, A., editors, *Proceedings of the NAACL Student Research Workshop*, pages 88–93, San Diego, California. Association for Computational Linguistics.
- Wei, J., Bosma, M., Zhao, V., Guu, K., Yu, A. W., Lester, B., Du, N., Dai, A. M., and Le, Q. V. (2022). Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.
- Wei, J. and Zou, K. (2019). EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In Inui, K., Jiang, J., Ng, V., and Wan, X., editors, *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.
- Whitney, C. D. and Norman, J. (2024). Real risks of fake data: Synthetic data, diversity-washing and consent circumvention. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency, FAccT '24*, page 1733–1744, New York, NY, USA. Association for Computing Machinery.
- Wiegand, M., Ruppenhofer, J., and Kleinbauer, T. (2019). Detection of Abusive Language: the Problem of Biased Datasets. In Burstein, J., Doran, C., and Solorio, T., editors, *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, Minneapolis, Minnesota. Association for Computational Linguistics.
- Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cis-

- tas, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. (2020). Transformers: State-of-the-art natural language processing. In Liu, Q. and Schlangen, D., editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Wullach, T., Adler, A., and Minkov, E. (2021). Fight fire with fire: Fine-tuning hate detectors using large samples of generated hate speech. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t., editors, *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4699–4705, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Xu, A., Pathak, E., Wallace, E., Gururangan, S., Sap, M., and Klein, D. (2021). Detoxifying language models risks marginalizing minority voices. In Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tur, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T., and Zhou, Y., editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2390–2397, Online. Association for Computational Linguistics.
- Yang, J., Jin, H., Tang, R., Han, X., Feng, Q., Jiang, H., Zhong, S., Yin, B., and Hu, X. (2024). Harnessing the power of llms in practice: A survey on chatgpt and beyond. *ACM Trans. Knowl. Discov. Data*, 18(6).
- Yu, D., Naik, S., Backurs, A., Gopi, S., Inan, H. A., Kamath, G., Kulkarni, J., Lee, Y. T., Manoel, A., Wutschitz, L., et al. (2022). Differentially

- private fine-tuning of language models. In *International Conference on Learning Representations (ICLR)*.
- Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., and Kumar, R. (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: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1415–1420, Minneapolis, Minnesota. Association for Computational Linguistics.
- Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., and Kumar, R. (2019b). SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media (OffensEval). In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 75–86, Minneapolis, Minnesota, USA. Association for Computational Linguistics.
- Zampieri, M., Nakov, P., Rosenthal, S., Atanasova, P., Karadzhov, G., Mubarak, H., Derczynski, L., Pitenis, Z., and Çöltekin, Ç. (2020). SemEval-2020 task 12: Multilingual offensive language identification in social media (OffensEval 2020). In Herbelot, A., Zhu, X., Palmer, A., Schneider, N., May, J., and Shutova, E., editors, *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1425–1447, Barcelona (online). International Committee for Computational Linguistics.
- Zeinert, P., Inie, N., and Derczynski, L. (2021). Annotating Online Misogyny. 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 3181–3197, Online. Association for Computational Linguistics.
- Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X. V., Mihaylov, T., Ott, M., Shleifer, S., Shuster,

K., Simig, D., Koura, P. S., Sridhar, A., Wang, T., and Zettlemoyer, L. (2022). Opt: Open pre-trained transformer language models.

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., Liu, P., Nie, J.-Y., and Wen, J.-R. (2023). A survey of large language models.

Zhou, X., Sap, M., Swayamdipta, S., Choi, Y., and Smith, N. (2021). Challenges in automated debiasing for toxic language detection. In Merlo, P., Tiedemann, J., and Tsarfaty, R., editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3143–3155, Online. Association for Computational Linguistics.



# Appendix A


## Variationist

Language data is at the core of a large body of work in many research fields and at their intersections. Language data is used to train large language models (LLMs) by natural language processing (NLP) practitioners, but also by linguists and social scientists to analyze human language and behavior. With a tendency in the NLP community to overlook what actually *is* in the training data of models (Bender et al., 2021), especially at the level of textual information, and how different characteristics of the data can be intertwined, we propose a tool that can help in inspecting language data in a straightforward and highly customizable manner.

While some language data exploration tools already exist, especially English-centric corpus linguistics tools (Anthony, 2013), these cannot typically handle different types of textual *units* (e.g., tokens, bigrams, characters, and more) and multiple *variables* or combinations thereof, only offering surface-level *metrics* that are not easily customizable, and providing low-dimensional visualization. On the other hand, modern analysis tools in NLP mainly focus on interpreting model outputs (Sarti et al., 2023; Attanasio et al., 2023, *inter alia*) rather than exploring the language data in itself.

VARIATIONIST<sup>1</sup> aims to fill this gap, offering the chance to researchers

---

<sup>1</sup>: <https://github.com/dhfbk/variationist>.

from diverse disciplines to easily explore the intersections between variables in textual corpora in a plethora of different configurations in a unified manner. Additionally, VARIATIONIST allows users to plug in their own custom tokenization functions and metrics in a seamless way, opening up an unlimited number of analysis configurations in just a few lines of code, and going beyond English-centric assumptions on what the definition of a unit in language actually is. In this work, we used VARIATIONIST to carry out lexical analyses in Chapters 4, 6, and 7 in order to compare the most informative tokens for the *abusive* class in gold and synthetic data, which allowed us to study lexical aspects of the usage of synthetic data more in depth in our experiments.

## A.1 Tool Design

In this section, we present the overall design and aim of VARIATIONIST. In Section A.1.1 we detail the guiding design principles, whereas in Section A.1.2 we summarize the core elements and functionalities around which VARIATIONIST is designed.

### A.1.1 Design Principles

The guiding design principles of VARIATIONIST are summarized in the following:

- **Ease of use:** VARIATIONIST is crafted to be as accessible and customizable as possible, to serve researchers from a wide range of fields who are interested in exploring textual data;
- **Modularity:** VARIATIONIST is built out of small building blocks, allowing users to pick and choose their desired features and metrics without running unnecessary calculations;



- **Extensibility:** VARIATIONIST is designed to be easily extended. By virtue of its intrinsic modularity, it is conceived to let users select their preferred features, and import their own custom tokenizers and metrics into the tool.

### A.1.2 Core Elements and Functionalities

VARIATIONIST is designed around a set of core elements useful for computation and visualization. We provide details on each of them in the following.

**DATASETS** The main input for the analysis. Datasets can be provided in the form of *i*) tab-separated (tsv) or *ii*) comma-separated (csv) files, or *iii*) pre-computed pandas dataframes. Moreover, *iv*) any dataset from the 🤗 Hugging Face Datasets (Lhoest et al., 2021) repository can be directly imported for analysis and visualization, too.

**TEXTS** The subset of the input data, in the form of column names or indices, containing textual data. While in most scenarios only a single text column is needed, VARIATIONIST handles up to two columns at once in the analysis. This is especially useful for exploring similarities and differences between texts associated to the same labels and/or metadata.

**UNITS** The language unit of interest, which can be anything from characters to “words” (whatever their definition may be) and longer sequences. VARIATIONIST seamlessly supports  $n$ -grams (i.e.,  $n$  contiguous language units) and co-occurrences of  $n$  units (not necessarily contiguous) that fall within a user-defined window size, with optional duplicate handling. For creating units, we rely on either built-in, publicly available, or user-defined tokenizers (see below). Units may

optionally undergo preprocessing with lowercasing and stopword removal. In the latter case, the user can rely on off-the-shelf stopword lists across multiple languages from the `stopwords-iso`<sup>2</sup> package, provide their own lists directly or as files, or combine them.

**TOKENIZERS** Since the driver for the computation is a language unit, we need ways to segment texts into desired units. `VARIATIONIST` allows the user to leverage *i*) a default whitespace tokenizer that goes beyond Latin characters, *ii*) any tokenizer from 🤗 Hugging Face Tokenizers (Wolf et al., 2020), or *iii*) a custom tokenizer. This way we avoid any assumptions on what actually *is* a language unit, also broaden the applicability of `VARIATIONIST` to a wide range of language varieties.

In this thesis, we used both whitespace tokenization with single word units, in order for the results of our analysis to be more easily interpretable, and sub-word tokenization using 🤗 Hugging Face Tokenizers to reflect the token representations of the models we studied. However, it would be interesting to also study lexical with regards to n-grams or co-occurrences, which could offer more insights into the language used by language models and how it differs from actual human uses of language.

**VARIABLES** Variables are essential components for computing association metrics with language units. While variables in NLP typically translate to human-annotated “labels”, those may be naturally generalized to any kind of meta-information associated to textual data (e.g., genres, dates, spatial information, sociodemographic characteristics of annotators or authors). `VARIATIONIST` natively supports a potentially

---

<sup>2</sup><https://github.com/stopwords-iso>.

unlimited number of variable combinations during analysis. Due to the variety of data types and semantic meanings that variables may take, each variable (i.e., column name) is defined through the following two attributes:

- **Variable *types***: the type of the variable for representation purposes. It can be either *nominal* (i.e., categorical variable without an intrinsic ordering/ranking), *ordinal* (variable that can be ordered/ranked), *quantitative* (numerical variable – either discrete or continuous – which may take any value), or *coordinate* (position of a point on the Earth’s surface, i.e., latitude or longitude);
- **Variable *semantics***: how the variable must be interpreted for visualization purposes. It may be either *temporal* (e.g., variables such as date or time), *spatial* (e.g., *coordinate* variables or *nominal* variables with spatial semantics such as countries, states, or provinces), or *general* (any variable that does not fall in the aforementioned categories).

**METRICS** The methods used for measuring associations between language units and a potentially unlimited combination of variables. **VARIATIONIST** includes metrics such as pointwise mutual information (PMI; Fano, 1961), its positive, normalized, and weighted variants, as well as their combinations, for a total of 8 different PMI flavors. It also includes a normalized class relevance metric based on Ramponi and Tonelli (2022) in its positive, weighted, and positive weighted versions. Besides unit–variables association metrics, **VARIATIONIST** also includes lexical diversity measures such as type-token ratio (TTR; Johnson, 1944), root TTR (Guiraud, 1960), log TTR (Herdan, 1960), and Maas’ index (Maas, 1972). Basic statistics such as frequencies, number of texts,

number of language units, duplicate instances, average text length, and vocabulary size are also provided. Finally, custom metrics can be easily defined by the user and used for subsequent analysis, therefore extending VARIATIONIST’s capabilities to specific use cases.

For example, in this work, we have used VARIATIONIST to compare gold and synthetic data with regards to abusive language detection from a lexical point of view. In practice, using our tool, this was done by selecting the *abuse/hate speech* label as the *variable*, which was treated as a *nominal* variable with *general* semantics. The main metrics that are built into VARIATIONIST that were used in the previous chapters are `npw_relevance` (one of the PMI-based metrics), and type token ratio (TTR).

**CHARTS** The visual components of the tool. VARIATIONIST orchestrates the automatic creation of interactive charts for each metric based on the combination of variable types and semantics from a previous analysis. It defines the optimal dimension or channel (e.g., `x`, `y`, `color`, `size`, `lat`, `lon`, or a dropdown component) for each variable, creating charts with up to five dimensions (of which one is reserved for the *quantitative* metric score, and the other to the *nominal* language unit). Possible charts currently include temporal line charts, choropleth maps, geographic and standard scatter plots, heatmaps, binned maps, and bar charts. For each metric, one or more charts are created (e.g., in the case of *nominal* variable types with *spatial* semantics, both a bar chart and a geographic scatter plot are created). Charts can be interactively filtered by language unit through a search input field supporting regular expressions or a drop-

```
from variationist import Inspector, InspectorArgs, Visualizer,
    VisualizerArgs

# 1) Define the inspector arguments
ins_args = InspectorArgs(text_names=["text"], var_names=["label"],
    metrics=["npw_pmi"], n_tokens=1, language="en", stopwords=True,
    lowercase=True)

# 2) Run the inspector and get the results
res = Inspector(dataset="data.tsv", args=ins_args).inspect()

# 3) Define the visualizer arguments
vis_args = VisualizerArgs(output_folder="charts", output_formats=["html"])

# 4) Create interactive charts for all metrics
charts = Visualizer(input_json=res, args=vis_args).create()
```

Figure A.1: Example showcasing the four steps for inspecting data and visualizing results using VARIATIONIST.

down menu<sup>3</sup> to smoothly explore associations between units and the variables of interest.

## A.2 Implementation and Usage

In this section, we present implementation details (Section A.2.1 and Section A.2.2) and an example usage of our Python library (Section A.2.3).

### A.2.1 User-facing Classes

There are two main elements a typical user interacts with: `Inspector` and `Visualizer`, as well as their respective `InspectorArgs` and `VisualizerArgs`, which store all of the parameters they work with.

**Inspector** The `Inspector` class takes care of orchestrating the analysis, from importing and tokenizing the data to handling variable combina-

---

<sup>3</sup>The choice depends on the chart type and its number of dimensions, with the goal of keeping the overall user experience and filtering time as smooth as possible.

tions and importing and calculating the metrics. It returns a dictionary (or a `.json` file, cf. Section A.2.2) with all the calculated metrics for each unit of language, variable, and combination thereof, according to a set of parameters that are set by the user through the `InspectorArgs`.

**InspectorArgs** Through the `InspectorArgs` class we tell `Inspector` how to carry out the analysis. While we refer the reader to our library and related resources for the full documentation, some of the analysis details that can be set using `InspectorArgs` include what texts and variable(s) of the data to focus on, whether to use  $n$ -grams or  $n$  co-occurrences (and if so, for what values of  $n$ ), what tokenizer to use, including any custom ones, the selection of metrics we want to calculate, whether and how to bin the variables, and more. In short, any preference regarding the analysis will have to go through `InspectorArgs`.

**Visualizer** The `Visualizer` class takes care of orchestrating the creation of a variety of interactive charts for each metric and variable combination associated to the language units of interest. It leverages the results and metadata from the dictionary (or `.json` file) resulting from a prior analysis using `Inspector`, creating charts up to five dimensions using the `altair` library (VanderPlas et al., 2018).<sup>4</sup> It relies on `VisualizerArgs`, a class storing specific user-defined arguments for visualization.

**VisualizerArgs** The `VisualizerArgs` class provides ways to customize the creation of charts and their serialization. In particular, it allows the user to specify whether to pre-filter the visualization based on selected language units (provided as lists) or top-scoring ones (by specifying a maximum per-variable amount), provide a `shapefile` for setting

---

<sup>4</sup>Due to the modular design of `VARIATIONIST`, we aim to integrate additional visualization libraries in future releases.

the background of spatial charts, and decide whether the charts have to be saved as files and in which format, among others.

### A.2.2 Data Interchange

The results of an Inspector analysis are either *i)* stored in a variable as a dictionary, or *ii)* serialized in a `.json` file. While the first case comes handy for direct use by the Visualizer in most cases, the second option is especially useful when dealing with large datasets and a high number of variable combinations (and possible values). Indeed, serialization will enable the results to be easily used for visualization in a later stage. Details on the structure of the interchange file are in our repository.

### A.2.3 Example Usage

Figure A.1 shows a basic usage example of VARIATIONIST, which consists of four steps: *i)* defining the `InspectorArgs`, *ii)* instantiating and running the computation with `Inspector`, *iii)* defining the `VisualizerArgs`, and finally *iv)* creating interactive charts for the previously specified metrics through the `Visualizer`. For details on all the available parameters and hands-on tutorials, we refer the reader to our resources.

## A.3 Conclusion

We introduced VARIATIONIST, a modular, customizable, and easy-to-use analysis and visualization tool that aims at helping researchers in understanding language variation and unveiling potential biases in written language corpora across many dimensions. Our tool has the potential to help researchers in answering a variety of questions across many dis-

ciplines, as well as to make it easier for research and practitioners to examine their data before using it for training models or carrying out experiments, potentially raising awareness with regards to the pivotal importance of data exploration for machine learning applications.

VARIATIONIST was a key component of the qualitative analyses of Chapters 4, 6, and 7, as it allowed us to investigate lexical aspects of synthetic data and uncovering potential issues and biases in LLM-generated texts. We believe such a tool could be useful for future research aiming at exploring textual data from a qualitative point of view and uncovering potentially hidden spurious associations that could be exploited by models.



# Appendix B

## Annotation Guidelines (Chapter 7)

In this section we report the annotation guidelines for the human evaluation of Chapter 7.

**Hate Speech** This is the definition of hate speech used for the Measuring Hate Speech Corpus (Kennedy et al., 2020):

*Does this comment contain hate speech, defined as “bias-motivated, hostile and malicious language targeted at a person/group because of their actual or perceived innate characteristics, especially when the group is unnecessarily labeled (e.g., “jew bankers”).” The comment contains hate speech if slurs or derogatory language are used toward any person or group regardless of whether or not the target is a member of the slur-target group. (e.g. a comment calling Trump a faggot or a n\*\*\*\*\* still contains hate speech).*

Possible choices are *yes, no, unclear*.

**Prompt Failure** Examples that should be labeled as *prompt failures* fall under two categories:

1. The model was not able to rewrite the message. *Examples:*

(a) “Sorry, I cannot do this. I am just an AI.”

(b) “This is a hateful message. Let’s try being respectful of everyone.”

2. Description of original gold, in which the model is not rewriting the text, but is describing it. *Examples:*

(a) “Here [USER] said something racist.”

**Grammar** This should assess the general morpho-syntactic grammaticality of the message. The register is informal, so attested language use that - although not book-correct - is common in informal scenarios should be considered correct. Code mixing and dialect use should also not be considered ungrammatical. If essential components of the utterance are missing, hindering the understanding of the message, we tend towards no. If they are missing but the message is understandable, we tend towards correctness.

*Examples:* “y’all, he don’t understand” would be correct. “She done did it” would be correct. “the was here” would be incorrect.

**World knowledge / plausibility** This should assess whether, regardless of morphosyntactic grammaticality, the message makes sense and is realistic/plausible from a semantic standpoint given our knowledge of the world.

*Examples:* “I will climb a tree with a whiteboard” would not make sense. “I got a paper cut with a steel knife” would not make sense.

**Target identity categories** If you find a message to be directed at or about a group category, regardless of hate speech presence, select all identity categories that apply.

Original question for the annotators of the MHS corpus: *Is the comment above directed at or about any individual or groups based on: Race or ethnicity, religion, national origin or citizenship status, gender, sexual orientation, age, disability status.*