Generation-Based Data Augmentation for Offensive Language Detection: Is It Worth It?

Camilla Casula[◇♣], **Sara Tonelli**[◇]
[◇]Fondazione Bruno Kessler, Trento, Italy
[♣]University of Trento, Italy
{ccasula,satonelli}@fbk.eu

Abstract

Generation-based data augmentation (DA) has been presented in several works as a way to improve offensive language detection. However, the effectiveness of generative DA has been shown only in limited scenarios, and the potential injection of biases when using generated data to classify offensive language has not been investigated. Our aim is that of analyzing the feasibility of generative data augmentation more in-depth with two main focuses. First, we investigate the robustness of models trained on generated data in a variety of data augmentation setups, both novel and already presented in previous work, and compare their performance on four widely-used English offensive language datasets that present inherent differences in terms of content and complexity. In addition to this, we analyze models using the HateCheck suite, a series of functional tests created to challenge hate speech detection systems. Second, we investigate potential lexical bias issues through a qualitative analysis of the generated data. We find that the potential positive impact of generative data augmentation on model performance is unreliable, and generative DA can also have unpredictable effects on lexical bias.

⚠ **Warning**: *this paper contains examples that may be offensive or upsetting.*

1 Introduction

Even though large language models have been found to have a tendency to encode and propagate undesirable social bias (Bender et al., 2021), going as far as generating toxic sequences starting from non-toxic prompts (Gehman et al., 2020), the use of synthetic data for offensive language detection has been found to be potentially helpful in improving models (e.g. Juuti et al. (2020); Wullach et al. (2021); D'Sa et al. (2021)). Indeed, data augmentation (DA) through generation has the potential to mitigate some of the known issues of smaller datasets, which are common in offensive language detection, such as lack of linguistic variation and risk of overfitting (Vidgen and Derczynski, 2020). Furthermore, synthetic data can overcome privacy issues related to the use of social media data obtained without user consent in research experiments. It can also mitigate dataset decay, an issue affecting reproducibility, since online messages, especially abusive ones, tend to be deleted over time, while synthetic examples do not present this issue (Klubicka and Fernández, 2018).

While generative DA has been shown to be potentially useful for the task of detecting offensive and abusive language online in multiple works, several aspects and implications of it remain unexplored. First of all, generative DA has 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 setups on the quality of the augmented data, as well as little exploration of cross-dataset or cross-domain performance. The first aim of our work is therefore that of assessing the *robustness* of models across different sources of variation as follows: *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; *iii)* we compare four different generation 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.

Our second aim for this work is that of conducting 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 PMI[1], and we test the models

[1] For this, we use the implementation by Ramponi and Tonelli (2022).

Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, pages 3359–3377
May 2-6, 2023 ©2023 Association for Computational Linguistics

trained on augmented data on the HateCheck suite (Röttger et al., 2021), which includes a series of functional tests aimed at finding model weaknesses.

2 Related Work

Model-based data augmentation exploiting large language models (LLMs) such as GPT-2 (Radford et al., 2019) has been found effective for various NLP tasks. One method that has been shown to be promising is fine-tuning GPT-2 on annotated data, and then using it to generate additional similar data. The most common approach is prepending labels to sequences during fine-tuning, and then using labels as prompts for the model to generate sequences belonging to specific classes (Anaby-Tavor et al., 2020; Tepper et al., 2020; Kumar et al., 2020).

Similar methods have also been successfully applied to abusive language classification. For instance, Juuti et al. (2020) find that DA using a fine-tuned GPT-2 model leads to performance improvements in very low-resource scenarios. 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) use GPT-2 to generate synthetic hate speech data. They find that the addition of large amounts of synthetic data helps classification when starting from datasets containing thousands of labeled instances. While D'Sa et al. (2021) follow the label-prepending approach of Anaby-Tavor et al. (2020), Wullach et al. (2021) train a separate generative model on data belonging to each class. Both approaches are found effective, but they have never been comparatively evaluated.

To our knowledge, the robustness of models trained using generation-based DA has not been analyzed in depth. While Wullach et al. (2021) test their models cross-dataset, results are presented in a setup in which models are trained on 4 datasets together and tested on a fifth one. This setup presupposes that multiple datasets can be used at once for training models. However, this might not always be the case when DA is needed, so we evaluate cross-dataset setups in which only one dataset is available for training. In addition, to our knowledge ours is the first work to pair a robustness analysis with a qualitative analysis of lexical bias in the context of generative DA for this task.

3 Data

3.1 Dataset Description

We use four English datasets annotated for offensive or abusive language for training and testing our models. These datasets have been chosen because they are widely used and they differ in terms of content, since they were created to study different aspects of offensive language. Intuitively, this should allow us to assess the out-of-domain behavior of models when doing cross-dataset testing.

Agreement [AG] This dataset by Leonardelli et al. (2021) is annotated for offensive language and agreement level among annotators. It contains over 10k tweets dealing with three widely discussed topics on Twitter: the Black Lives Matter movement, the 2020 US elections, and Covid-19. Offensive tweets constitute 31% of the dataset.[2]

Founta [FO](Founta et al., 2018). This dataset 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 using four labels: hateful (7.5%), abusive (11%), normal (59%), and spam. In order to keep a binary classification setup that is consistent with the other datasets we use in our experiments, we group the *hateful* and *abusive* classes together into one single *abusive* class, following Leonardelli et al. (2021).[3]

OLID [OL] The Offensive Language Identification Dataset (Zampieri et al., 2019). This dataset consists of 14,100 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 test set we pair with this dataset is **SOLID [SO]** (Zampieri et al., 2020), which was used in the OffensEval 2020 shared task and follows the same annotation guidelines.[4]

SBIC [SB] The Social Bias Inference Corpus (Sap et al., 2020) contains 40k posts from Twitter, Reddit, and Stormfront, of which 44.8% offensive. While this dataset provides fine-grained annotations on social biases, we only consider the binary offensive/not offensive labels in our experiments.[5]

[2] https://github.com/dhfbk/annotators-agreement-dataset

[3] https://zenodo.org/record/3678559

[4] https://sites.google.com/site/offensevalsharedtask/olid

[5] homes.cs.washington.edu/~msap/social-bias-frames/

The above datasets present different characteristics. We consider **[FO]** and **[OL]** rather easy to classify, since standard BERT-based approaches trained and tested on these datasets yield results above 0.90 macro-F1 (Zhou et al., 2021; Zampieri et al., 2020). Past works showed that, in case of **[OL]**, classifiers may perform very well because of the limited presence of ambiguous tweets (Leonardelli et al., 2021). In contrast, **[AG]** 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 ∼0.75 macro-F1 (Leonardelli et al., 2021). Finally, **[SB]** includes data from different sources, with annotations for diverse targets of hate. The best classification result reported by the authors is ∼0.80 F1 (Sap et al., 2020).

3.2 Data Splits and Preprocessing

We use the default train/test splits of each dataset, where available. For **[FO]**, which has no default splits, we randomly partition the data into train and test using an 80/20 split. For all datasets, we replace URLs and user mentions with `URL` and `@USER` respectively. We then remove all duplicates. We also remove the substring "`RT:`" from the beginning of sequences in the **[FO]** 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). Since there is a partial overlap between **[SB]** and **[FO]**, 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.

4 Methods

We aim at comparing the performance of different data augmentation setups, both novel and already employed in previous work. We test them in within-dataset and cross-dataset scenarios, to assess the impact of synthetic data on model robustness across setups. Below is an overview of the process we follow, whose specifics are detailed in Section 5.

1. We randomly undersample the training data, obtaining the data subset X consisting of n examples (Sec. 5.1).
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. 5.2) 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 preprocessed and then filtered based on the probability assigned to them by the classification model C_X (Sec. 5.3).
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 $C_{X_{aug}}$.

Model choice We focus on the generation of synthetic data using GPT-2 `large` (774M parameters) (Radford et al., 2019). [6] Some recent works exploit the generative capabilities of GPT-3 for the creation of new datasets, either in human-in-the-loop setups (Liu et al., 2022) or in very resource-intensive scenarios (Hartvigsen et al., 2022a). We choose to experiment with GPT-2 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 (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.

For generation, we fine-tune GPT-2 `large`, following Wullach et al. (2021). We use 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. Classifiers and generative models are trained for 3 epochs. 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

[6] 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.

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 a very frequent token combination 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 ensure 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 are run on a NVIDIA Quadro RTX 5000 GPU in ~80 hours total, including both training and inference for all setups.

5 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.
- **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.
- **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.

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

5.1 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 **[AG]** 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 out for validation.

5.2 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.

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.

5.3 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 ≤ 5 characters long, and normalize the generated data following the steps described in Section 3.2.

Then, we 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 $p > 0.7$ [7]

After filtering, we randomly select 2,000 generated examples from the accepted ones in each setup.

5.4 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 the performance of classifiers trained using simple 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 the training data in each setup, split evenly across labels. Using oversampling as a baseline allows us to compare more resource-intensive DA methods such as the ones we are evaluating with a simpler strategy.

[7] This is the same threshold used by Wullach et al. (2021).

Gold data: 500 examples		Test			
Train: **AGREEMENT**		**AG**	**FO**	**SB**	**SO**
No augmentation		0.655 (0.603)	0.805 (0.743)	0.543 (0.537)	0.807 (0.734)
Oversampling		**0.725** (0.623)*	**0.882** (0.768)	0.554 (0.566)	**0.875** (0.757)*
	tag-prompt	0.700 (0.638)*	0.859 (0.810)*	0.547 (0.524)	0.862 (0.804)*
Filtering:	nl-prompt	0.694 (0.638)*	0.863 (0.820)*	0.560 (0.546)	0.863 (0.806)*
$p > 0.5$	cloze-prompt	0.692 (0.634)*	0.860 (0.815)*	0.545 (0.524)	0.859 (0.803)*
	1/label	0.716 (0.656)*	0.872 (0.834)*	**0.572** (0.567)	0.874 (0.823)*
Train: **FOUNTA**		**AG**	**FO**	**SB**	**SO**
No augmentation		0.683 (0.622)	0.904 (0.874)	0.540 (0.504)	0.888 (0.844)
Oversampling		0.637 (0.585)	0.900 (0.876)	**0.589** (0.591)*	0.896 (0.841)
	tag-prompt	0.679 (0.620)	0.909 (0.881)	0.567 (0.542)	**0.897** (0.857)
Filtering:	nl-prompt	0.660 (0.611)	0.909 (0.882)	**0.589** (0.575)*	0.895 (0.854)
$p > 0.5$	cloze-prompt	**0.688** (0.626)	**0.913** (0.884)*	0.559 (0.527)	0.891 (0.850)
	1/label	0.683 (0.624)	0.910 (0.882)	0.579 (0.563)*	0.893 (0.851)
Train: **SBIC**		**AG**	**FO**	**SB**	**SO**
No augmentation		0.556 (0.413)	0.646 (0.472)	0.746 (0.780)	0.714 (0.570)
Oversampling		**0.591** (0.481)*	**0.700** (0.540)*	**0.780** (0.801)*	**0.766** (0.643)*
	tag-prompt	0.561 (0.447)	0.679 (0.531)	0.765 (0.805)*	0.744 (0.618)
Filtering:	nl-prompt	0.578 (0.449)	0.687 (0.540)	0.763 (0.803)*	0.746 (0.622)
$p > 0.5$	cloze-prompt	0.574 (0.438)	0.663 (0.497)	0.762 (0.799)*	0.737 (0.604)
	1/label	0.584 (0.477)*	0.676 (0.524)	0.771 (0.805)*	0.757 (0.636)*
Train: **OLID**		**AG**	**FO**	**SB**	**SO**
No augmentation		0.568 (0.515)	0.766 (0.676)	0.585 (0.588)	0.797 (0.707)
Oversampling		0.584 (0.570)	**0.838** (0.792)*	**0.637** (0.717)*	**0.865** (0.804)*
	tag-prompt	0.578 (0.567)	0.812 (0.755)	0.610 (0.644)	0.845 (0.786)
Filtering:	nl-prompt	0.581 (0.564)	0.811 (0.763)	0.615 (0.652)	0.838 (0.781)
$p > 0.5$	cloze-prompt	**0.586** (0.565)	0.816 (0.763)	0.618 (0.656)	0.843 (0.783)
	1/label	0.575 (0.584)	0.831 (0.791)	0.631 (0.697)	0.855 (0.810)

Table 1: Average macro-F1 scores (over 10 runs) obtained by RoBERTa-base fine-tuned on augmented data, starting with 500 gold examples. F1 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.

6 Results

In this section we report the results of our experiments. Each experiment is run 10 times, with different random seeds. The metric we use to evaluate models is macro-F1 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.[8]

Table 1 and Table 2 show the results obtained by RoBERTa-base models fine-tuned on augmented data when starting with 500 and 2,000 gold examples respectively. While for the setup in which we start with 2,000 annotated examples (Table 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 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 where 2,000 gold examples are available, there are very few significant improvements in performance when using generative data augmentation.

[8] 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: 2,000 examples			Test		
Train: **AGREEMENT**		**AG**	**FO**	**SB**	**SO**
No augmentation		0.770 (0.708)	**0.900** (0.861)	0.568 (0.580)	0.895 (0.840)
Oversampling		0.761 (0.684)	0.894 (0.848)	0.592 (0.580)*	0.877 (0.830)
	tag-prompt	**0.773** (0.714)	**0.900** (0.868)	0.582 (0.563)*	0.890 (0.840)
Filtering:	nl-prompt	0.771 (0.712)	**0.900** (0.867)	0.576 (0.555)	0.895 (0.850)
$p > 0.5$	cloze-prompt	0.771 (0.713)	**0.900** (0.868)	0.576 (0.555)	**0.896** (0.850)
	1/label	0.769 (0.712)	0.893 (0.861)	0.594 (0.585)*	0.885 (0.837)
	tag-prompt	0.766 (0.708)	0.895 (0.861)	0.590 (0.580)*	0.887 (0.840)
Filtering:	nl-prompt	0.771 (0.714)	0.898 (0.866)	0.586 (0.572)*	0.892 (0.847)
$p > 0.7$	cloze-prompt	0.769 (0.712)	0.897 (0.864)	0.586 (0.570)*	0.891 (0.846)
	1/label	0.768 (0.713)	0.894 (0.862)	**0.596** (0.586)*	0.886 (0.838)
Train: **FOUNTA**		**AG**	**FO**	**SB**	**SO**
No augmentation		0.635 (0.619)	0.910 (0.883)	0.611 (0.612)	0.904 (0.866)
Oversampling		0.628 (0.604)	0.907 (0.883)	0.615 (0.618)	0.901 (0.859)
	tag-prompt	0.645 (0.620)	0.911 (0.883)	0.614 (0.618)	0.901 (0.863)
Filtering:	nl-prompt	0.635 (0.616)	0.911 (0.885)	0.625 (0.633)	0.905 (0.868)
$p > 0.5$	cloze-prompt	0.644 (0.619)	**0.915** (0.888)	0.607 (0.607)	0.906 (0.870)
	1/label	0.633 (0.613)	0.910 (0.881)	0.612 (0.615)	0.902 (0.864)
	tag-prompt	**0.650** (0.623)	0.913 (0.885)	0.619 (0.624)	0.903 (0.865)
Filtering:	nl-prompt	0.645 (0.619)	0.914 (0.887)	0.615 (0.617)	**0.908** (0.872)
$p > 0.7$	cloze-prompt	0.640 (0.619)	0.913 (0.885)	**0.621** (0.625)	0.904 (0.866)
	1/label	0.647 (0.619)	0.914 (0.886)	0.612 (0.614)	0.907 (0.871)
Train: **SBIC**		**AG**	**FO**	**SB**	**SO**
No augmentation		0.608 (0.555)	**0.737** (0.618)	0.813 (0.844)	0.804 (0.712)
Oversampling		0.591 (0.526)	0.722 (0.590)	0.810 (0.829)	0.789 (0.683)
	tag-prompt	0.603 (0.550)	0.725 (0.597)	0.812 (0.840)	0.803 (0.708)
Filtering:	nl-prompt	0.604 (0.547)	0.730 (0.605)	**0.814** (0.844)	0.802 (0.708)
$p > 0.5$	cloze-prompt	0.608 (0.552)	0.729 (0.607)	**0.814** (0.844)	0.806 (0.714)
	1/label	0.606 (0.548)	0.725 (0.598)	0.811 (0.840)	0.800 (0.704)
	tag-prompt	0.608 (0.560)	0.733 (0.611)	0.811 (0.841)	**0.807** (0.716)
Filtering:	nl-prompt	**0.618** (0.546)	0.724 (0.593)	**0.814** (0.842)	0.801 (0.703)
$p > 0.7$	cloze-prompt	0.611 (0.555)	0.735 (0.615)	0.813 (0.844)	**0.807** (0.714)
	1/label	0.609 (0.558)	0.733 (0.612)	**0.814** (0.844)	0.804 (0.709)
Train: **OLID**		**AG**	**FO**	**SB**	**SO**
No augmentation		0.584 (0.599)	0.874 (0.841)	0.633 (0.668)	**0.897** (0.859)
Oversampling		0.576 (0.580)	0.858 (0.824)	0.637 (0.709)	0.887 (0.845)
	tag-prompt	0.570 (0.593)	0.867 (0.832)	0.636 (0.681)	0.891 (0.852)
Filtering:	nl-prompt	0.586 (0.598)	0.875 (0.841)	0.641 (0.681)	0.895 (0.856)
$p > 0.5$	cloze-prompt	**0.592** (0.603)	**0.878** (0.845)	0.638 (0.672)	**0.897** (0.861)
	1/label	0.573 (0.594)	0.871 (0.839)	**0.644** (0.687)	0.892 (0.855)
	tag-prompt	0.578 (0.597)	0.864 (0.831)	0.634 (0.675)	0.892 (0.853)
Filtering:	nl-prompt	0.581 (0.597)	0.873 (0.841)	0.642 (0.681)	0.896 (0.858)
$p > 0.7$	cloze-prompt	0.582 (0.597)	0.871 (0.839)	0.638 (0.676)	0.895 (0.857)
	1/label	0.579 (0.597)	0.872 (0.839)	0.643 (0.684)	0.895 (0.858)

Table 2: Average macro-F1 scores (over 10 runs) obtained by RoBERTa-base fine-tuned on augmented data, starting with 2,000 gold examples. F1 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.

Impact of prompting and filtering Interestingly, no prompting type seems to clearly outperform the others across setups. For instance, augmenting the Agreement **[AG]** 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 **[FO]** and **[SO]**, while when tested on **[SB]** 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. However, looking at Table 2, the situation is reversed: the model trained on **[AG]** only significantly benefits from data augmentation when tested on **[SB]** 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 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, 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 advise against considering this type of DA a reliable 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. 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.

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 7.1, we analyze the generated data to assess whether it could lead to benefits with regards to fairness, perhaps due to more representation of minorities given the higher lexical variety.

The results for BERT-based models are in general in line with those for RoBERTa-based models, although BERT-based 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. Since the overall findings are similar to those of RoBERTa-based models, we do not report BERT results in this section, but in Appendix A.

7 Qualitative Analysis

In order to estimate the quality of the generated examples and the impact of the prompting method, we randomly select a subset of 10 generated examples for every dataset / data size combination for manual analysis. We find that there are some clear differences between the prompting setups, and that the methods that exploit prompting in natural language (*nl-prompt* and *cloze-prompt*) 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 can be found in Appendix B.

7.1 Lexical Artifacts Analysis

To investigate the lexical variation between the gold data and the generated data, we use pointwise mutual information (PMI), following Ramponi and Tonelli (2022). In particular, 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.

The first tendency that can be noticed when looking at how the ranking of tokens' informativeness changes between gold and generated data is that for some of the datasets the changes are more evident (i.e. for **[AG]** and **[SB]**). For example, in the gold **[SB]** data, the word *fucking* is ranked as the 10,203rd 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 with 2,000.

While the prominence of a potentially predictive word for offensive language, like *fucking*, is 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 7.2. For further details, the full lists of top 11 informative tokens for each dataset, both gold and generated, is reported in Appendix C.

7.2 HateCheck Analysis

We explore some of the weaknesses of classifiers trained on gold and augmented data through the HateCheck suite (Röttger et al., 2021), consisting of a series of functional tests aimed at exploring the weak points of hate speech models.[9]

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 6, 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 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 augmentation tends to steer models into overfitting identity terms and slurs.

Further details on the performances of models on each HateCheck test and on the targets contained in the tests are found in Appendix D.

8 Conclusions

In this work, 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.

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. 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, we advise against using generative DA for this task, since it is computationally intensive and it does not appear to consistently make models perform better or be more robust.

[9] Since our models are aimed at detecting offensive language in general and HateCheck is focused on hate speech, a narrower phenomenon, not all tests are entirely informative in our case, such as test number 11, testing the performance on non-hateful profanities. In general, however, the labels of HateCheck tests are aligned with our task.

Limitations

The main limitation of this work is its focus on English, leaving out other languages that may benefit more from a thorough evaluation of data augmentation methods because they have fewer resources. We selected English for this work because it allowed us to evaluate the system performance on four datasets with different characteristics and a number of configurations, thanks also to the availability of language-specific GPT2, BERT and RoBERTa. We are aware that generation-based data augmentation would be potentially more useful in real low-resource scenarios, and we plan on investigating in the future whether our findings hold for other languages.

Furthermore, this work deals only with one type of data augmentation, i.e. the one using fine-tuned generators, while there are others that we left out because of space limits and that may be investigated in the future. For example, we might compare generative DA with rule-based augmentation, synonym-based approaches and backtranslation, among others, and investigate whether different data augmentation approaches present differences in terms of robustness or lexical biases. In addition, we only experiment with one generative model, while we could in the future compare generative DA using GPT-2 with other kinds of generative models, especially more advanced ones.

Another aspect left unexplored in our work is whether this type of data augmentation could help models dedicated to this task that are widely available, such as HateBERT (Caselli et al., 2021) or ToxiGen-RoBERTa (Hartvigsen et al., 2022b). While in this work we focus on scenarios in which little data is available, experimenting with these models could yield interesting results in future work with a broader scope.

Ethics Statement

In our experiments, we compare different setups in which large language models are exploited in order to artificially create more data to train models aimed at detecting offensive language. In this case, synthetic data has two main potential advantages: first, it limits the amount of data gathered from online spaces without user consent, and second, it reduces the amount of manual annotation required to create labeled datasets for offensive language detection, which can have a negative psychological impact on annotators (Riedl et al., 2020).

While using generative models to augment data can in some cases be beneficial for classification performance, the sequences generated by these models can exhibit unpredictable characteristics that exacerbate existing bias or produce new forms of it. Given that the improvements provided by generative DA are inconsistent, there are no clear advantages to this method when considering its potential risks. As a consequence, we advise against deploying models trained on generated data in practice.

Since the main contribution of our work is not a novel model or algorithm, but rather an evaluation of different approaches for generative data augmentation, we share as much information as we can for the reproducibility of our experiments, but we choose not to release the code openly to the public, in order to limit potential misuse of models that can generate offensive language. We also choose not to publicly release the generated data for various reasons. As shown by our results, the generated examples are not reliable for improving existing systems, so their utility is limited. Furthermore, the generated texts are not curated, which could result in including personal user information or harmful statements targeting specific individuals being generated. We will, however, share the data upon request to other interested researchers.

Acknowledgements

This work has been partly supported by the PROTECTOR European project (ISFP-2020-AG-PROTECT) on “Protecting places of worship” (101034216) and the STAND BY ME 2.0 project (CERV-2021-DAPHNE) on “Stop gender-based violence by addressing masculinities and changing behaviour of young people through human rights education” (101049386).

References

Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. 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.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Trans-*

parency, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

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

Eustasio Del Barrio, Juan A Cuesta-Albertos, and Carlos Matrán. 2018. An optimal transportation approach for assessing almost stochastic order. In *The Mathematics of the Uncertain*, pages 33–44. Springer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Rotem Dror, Segev Shlomov, and Roi Reichart. 2019. Deep dominance - how to properly compare deep neural models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2773–2785, Florence, Italy. Association for Computational Linguistics.

Ashwin Geet D'Sa, Irina Illina, Dominique Fohr, Dietrich Klakow, and Dana Ruiter. 2021. Exploring Conditional Language Model Based Data Augmentation Approaches for Hate Speech&#xa0;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.

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

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A Smith. 2020. RealToxicityPrompts: Evaluating Neural Toxic Degeneration in Language Models. pages 3356–3369.

Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. 2022a. ToxiGen: A Large-Scale Machine-Generated Dataset for Adversarial and Implicit Hate Speech Detection. *arXiv:2203.09509 [cs]*.

Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. 2022b. ToxiGen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. In *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.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The Curious Case of Neural Text Degeneration. In *arXiv:1904.09751 [Cs]*.

Mika Juuti, Tommi Gröndahl, Adrian Flanagan, and N. Asokan. 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.

Filip Klubicka and Raquel Fernández. 2018. Examining a hate speech corpus for hate speech detection and popularity prediction. In *Proceedings of 4REAL Workshop - Workshop on Replicability and Reproducibility of Research Results in Science and Technology of Language*.

Varun Kumar, Ashutosh Choudhary, and Eunah Cho. 2020. Data Augmentation using Pre-trained Transformer Models. In *Proceedings of the 2nd Workshop on Life-long Learning for Spoken Language Systems*, pages 18–26, Suzhou, China. Association for Computational Linguistics.

Elisa Leonardelli, Stefano Menini, Alessio Palmero Aprosio, Marco Guerini, and Sara Tonelli. 2021. Agreeing to disagree: Annotating offensive language datasets with annotators' disagreement. In *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.

Alisa Liu, Swabha Swayamdipta, Noah A. Smith, and Yejin Choi. 2022. WANLI: Worker and AI Collaboration for Natural Language Inference Dataset Creation. *arXiv:2201.05955 [cs]*.

Ruibo Liu, Guangxuan Xu, Chenyan Jia, Weicheng Ma, Lili Wang, and Soroush Vosoughi. 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.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *ArXiv*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners.

Alan Ramponi and Sara Tonelli. 2022. Features or spurious artifacts? data-centric baselines for fair and robust hate speech detection. In *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.

Martin J Riedl, Gina M Masullo, and Kelsey N Whipple. 2020. The downsides of digital labor: Exploring the toll incivility takes on online comment moderators. *Computers in Human Behavior*, 107:106262.

Paul Röttger, Bertie Vidgen, Dong Nguyen, Zeerak Waseem, Helen Margetts, and Janet Pierrehumbert. 2021. HateCheck: Functional tests for hate speech detection models. 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 41–58, Online. Association for Computational Linguistics.

Maarten Sap, Saadia Gabriel, Lianhui Qin, Dan Jurafsky, Noah A. Smith, and Yejin Choi. 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.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *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.

Naama Tepper, Esther Goldbraich, Naama Zwerdling, George Kour, Ateret Anaby Tavor, and Boaz Carmeli. 2020. Balancing via Generation for Multi-Class Text Classification Improvement. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1440–1452, Online. Association for Computational Linguistics.

Dennis Ulmer, Christian Hardmeier, and Jes Frellsen. 2022. deep-significance-easy and meaningful statistical significance testing in the age of neural networks. *arXiv preprint arXiv:2204.06815*.

Bertie Vidgen and Leon Derczynski. 2020. Directions in abusive language training data, a systematic review: Garbage in, garbage out. *PLOS ONE*, 15(12):e0243300.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Tomer Wullach, Amir Adler, and Einat Minkov. 2021. Fight fire with fire: Fine-tuning hate detectors using large samples of generated hate speech. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4699–4705, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. 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.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. 2020. SemEval-2020 Task 12: Multilingual Offensive Language Identification in Social Media (OffensEval 2020). In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1425–1447, Barcelona (online). International Committee for Computational Linguistics.

Xuhui Zhou, Maarten Sap, Swabha Swayamdipta, Yejin Choi, and Noah Smith. 2021. Challenges in automated debiasing for toxic language detection. In *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.

A BERT-Based Models Results

In this section, we present the results of BERT-based models in the setup where we start with 500 gold examples in Table 3 and with 2,000 gold examples in Table 4.

Gold data: 500 examples		Test			
Train: **AGREEMENT**		**AG**	**FO**	**SB**	**SO**
No augmentation		0.630	0.716	0.550	0.700
Oversampling		**0.696***	**0.823***	0.573	**0.825***
	tag-prompt	0.663	0.775*	0.562	0.774
Filtering:	nl-prompt	0.654	0.752	**0.584**	0.767
$p > 0.5$	cloze-prompt	0.665*	0.773*	0.554	0.780*
	1/label	0.688*	0.798*	0.575	0.797*
Train: **FOUNTA**		**AG**	**FO**	**SB**	**SO**
No augmentation		0.619	0.890	0.613	0.847
Oversampling		**0.638**	**0.906***	0.598	**0.885***
	tag-prompt	0.636	0.904	0.600	0.876*
Filtering:	nl-prompt	0.614	0.900	**0.641**	0.874*
$p > 0.5$	cloze-prompt	0.632	0.900	0.606	0.857
	1/label	0.629	0.899	0.633	0.878*
Train: **SBIC**		**AG**	**FO**	**SB**	**SO**
No augmentation		0.566	0.629	0.747	0.727
Oversampling		**0.579***	**0.682**	**0.766***	**0.756**
	tag-prompt	0.575	0.679	0.754	0.755
Filtering:	nl-prompt	0.576	0.677	0.757	0.754
$p > 0.5$	cloze-prompt	0.566	0.656	0.754	0.738
	1/label	0.574	0.664	0.762*	0.743
Train: **OLID**		**AG**	**FO**	**SB**	**SO**
No augmentation		0.555	0.757	0.635	0.770
Oversampling		0.555	**0.832***	0.653	**0.852***
	tag-prompt	0.554	0.795	0.641	0.813
Filtering:	nl-prompt	0.559	0.810*	**0.658***	0.832*
$p > 0.5$	cloze-prompt	**0.562**	0.803*	0.648	0.823
	1/label	0.537	0.805*	0.648	0.821*

Table 3: Average macro-F1 scores (over 10 runs) obtained by BERT-base-uncased fine-tuned on augmented data, starting with 500 gold examples. 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: **AGREEMENT**		**AG**	**FO**	**SB**	**SO**
No augmentation		0.756	0.894	0.573	0.891
Oversampling		0.746	0.884	0.592	0.880
	tag-prompt	0.759	0.900	0.567	0.893
Filtering:	nl-prompt	0.761	0.901*	0.567	**0.900***
$p > 0.5$	cloze-prompt	0.756	0.901*	0.572	0.899*
	1/label	0.749	0.892	0.584	0.891
	tag-prompt	0.760	0.899	0.578	0.893
Filtering:	nl-prompt	0.760	0.899	0.572	0.897*
$p > 0.7$	cloze-prompt	**0.762**	**0.902***	0.572	0.898*
	1/label	0.753	0.897	**0.593***	0.893
Train: **FOUNTA**		**AG**	**FO**	**SB**	**SO**
No augmentation		0.616	0.913	0.628	0.905
Oversampling		0.635*	0.911	0.617	0.899
	tag-prompt	0.634*	0.914	0.621	0.905
Filtering:	nl-prompt	0.632	0.914	0.627	0.905
$p > 0.5$	cloze-prompt	0.617	0.914	**0.630**	0.903
	1/label	0.629	0.914	0.628	0.904
	tag-prompt	**0.636***	0.913	0.624	0.905
Filtering:	nl-prompt	0.634	**0.915**	0.629	0.904
$p > 0.7$	cloze-prompt	0.633	0.914	**0.630**	**0.907**
	1/label	0.629	0.913	0.627	0.903
Train: **SBIC**		**AG**	**FO**	**SB**	**SO**
No augmentation		0.589	**0.743**	0.806	0.807
Oversampling		0.588	0.716	0.799	0.786
	tag-prompt	0.584	0.742	0.806	**0.809**
Filtering:	nl-prompt	0.594	0.734	0.807	0.802
$p > 0.5$	cloze-prompt	0.593	0.735	0.806	0.802
	1/label	0.586	0.739	0.804	0.800
	tag-prompt	0.582	**0.743**	**0.809**	0.806
Filtering:	nl-prompt	0.588	0.734	0.806	0.803
$p > 0.7$	cloze-prompt	**0.598**	0.742	0.807	0.807
	1/label	0.591	0.732	0.806	0.803
Train: **OLID**		**AG**	**FO**	**SB**	**SO**
No augmentation		0.562	0.874	0.653	0.897
Oversampling		0.549	0.859	0.661	0.885
	tag-prompt	0.555	0.862	0.662	0.890
Filtering:	nl-prompt	0.553	0.868	0.668*	0.893
$p > 0.5$	cloze-prompt	0.568	0.875	0.659	0.897
	1/label	0.541	0.858	0.669*	0.885
	tag-prompt	0.555	0.862	0.663	0.892
Filtering:	nl-prompt	0.561	0.873	0.665*	0.896
$p > 0.7$	cloze-prompt	**0.575**	**0.879**	0.658	**0.898**
	1/label	0.548	0.863	**0.671**	0.889

Table 4: Average macro-F1 scores (over 10 runs) obtained by BERT-base-uncased fine-tuned on augmented data, starting with 2,000 gold examples. 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.

prompting	dataset	generated text	label
tag-prompt	[AG]	#SEP #BlackLivesMatter #BIDEN #MAGA URL	0
	[FO]	∼∼∼@USER	0
	[SB]	∼∼∼∼∼∼∼∼∼∼∼∼∼∼∼∼∼∼∼∼∼∼∼∼∼∼∼∼∼∼∼∼∼∼∼∼∼∼ [...]	0
	[OL]	***********@USER You’re fucking awesome	0
nl-prompt	[AG]	@USER It is time to end this corrupt criminal republic #DemocRATting #BlackLivesMatter URL	0
	[FO]	??????????????????????????????????	1
	[SB]	I hate all of you who call yourself patriots. You are useless.	1
	[OL]	@USER @USER you are not a Christian..you are a hypocrite	1
cloze-prompt	[AG]	”I love how the media is pushing the idea that the protesters are racist	0
	[FO]	USER: hey, u should really try and figure out what is going on in the world of 4chan	0
	[SB]	If you can’t get a girl to like you, you’re a loser...	1
	[OL]	@USER @USER He is the worst	0
1/label	[AG]	Coronavirus was first identified in 1984 in the U.S. as a contaminant 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
	[FO]	#6a8a2a color RGB value is (135,180,225). #6b98f6 color hex is (130,180,.25). the number is on its way to being determined so I don’t need to change it [...]	0
	[SB]	"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 possibility 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
	[OL]	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 #ToriesOut #ToryOut’sOut#POTUSOut #MAGADump #MAGAMAG #MAGAZINEAU’S MAGA NEWSLETTER @USER @USER The way he [...]	0

Table 5: Examples of sequences generated by GPT-2 large models trained on 500 gold examples for every combination of prompting and dataset.

B Examples of Generated Texts

Examples of sequences generated by the models can be found in Table 5 for generative models trained on 500 labeled instances and in Table 6 for models trained on 2,000 labeled instances. Some texts have been truncated (signaled by “[...]”) due to space constraints. The labels reported in Tables 5 and 6 are those assigned by the RoBERTa baseline.

prompting	dataset	generated text	label
tag-prompt	[AG]	URL #Elections2020	0
	[FO]	!!!! &#128580;&#127907;&amp;&gt; I fucking hate you URL	1
	[SB]	phone 7, apple iphones 8, iphONES 8, android ips 8, blackberry ips, ixp8 iph, [...]	0
	[OL]	URL You guys are pathetic	1
nl-prompt	[AG]	@USER Because the Democrats lie, cheat, steal and intentionally create the chaos in our country. They are beyond evil!	0
	[FO]	????????? @USER We are listening..if you have any ideas please send us a tweet at URL	0
	[SB]	You're never too young to dream, you just gotta work it.	0
	[OL]	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
cloze-prompt	[AG]	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
	[FO]	@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
	[SB]	Do you think these fake news articles on vaccines and autism are getting too much play on social media?	0
	[OL]	@USER @USER @USER @USER He is just one more nut job in Congress.	1
1/label	[AG]	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
	[FO]	$4.99 Featuring all the fan favorites including: • Exclusive and classic illustrations from some of comics' most beloved artists [..]	0
	[SB]	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
	[OL]	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 6: Examples of sequences generated by GPT-2 large models trained on 2,000 gold examples for every combination of prompting and dataset.

C Lexical Artifacts

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 7, and those for the setup with 2,000 gold instances are in Table 8.

⚠ **Content warning**: *Tables 7 and 8 contain uncensored profanities and slurs.* [10]

D HateCheck

Table 9 and Table 10 present the results on HateCheck tests and targets for models in the 500 gold examples setup. Table 11 and Table 12 present the results on the functional tests and targets for models in the setup in which we start with 2,000 annotated examples.

[10] These are left uncensored for increased readability, since special characters are already used to signal boundaries of sub-word tokens, and to avoid confusion with words that are self-censored by the users.

AGREEMENT

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	bitch	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	bitch	4	fuck	4	fuck
2	user	2	user	4	fuck	2	user	6	hate
3	bitch	6	hate	6	hate	6	hate	11	shit
4	fuck	3	bitch	11	shit	3	bitch	1	fucked
5	ass	5	ass	10	stupid	5	ass	10	stupid
6	hate	1	fucked	8	idiot	1	fucked	3	bitch
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	bitch	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	bitch	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 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.

AGREEMENT									
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	bitch	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	bitch	4	fuck	3	bitch	6	hate	6	hate
4	fuck	3	bitch	9	shit	4	fuck	3	bitch
5	ass	16339	[SEP]	6	hate	3	bitch	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	bitch	15008	[SEP]	3	white	3	white	3	white
2	difference	10	##s	1	bitch	2	difference	5	sex
3	white	10203	fucking	14	##gger	15	woman	2	difference
4	##es	1	bitch	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	bitch	1	bitch
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	bitch	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 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.

	AGREEMENT						FOUNTA						SBIC						OLID					
Functionality	base	overs	tag	nl	cloze	1/label	base	overs	tag	nl	cloze	1/label	base	overs	tag	nl	cloze	1/label	base	overs	tag	nl	cloze	1/label
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	0.82	0.82	0.87	0.82	0.93
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	0.96	0.88	0.87	0.86	0.97
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	0.91	0.84	0.86	0.83	0.93
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	0.72	0.66	0.63	0.59	0.71
5 threat_dir_h	0.19	0.27	0.21	0.21	0.20	0.18	0.38	0.73	0.65	0.71	0.50	0.65	0.75	0.90	0.77	0.71	0.82	0.84	0.61	0.81	0.74	0.75	0.69	0.82
6 threat_norm_h	0.18	0.22	0.16	0.20	0.19	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	0.84	0.73	0.74	0.69	0.81
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	0.87	0.79	0.80	0.77	0.84
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	0.48	0.57	0.52	0.58	0.47
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	0.23	0.28	0.27	0.29	0.17
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	0.99	0.90	0.90	0.91	1.00
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	0.03	0.14	0.11	0.14	0.01
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	0.88	0.85	0.86	0.81	0.92
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	0.92	0.88	0.89	0.85	0.96
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	0.49	0.65	0.64	0.51	0.52
15 negate_neg_nh	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	0.13	0.43	0.30	0.33	0.31	0.44	0.28
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	0.91	0.83	0.85	0.83	0.96
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	0.94	0.84	0.85	0.82	0.93
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	0.88	0.92	0.94	0.88	0.86
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	0.93	0.89	0.92	0.89	0.88
20 counter_quote_nh	0.42	0.52	0.47	0.42	0.50	0.45	0.24	0.11	0.17	0.08	0.24	0.16	0.21	0.13	0.19	0.19	0.12	0.14	0.14	0.04	0.12	0.12	0.13	0.02
21 counter_ref_nh	0.48	0.42	0.45	0.43	0.44	0.43	0.26	0.11	0.19	0.10	0.22	0.17	0.10	0.14	0.16	0.14	0.07	0.11	0.16	0.08	0.13	0.14	0.17	0.05
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	0.23	0.28	0.27	0.32	0.20
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	0.12	0.22	0.18	0.26	0.09
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	0.16	0.22	0.22	0.25	0.16
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	0.54	0.59	0.61	0.61	0.67

Table 9: Accuracy on the first 25 functional HateCheck tests of RoBERTa models trained on 500 gold examples and on the augmented data.

	AGREEMENT						FOUNTA						SBIC						OLID					
Target	base	overs	tag	nl	cloze	1/label	base	overs	tag	nl	cloze	1/label	base	overs	tag	nl	cloze	1/label	base	overs	tag	nl	cloze	1/label
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	0.64	0.61	0.62	0.60	0.65
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	0.63	0.61	0.61	0.58	0.64
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	0.65	0.62	0.63	0.61	0.65
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	0.66	0.62	0.64	0.61	0.66
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	0.63	0.60	0.61	0.60	0.64
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	0.63	0.60	0.62	0.60	0.63
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	0.62	0.59	0.60	0.58	0.62

Table 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.

	AGREEMENT						FOUNTA						SBIC						OLID					
Functionality	**base**	**overs**	**tag**	**nl**	**cloze**	**1/label**	**base**	**overs**	**tag**	**nl**	**cloze**	**1/label**	**base**	**overs**	**tag**	**nl**	**cloze**	**1/label**	**base**	**overs**	**tag**	**nl**	**cloze**	**1/label**
1 derog_neg_emote_h	0.07	0.20	0.11	0.07	0.07	0.12	0.91	0.89	0.89	0.91	0.91	0.89	0.87	0.88	0.88	0.89	0.91	0.79	0.93	0.89	0.91	0.95	0.86	0.94
2 derog_neg_attrib_h	0.51	0.57	0.57	0.55	0.55	0.56	0.94	0.94	0.94	0.95	0.95	0.95	0.95	0.98	0.96	0.98	0.98	0.95	0.99	0.98	1.00	0.99	0.99	0.98
3 derog_dehum_h	0.57	0.66	0.67	0.64	0.62	0.66	0.90	0.89	0.88	0.89	0.89	0.92	0.96	0.98	0.96	0.97	0.96	0.92	0.97	0.95	0.97	0.98	0.96	0.98
4 derog_impl_h	0.11	0.20	0.13	0.13	0.11	0.16	0.52	0.59	0.53	0.54	0.50	0.50	0.92	0.94	0.89	0.92	0.95	0.85	0.70	0.76	0.76	0.75	0.68	0.83
5 threat_dir_h	0.23	0.46	0.34	0.25	0.28	0.34	0.82	0.87	0.86	0.84	0.87	0.84	0.95	0.97	0.93	0.95	0.96	0.92	0.87	0.90	0.87	0.92	0.85	0.95
6 threat_norm_h	0.11	0.37	0.18	0.12	0.11	0.22	0.81	0.87	0.81	0.82	0.84	0.78	0.95	0.97	0.94	0.96	0.97	0.93	0.86	0.93	0.90	0.90	0.85	0.93
7 slur_h	0.76	0.76	0.79	0.77	0.76	0.79	0.77	0.77	0.76	0.79	0.77	0.79	0.85	0.88	0.84	0.85	0.85	0.81	0.88	0.91	0.90	0.90	0.86	0.92
8 slur_homonym_nh	0.50	0.50	0.50	0.50	0.50	0.50	0.46	0.49	0.46	0.47	0.47	0.45	0.51	0.47	0.48	0.43	0.43	0.48	0.49	0.46	0.42	0.45	0.48	0.44
9 slur_reclaimed_nh	0.22	0.23	0.22	0.22	0.22	0.23	0.14	0.14	0.14	0.14	0.15	0.13	0.21	0.24	0.19	0.15	0.17	0.23	0.19	0.18	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	1.00	1.00	0.99	0.98	0.99	0.98	0.99	0.96	1.00	1.00	1.00	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	0.00	0.01	0.65	0.66	0.66	0.59	0.59	0.65	0.01	0.02	0.01	0.01	0.02	0.01
12 ref_subs_clause_h	0.36	0.46	0.39	0.40	0.38	0.45	0.88	0.87	0.87	0.87	0.90	0.87	0.96	0.98	0.96	0.98	0.97	0.94	0.95	0.95	0.94	0.97	0.93	0.96
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	0.95	0.93	0.97	0.97	0.97	0.98	0.97	0.97	0.97	0.98	0.95	0.98	0.96	0.98
14 negate_pos_h	0.02	0.05	0.03	0.03	0.02	0.06	0.43	0.52	0.46	0.46	0.45	0.46	0.83	0.93	0.80	0.89	0.88	0.79	0.49	0.59	0.57	0.63	0.51	0.68
15 negate_neg_nh	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	0.13	0.08	0.06	0.12	0.22	0.22	0.21	0.22	0.25	0.17
16 phrase_question_h	0.32	0.40	0.33	0.33	0.32	0.35	0.87	0.89	0.89	0.90	0.91	0.88	0.95	0.96	0.97	0.97	0.96	0.98	0.94	0.95	0.95	0.96	0.91	0.97
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	0.98	0.94	0.97	0.98	0.95	0.95	0.97	0.95	0.96	0.93	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	0.33	0.50	0.86	0.77	0.80	0.80	0.82	0.71
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	0.27	0.35	0.95	0.86	0.88	0.91	0.93	0.85
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	0.06	0.05	0.05	0.06	0.01	0.02	0.01	0.00	0.01	0.01
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	0.08	0.06	0.05	0.08	0.04	0.04	0.03	0.03	0.05	0.02
22 target_obj_nh	0.83	0.77	0.80	0.82	0.81	0.81	0.47	0.46	0.45	0.43	0.43	0.44	0.80	0.74	0.75	0.77	0.77	0.81	0.21	0.21	0.18	0.19	0.25	0.20
23 target_indiv_nh	0.60	0.48	0.51	0.57	0.55	0.50	0.13	0.13	0.14	0.13	0.13	0.16	0.45	0.38	0.39	0.33	0.37	0.47	0.09	0.06	0.08	0.06	0.10	0.06
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	0.35	0.33	0.38	0.15	0.14	0.12	0.10	0.16	0.12
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	0.89	0.90	0.84	0.64	0.69	0.76	0.73	0.65	0.76

Table 11: Accuracy on the first 25 functional HateCheck tests of RoBERTa models trained on 2,000 gold examples and on the augmented data.

	AGREEMENT						FOUNTA						SBIC						OLID					
Target	**base**	**overs**	**tag**	**nl**	**cloze**	**1/label**	**base**	**overs**	**tag**	**nl**	**cloze**	**1/label**	**base**	**overs**	**tag**	**nl**	**cloze**	**1/label**	**base**	**overs**	**tag**	**nl**	**cloze**	**1/label**
1 women	0.47	0.51	0.50	0.48	0.47	0.51	0.62	0.62	0.61	0.62	0.61	0.61	0.59	0.57	0.59	0.61	0.58	0.58	0.65	0.66	0.67	0.68	0.64	0.68
2 trans people	0.42	0.46	0.45	0.45	0.44	0.44	0.59	0.59	0.57	0.61	0.60	0.61	0.55	0.53	0.54	0.51	0.53	0.53	0.63	0.63	0.65	0.66	0.62	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	0.57	0.59	0.52	0.51	0.52	0.48	0.48	0.51	0.63	0.60	0.59	0.63	0.62	0.56
4 black people	0.46	0.52	0.48	0.48	0.48	0.49	0.61	0.58	0.58	0.64	0.62	0.61	0.52	0.49	0.47	0.45	0.45	0.48	0.69	0.63	0.67	0.68	0.66	0.66
5 disabled people	0.43	0.45	0.45	0.45	0.44	0.44	0.57	0.57	0.57	0.59	0.59	0.59	0.57	0.56	0.59	0.56	0.56	0.57	0.66	0.66	0.65	0.68	0.64	0.67
6 Muslims	0.45	0.52	0.48	0.46	0.46	0.49	0.60	0.62	0.62	0.63	0.63	0.61	0.54	0.48	0.51	0.47	0.46	0.50	0.64	0.66	0.66	0.67	0.64	0.66
7 immigrants	0.43	0.48	0.45	0.44	0.44	0.46	0.58	0.59	0.57	0.58	0.59	0.59	0.59	0.54	0.57	0.55	0.56	0.56	0.63	0.64	0.65	0.66	0.62	0.66

Table 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.