Proceedings of the Eighteenth International AAAI Conference on Web and Social Media (ICWSM 2024)

Improving Quantification with Minimal In-Domain Annotations: Beyond Classify and Count

Pius von Däniken[1], **Jan Milan Deriu**[1], **Alvaro Rodrigo**[2], **Mark Cieliebak**[1]
[1]Zurich University of Applied Sciences (ZHAW), Winterthur, Switzerland
[2] NLP & IR Group at UNED, Madrid, Spain
vode@zhaw.ch, deri@zhaw.ch, alvarory@lsi.uned.es, ciel@zhaw.ch

Abstract

Quantification is the task of estimating the class distribution in a given collection. With the growing availability of classification models, the use of classifiers for quantification has become increasingly popular, carrying the promise of eliminating the need for manual annotation. However, the naive classify and count approach presents clear limitations, especially evident in the face of domain discrepancies. In this work, we introduce two novel quantification methods, called CPCC and BCC, which can adapt to new target datasets with a small number of annotated in-domain samples ($N = 100$). To explore their real-world applicability, we apply our methods to a range of quantification tasks in the realm of hateful and offensive language, where they perform markedly better than classify and count and other existing methods.

1 Introduction

The advent of large pre-trained language models (Devlin et al. 2019; Liu et al. 2019; Brown et al. 2020, i.a.) based on the transformer architecture (Vaswani et al. 2017) has significantly improved many Natural Language Processing (NLP) tasks. In particular, they represent the de-facto standard for modern text classification systems. Combined with open science efforts encouraging authors to share their models, and platforms such as *Hugging Face* (Wolf et al. 2019) simplifying distribution, it has become easier than ever to gain access to high-quality off-the-shelf models for many classification tasks.

As a consequence, the end-user of a specific model is not necessarily the same as its developer. This also means that the data a classifier is applied to often does not necessarily match its training data. This is known as domain transfer, which is still a challenging problem (Deriu et al. 2017; Talat, Thorne, and Bingel 2018; Toraman, Şahinuç, and Yilmaz 2022).

In this work, we will explore the common setting where an analyst wants to determine the number of offensive posts in a given data collection. To avoid the costly process of manual annotation, they download an off-the-shelf classifier and run it on their data to count the number of offensive posts. This approach seems reasonable but gives rise to several particular challenges. The first is that the classifier was probably trained on a different domain, and the performance achieved on the training domain does not transfer to the target domain. Secondly, the annotation scheme used in the training data might differ significantly from the intended use of the analyst (e.g., offensive language is not uniformly defined (Vidgen and Derczynski 2020)). Thirdly, the prevalence of offensive language often differs in the training regime and the real world, leading to overestimating the true prevalence of offensive language. Thus, counting the number of offensive posts (i.e., the prevalence) according to a classifier is not straightforward.

Estimating the prevalence or empirical distribution of a certain class in a given set of samples is known as *quantification* (Forman 2008). This is a collection-level task, whereas, in classification, the task is to make accurate sample-level predictions. Estimating the prevalence of hateful and offensive language in social media is of keen interest to social science researchers to study societal trends (Solovev and Pröllochs 2022; Siegel et al. 2018).

The approach taken by our imaginary analyst is known as *classify and count* (Forman 2008). It can be shown that the resulting prevalence estimate depends on the true positive rate (TPR) and false positive rate (FPR) of the binary classifier at hand. Forman (2008) shows that there is the following relationship between the true prevalence p and the estimated prevalence $\hat{p}$ depending on the TPR and the FPR: $\hat{p} = p(TPR - FPR) + FPR$. Hence, the estimated prevalence can be substantially biased. Intuitively, we should be able to solve the above formula for p to get an improved estimate. However, in our scenario, the true TPR and FPR on the target dataset are unknown and often do not match the ones achieved on validation data during development. This point is of particular importance in the case of hateful and offensive language, where the exact inclusion criteria can be fuzzy and differ significantly depending on the data curator (Vidgen and Derczynski 2020), which can lead to underwhelming performance when applying a classifier to a different dataset.

In this work, we tackle the issue of applying a black-box classifier to estimate the prevalence of offensive language. In contrast to work in traditional transfer learning, we assume that we do not have access to the underlying classifier, which does not allow for fine-tuning on annotated data. However,

our proposed solution assumes that the analyst collects annotations for a small subset (e.g., $N = 100$) of their target data. In this case, we will show that we can get high-quality prevalence estimates for a range of classifiers and training datasets. More concretely, our main contributions are:

- Two novel quantification approaches that specifically tackle the issue of transferring the quantification capabilities of a classifier from one domain to another with a small set of only $N = 100$ samples.
- A novel sample selection method to collect the small set of samples used for domain adaptation.
- A large-scale set of experiments[1] showcasing the difficulty of domain transfer in the quantification setting, and that the performance of our novel approaches is superior to existing ones.

2 Related Work

2.1 Quantification

The problem of using classifiers for quantification has been studied in depth by Forman (2008, 2005). Indeed, many of the methods we will describe in Section 3 were first described there. González et al. (2017) provide a survey of quantification methods. In particular, they distinguish three high-level approaches: *classify, count, and correct*; *adapting classification training algorithms*; and *distribution matching*. The methods explored in this work all follow the *classify, count, and correct* approach. This is mainly due to our imagined setting, where we assume access to a black-box classifier. In the *correct* step of this class of approaches, we will need access to estimates of the TPR and FPR of the classifier. Forman (2008) proposes to estimate these during training using cross-validation. We will instead assume that there is a small set of in-domain annotated data that we can use for this.

Training dedicated quantification models, for example by adapting classification training algorithms for quantification, represents a promising avenue of research. Esuli and Sebastiani (2015) propose to adapt a structured SVM learning paradigm (Joachims 2005) in order to directly minimize the quantification error. This class of approaches does not fit our setting, as we will assume that we do not have any control over the training process, but only have access to a final model trained as a classifier.

The problem of quantification in textual data has been the topic of a few shared tasks. The *SemEval* series of shared tasks included quantification as a sub-task in their sentiment analysis challenges (Nakov et al. 2016; Rosenthal, Farra, and Nakov 2017). The *LeQua* shared task (Esuli et al. 2022) involved creating quantification systems for sentiment and the topic of product reviews.

2.2 Hateful and Offensive Language Detection

The quick growth of social networks has increased the interest in the automatic detection of offensive or harmful language (Hajibabaee et al. 2022). Although there are different definitions, the task mainly focuses on detecting content targeting individuals or entities using aggressive language (Vidgen and Derczynski 2020). Current technologies rely on data-driven approaches, based on training models on annotated data. However, the annotation criteria usually depend on the target receiving the offense comment (Vidgen et al. 2019), which might produce some bias. As a consequence, knowledge transference among collections is difficult, and therefore, the application to new data. Additionally, the class distribution highly depends on the method followed for data acquisition and might not reflect a real scenario.

Current methods mostly use transformer-based models (Liu et al. 2022). Some innovations apply an ensemble of models (Roy, Bhawal, and Subalalitha 2022), or include additional information, such as the detection of improper language (Plaza-del Arco et al. 2022) or multiword expressions (Zampieri, Illina, and Fohr 2023), for improving results. Although there have been several studies about transfer learning for hate speech detection (Markov and Daelemans 2021; Ali et al. 2022), the proposed systems usually only perform well on a test set with a distribution similar to the one seen during training (Toraman, Şahinuç, and Yilmaz 2022).

A high interest in hate speech detection can also be seen in the constant proposal of shared tasks. For example, *HatEval* (Basile et al. 2019), *OffensEval* (Zampieri et al. 2019b) at *SemEval 2019*, and a second edition of *OffensEval* at *SemEval 2020* (Zampieri et al. 2020) attracted a lot of participants and the datasets are still widely used. In fact, we include these datasets in our experiments (see Section 5.1). But there have been also several efforts in other languages such as Spanish with the *MEX-A3T* (Aragón et al. 2019, 2020) and *MeOffendEs* (Plaza-del Arco et al. 2021) tasks at *IberLEF*, the *OSACT4* shared task on Arabic (Mubarak et al. 2020) and in German at *GermEval 2018* (Wiegand, Siegel, and Ruppenhofer 2019).

2.3 Calibration

A desirable property of probabilistic classifiers is that they be *calibrated*. This means that their predicted score represents a true probability, meaning, for example, that in 80% of cases where the model gives a score of 0.8, the true label should be positive.

Guo et al. (2017) find that many neural network based classifiers at the time were poorly calibrated and discuss a variety of approaches to re-calibrate them. We will rely on *Platt Scaling* (Platt et al. 1999) in Section 4. Minderer et al. (2021) report that modern vision models are well calibrated. Desai and Durrett (2020) study the calibration of pre-trained transformer models, such as *RoBERTa* (Liu et al. 2019). They find that transformer models are generally well calibrated in-domain. Kong et al. (2020) tackle the challenge of getting calibrated out-of-domain scores from pre-trained transformer models using synthetic data.

3 Existing Approaches on Quantification Methods

In the following, we will assume that we are presented with an unlabelled set of text samples Q and tasked to find the

[1]The code to run our experiments is available at https://github.com/vodezhaw/icwsm2024/

fraction p of samples that could be considered hateful. We further assume that we have access to a real-valued classification function f and a decision threshold τ such that $f(x_i) \geq \tau$ means that the classifier considers the sample x_i to be hateful.

In this work, we consider the case where we have no control over how f was created. In this setting, we propose to partition Q into the sets Q_{calib} and Q_{eval}. We then collect labels for the samples in Q_{calib} which will be used by the different quantification algorithms described below to compute the prevalence estimate p on the remaining samples Q_{eval}.

We will notate $N_{calib} = |Q_{calib}|$ and $N_{eval} = |Q_{eval}|$ for the sizes of the two sets of samples. Furthermore, $s_i = f(x_i)$ is the real-valued output of f for a given sample x_i and $\hat{y}_i = \mathbb{I}[s_i \geq \tau]$ is the predicted label, where $\mathbb{I}$ denotes the indicator function (we regard $\hat{y}_i = 1$ as the classifier stating that the sample contains hate-speech).

We now introduce quantification approaches from existing literature that we adapted to our setting.

Classify and Count (CC). The simplest quantification method is called *classify and count (CC)* (Forman 2008) and consists in averaging the predictions of the given classifier: $p_{CC} = \frac{1}{N_{eval}} \sum_{i=1}^{N_{eval}} \hat{y}_i$.

Adjusted Classify and Count (ACC). As explained in Section 1, the naive *CC* estimate, p_{CC}, is biased by both the true positive rate (TPR) and false positive rate (FPR) of our classifier f at the given threshold τ. Since we know that $p_{CC} = p(TPR - FPR) + FPR$, we can solve for p. Since, in general, the true values of TPR and FPR are unknown, we compute their point estimates on Q_{calib}: $\hat{TPR} = \frac{\sum_{j=1}^{N_{calib}} \mathbb{I}[\hat{y}_j=1 \wedge y_j=1]}{\sum_{j=1}^{N_{calib}} \mathbb{I}[y_j=1]}$ and $\hat{FPR} = \frac{\sum_{j=1}^{N_{calib}} \mathbb{I}[\hat{y}_j=1 \wedge y_j=0]}{\sum_{j=1}^{N_{calib}} \mathbb{I}[y_j=0]}$. We then define the *adjusted classify and count (ACC)* (Forman 2008) estimate as: $p_{ACC} = \frac{p_{CC} - \hat{FPR}}{\hat{TPR} - \hat{FPR}}$.

Probabilistic Classify and Count (PCC). For probabilistic classifiers, the score s_i corresponds to the predicted conditional probability of the positive label: $s_i = p(y_i = 1|x_i)$. We can, therefore, average the classifier scores to get the expected fraction of hateful samples: $p_{PCC} = \frac{1}{N_{eval}} \sum_{i=1}^{N_{eval}} s_i$. This method is called *probabilistic classify and count (PCC)* (Forman 2008; Bella et al. 2010) and can only be applied to probabilistic classifiers.

Probabilistic Adjusted Classify and Count (PACC). Analogous to *CC*, we can define an adjusted version of *PCC* called *probabilistic adjusted classify and count (PACC)* (Bella et al. 2010). In this case, we rely on probabilistic estimates of TPR and FPR: $\hat{TPR} = \frac{\sum_{j=1}^{N_{calib}} s_i \mathbb{I}[y_j=1]}{\sum_{j=1}^{N_{calib}} \mathbb{I}[y_j=1]}$ and $\hat{FPR} = \frac{\sum_{j=1}^{N_{calib}} s_i \mathbb{I}[y_j=0]}{\sum_{j=1}^{N_{calib}} \mathbb{I}[y_j=0]}$. The resulting estimate of p is: $p_{PACC} = \frac{p_{PCC} - \hat{FPR}}{\hat{TPR} - \hat{FPR}}$.

4 Our Novel Quantification Methods

Next, we introduce our two novel approaches. The two approaches differ in that the first uses calibration to adjust the classifier outputs, and the second extends the *adjusted count* approach by including the uncertainty over TPR and FPR, instead of relying on point estimates.

Method	Uses τ	Uses Q_{calib}	Assumes probabilistic ratings
CC	x		
ACC	x	x	
PCC			x
PACC		x	x
CPCC		x	
BCC	x	x	

Table 1: Overview of the quantification methods described in Sections 3 and 4.

Calibrated Probabilistic Classify and Count (CPCC). One of the main points of focus of this work is quantification using a classification function f in cases where the set Q deviates substantially from the training data used to create f. In this setting, a probabilistic classifier f is often improperly calibrated for samples in Q (Guo et al. 2017; Kong et al. 2020). Therefore, we use *Platt-Scaling* (Platt et al. 1999) to re-calibrate f for Q[2]. This means training a logistic regression model on $(f(x_j), y_j)_{j=1}^{N_{calib}}$ and using the probabilistic predictions of this logistic regression model for samples in Q_{eval} as the new scores $\tilde{s}_i$. The resulting prevalence estimate is then: $p_{CPCC} = \frac{1}{N_{eval}} \sum_{i=1}^{N_{eval}} \tilde{s}_i$.

Bayesian Classify and Count (BCC). Recently, von Däniken et al. (2022) introduced a Bayesian model which can be re-purposed for quantification. It was originally developed in the context of automated evaluation of text generation systems. The main idea is to estimate the prior probability distributions of p, TPR, and FPR from Q_{calib}, which can be combined with the predicted labels $\hat{y}_i$ on Q_{eval}. The prevalence estimate p_{BCC} is then the expected value of the resulting posterior distribution.

We summarize the quantification methods described in this section and in Section 3 in Table 1. *CPCC* and *BCC* were specifically developed to take advantage of our assumed setting where we have access to a labelled in-domain set of samples Q_{calib}. By design, *CC* and *PCC* cannot incorporate the additional information. In contrast, the *adjusted count* methods can easily be adapted to our setting.

5 Evaluation Framework

In this section, we describe the framework for evaluating the proposed quantification methods. We will rely on a range of existing datasets to create various classifiers f[3]. We will also use the test sets of these collections as the quantification

[2]We compare to *Isotonic Regression* (Zadrozny and Elkan 2002) and *Histogram Binning* (Zadrozny and Elkan 2001) in Appendix A.

[3]While we train a range of classifiers here, for our evaluation we treat them as if we took them off-the-shelf.

targets Q. We give the details of the datasets, classifiers, and evaluation measures selected for our study below.

5.1 Datasets

We will now give a brief overview of the different datasets used in this study. A summary of their properties is given in Table 2. We relied on hatespeechdata.com maintained by Vidgen and Derczynski (2020) for curating this collection. Our main inclusion criterion was that a dataset should be relatively easily available for download and that its annotations can be converted into a binary label. We aimed to maintain the canonical splits into train C_{train}, development C_{dev}, and test sets C_{test} wherever possible. Datasets that do not have a canonical separation between C_{train} and C_{test} will be used as C_{test} only. If a dataset does not have a development set C_{dev}, we use 10% of the training data as C_{dev}. Our experimental setting comprises a highly diverse set of 11 datasets with different domains, annotation schemes, and prevalence of offensive language.

Davidson. Davidson et al. (2017) collected tweets from users who had used words and phrases from a hate speech lexicon. The tweets were annotated by crowd-workers as *hate*, *offensive*, or *neither*. The main distinction between *hate* and merely *offensive* tweets is that hateful tweets express explicit hatred towards a target group. For our experiments, we consider the case where *hate* and *offensive* together constitute the positive class, which we call *Davidson Hate & Offensive*. We also consider the case where only the *hate* class constitutes the positive class and call this *Davidson Hate Only*.

Dynamically Generated Hate (DGH). Vidgen et al. (2021) set out to create a dataset to train robust classifiers. Starting from a classifier trained on multiple existing hate speech datasets, they proceed in multiple rounds. In each round, they ask annotators to produce texts that trigger misclassifications from the model. A new model is then re-trained based on this data. They annotated both whether a text is hateful as well as the type and target of hate. In this work, we rely on the top-level binary hate label.

ETHOS. The *ETHOS* (Mollas et al. 2022) dataset comes in both a binary and a multi-label version. It consists of comments from *YouTube* and *Reddit*. The main goal was to create a balanced multi-label dataset covering aspects such as race, gender, and religion, as well as whether a comment is directed or generalized, or incites violence. We consider positive cases those where strictly more than half of the annotators consider a comment as hateful. Due to its limited size, we use it only as a test set.

Ex Machina. Wulczyn, Thain, and Dixon (2017) study personal attacks in discussions on *Wikipedia*. They had crowd workers annotate random comments, as well as comments sampled specifically from users that were banned for personal attacks. Each comment was annotated by 10 workers. In our experiments, we consider positive cases those where more than half of the annotators consider a comment a personal attack.

HASOC 2019. The *HASOC 2019* (Mandl et al. 2019) shared task contained three sub-tasks. The first sub-task consisted of distinguishing hateful and offensive content from non-offensive posts. The other sub-tasks involved classifying the type of hate or offense and its target. The task organizers provided datasets in English, German, and Hindi. In this work, we rely on the binary labels and the English dataset.

HatEval 2019. The *HatEval 2019* (Basile et al. 2019) shared task focused on hate that targets women and immigrants. The main sub-task involved classifying Twitter posts into hateful and not hateful. The other sub-tasks were predicting the specificity of the target and aggressiveness of the tweet. The organizers provided data in English and Spanish. We focus on the binary annotations and English data.

Jigsaw. The *Jigsaw Toxic Comment Classification Challenge* (cjadams et al. 2017) is a popular dataset on *Kaggle*[4], an online platform that hosts machine learning challenges. The dataset contains comments from Wikipedia discussions annotated with multiple binary labels: toxic, severely toxic, obscene, threat, insult, and identity hate. In this work, we assign a positive label if any of these labels are present.

OLID. The *Offensive Language Identification Dataset (OLID)* (Zampieri et al. 2019a) contains layered annotations whether a tweet contains offensive language, whether the offense is targeted, and whether the target is an individual or a group. The dataset has been used to run the *OffensEval 2019* (Zampieri et al. 2019b) shared task. In this work, we rely on the top-level binary offensiveness labels.

SOLID. The *Semi-Supervised Offensive Language Identification Dataset (SOLID)* (Rosenthal et al. 2021) is a semi-supervised extension of *OLID*. The annotation scheme follows *OLID*, and they provide a large corpus of English tweets with automatic annotations. In addition, they include a larger manually annotated test set. The *SOLID* data was used for the *OffensEval 2020* (Zampieri et al. 2020) shared task. In this work, we will only use the manually annotated test set.

SWAD. The *Swear Word Abusiveness Dataset (SWAD)* (Pamungkas, Basile, and Patti 2020) is also based on *OLID*. The authors provide new annotations indicating whether the use of a specific swear word makes a post abusive or not. We use these binary labels as an additional test set.

WASSA. Grimminger and Klinger (2021) created a dataset for both stance detection and hatefulness and offensiveness. It consists of tweets about the candidates of the 2020 US elections. We focus on the binary hate / offensiveness label.

5.2 Classifiers

Next, we describe the classifiers we use in our experiments. We train three types of classifiers on each training set individually and collect their predictions for all test sets. The aim is to use a diverse set of classifiers in terms of architectures, models, and performance to showcase the generality of our quantification approach.

Electra. *Electra* (Clark et al. 2020) models have the same architecture as *BERT* (Devlin et al. 2019), but follow a different pre-training procedure. During masked language modeling (MLM) pre-training, there are two models: a generator G and a discriminator D. G is trained to predict the

[4]https://kaggle.com

Dataset	Size Train	Size Dev	Size Test	Prev. Train	Prev. Dev	Prev. Test
Davidson Hate & Offensive	22304	-	2479	83.2%	-	83.2%
Davidson Hate Only	22304	-	2479	5.8%	-	5.8%
DGH	32924	4100	4120	53.9%	52.9%	55.0%
ETHOS	-	-	998	-	-	43.4%
Ex Machina	69526	23160	23178	13.4%	13.8%	13.2%
HASOC 2019	5852	-	1153	38.6%	-	25.0%
HatEval 2019	9000	1000	3000	42.0%	42.7%	42.0%
Jigsaw	159571	-	63978	10.2%	-	9.8%
OLID	13240	-	860	33.2%	-	27.9%
SOLID	-	-	5993	-	-	50.1%
SWAD	-	-	2578	-	-	32.7%
WASSA	2400	-	600	12.2%	-	9.8%

Table 2: Overview of the datasets used for our experiments. We report the prevalence of the positive class for every dataset as described in Section 5.1.

masked tokens. The masked tokens are then replaced by the top prediction of G, and D is trained to predict which tokens were generated by G and which ones are original.

For our experiments, we use a discriminator model checkpoint [5] from *Hugging Face*. We train the model for up to 50 epochs with batch size 16, stopping early if the loss on C_{dev} does not improve for 5 epochs. The model weights are updated using *AdamW* (Loshchilov and Hutter 2019) with a learning rate of $5\text{e}{-5}$ and weight decay of 0.01. Texts that are longer than the maximum of 512 tokens are truncated to that length.

Twitter-RoBERTa. Since many datasets consist of tweets, we include classifiers pre-trained on Twitter data. Specifically, we fine-tune a *RoBERTa* (Liu et al. 2019) model that was pre-trained by Barbieri et al. (2020) on 58M tweets [6]. Our training procedure is identical to the *Electra* models.

TF-IDF SVM. We train *Support Vector Machine (SVM)* (Cortes and Vapnik 1995) models on *TF-IDF* (Manning, Raghavan, and Schütze 2008) features. We use the *TfidfVectorizer* and *LinearSVC* implementations provided by *scikit-learn* (Pedregosa et al. 2011). During training, we perform a grid search over three hyper-parameters and select the model that has the highest F1 score on C_{dev}. The hyperparameter ranges are: maximum n-gram length (2, 3, 5, or 7), whether to use binary term frequencies or log-scale them, and the regularization strength of the SVM loss (0.001, 0.01, 0.1, 1, 10, 100, or 1000).

Perspective API. Further, we collect predictions from an online service [7]. Since the organisations developing this service are also the providers of the *Jigsaw* dataset, its predictions are in line with the *Jigsaw* multi-label annotations. It will predict a score between 0 and 1 for each of the *Jigsaw* labels. We will use the maximum score over all labels and a decision threshold of $\frac{1}{2}$.

[5] https://huggingface.co/google/electra-base-discriminator

[6] https://huggingface.co/cardiffnlp/twitter-roberta-base

[7] https://www.perspectiveapi.com/

N_{calib}	10	20	30	40	50	60	70+
Random	0.13	0.08	0.03	0.02	0.01	0.01	0.00
Quantile	0.10	0.02	0.00	0.00	0.00	0.00	0.00

Table 3: Fraction of cases where our selection strategies do not produce any positive samples depending on the number N_{calib} of samples selected.

5.3 Selecting Samples to Annotate

As described in Section 3, we select a certain number of samples of the unlabelled quantification set Q to be labelled and then compute the prevalence estimate on the remaining samples. We will explore two methods of selecting samples for annotation to partition Q into Q_{calib} and Q_{eval}.

The first sample selection method we will explore is to select N_{calib} random samples to annotate form Q_{calib}. We will call this method *random*. The other method is based on the classifier scores $s_i = f(x_i)$. We first sort the samples by their score s_i. We then split the sorted samples into $N_q = 10$ consecutive equal length segments. We then select $\frac{N_{calib}}{N_q}$ random samples from each segment to form Q_{calib}. We will call this method *quantile*, since each of the N_q segments correspond to a quantile of the score distribution.

Ideally, our selection method will produce a Q_{calib} that is representative of the whole Q, which is a pre-requisite for the methods in Sections 3 and 4 to work. A particular failure mode of our selection strategies is when they fail to select any positive samples at all. In this case, we cannot properly estimate the TPR and FPR of f, and calibration using *Platt Scaling* is also impossible.

In Table 3, we show the fraction of cases in which neither selection method produces any positive samples for various values of N_{calib}. For this, we applied each sample selection strategy to each of our 12 test sets 10 times with different random seeds. In general, we observe that the *quantile* strategy produces fewer failures and both strategies succeed when $N_{calib} \geq 70$. Based on this, we will use $N_{calib} = 100$ for our experiments, unless otherwise stated.

One other factor that influences the success of our selec-

p	0.01	0.02	0.03	0.05	0.07	0.10
Random	0.26	0.11	0.02	0.03	0.00	0.00
Quantile	0.36	0.16	0.09	0.02	0.00	0.00

Table 4: Fraction of cases where our selection strategies do not produce any positive samples depending on the prevalence p in the target set Q.

tion methods is the true prevalence p. In general, the smaller p is, the harder it is to select positives. In Table 2 we can see that the smallest p for any of our test sets is 9.8%. To see how often our selection methods fail for a small p, we sub-sample the positive class of our test sets to a number of fixed prevalences. In cases where sub-sampling would lead to fewer than 15 positives, we remove the test set from consideration for that given prevalence. Table 4 shows the fraction of cases where either selection method does not produce any positive samples for different values of p. In this case, we fixed $N_{calib} = 100$. We can see that as long as the true prevalence $p > 3\%$, both selection methods produce only few or no failures.

In what follows, we will generally operate in the regime where there are very few or no failures during the sample selection process and any sample selection failures that do occur will be excluded from the analysis.

5.4 Performance Measures

To measure the performance of a given quantification method given an unlabelled set Q and classifier f (with threshold τ), we first apply a sample selection method from Section 5.3, resulting in $Q_{eval} \cup Q_{calib} = Q$. Let p_x be the estimated prevalence produced by a method described in Section 3 or 4 and p be the true prevalence in Q_{eval}.

Many different measures for the quality of a quantification approach exist (González et al. 2017). In this work, we will use the *symmetric absolute percentage error (SAPE)* which is defined as $\frac{|p_x - p|}{p_x + p}$. The *SAPE* is bounded and ranges from 0 to 1 with lower values indicating better quantification performance. In particular, it captures the intuition that the severity of some constant absolute error depends on the true underlying prevalence, i.e., an estimate that is 0.01 off is less severe when the true prevalence is 0.50 than if the true prevalence were 0.02.

6 Experiments

In this section, we show the results of our experiments. We fine-tune the three trainable models on each of the datasets where a train set C_{train} is available (see Table 2) and generate the outputs for each of the test sets. Overall, we note that the sample-based performances are highly diverse and the ROC-AUC scores range from 0.486 to 0.989 depending on the classifier, train and test set combination (see Appendix B). Overall, we train 3 classifiers on 9 training sets and add 1 online service for a total of 28 classifiers f. For each model, we then apply each of the quantification methods and the test splits serve as the unlabelled data Q. In the following, we will show the main results comparing quantification approaches and sample selection strategies and explore other factors.

	Random			Quantile		
	μ	Med.	95	μ	Med.	95
CC	0.368	0.264	1.000	0.368	0.264	1.000
ACC	0.352	0.230	1.000	0.240	0.125	1.000
PCC	0.299	0.220	0.827	0.299	0.219	0.827
PACC	0.306	0.167	1.000	0.151	0.077	0.635
CPCC	0.073	0.049	0.211	0.085	0.059	0.275
BCC	**0.067**	**0.046**	**0.209**	0.082	0.059	0.259

Table 5: *SAPE* of different quantification methods. We report the mean (μ), median (Med.), and 95th percentile (95) of errors for all quantification methods and sample selection methods.

6.1 Comparing Quantification Methods

Here, we compare the performance of the quantification methods described in Section 4. We also include methods from Section 3 as baselines. We applied each quantification approach to each pair of classifier and test set, using a set of $N_{calib} = 100$ in-domain calibration samples. We ran each experiment 10 times with different random seeds, yielding 28 classifiers $\times$ 12 test sets $\times$ 10 results for each quantification approach (i.e., 3360 results). In Table 5, we show the mean, median, and 95th percentile *SAPE* for each quantification method and sample selection strategy.

The results show that our novel approaches *CPCC* and *BCC* markedly outperform the existing quantification approaches. They achieve a median *SAPE* of 0.049 and 0.046, respectively, for *random* sample selection, and both achieve a median *SAPE* of 0.059 for *quantile* selection. Out of the existing approaches, *PACC* in combination with *quantile* sampling performs best, achieving a median *SAPE* of 0.077. However, when using the *random* sampling approach, the error rate of *PACC* rises substantially to a *SAPE* of 0.167. The naive *classify and count* approach only achieves high median *SAPE* scores of 0.264. Furthermore, the variance is much lower for our novel approaches. In fact, *CPCC* and *BCC* have a 95th percentile *SAPE* of 0.211 and 0.209, respectively, for *random* sample selection. In contrast, the 95th percentile *SAPE* of the other approaches ranges from 0.635 to 1.0.

Thus, our novel quantification approaches achieve superior performance scores and are also more robust in regard to domains and underlying classifiers.

6.2 Out-of-Domain Calibration

In the above experiment, we used $N_{calib} = 100$ in-domain calibration samples. Here, we showcase the necessity for these by repeating the same type of experiment but using an entire out-of-domain dataset for calibration. This corresponds to the setting where we rely on a different pre-existing set of labelled samples (for example from a shared task) instead of selecting a number of samples from Q to be labelled. We simulate this scenario by pairing all test splits in Table 2 with each other to build (Q_{calib}, Q_{eval}) pairs (such

	Out-of-Domain		
	μ	Med.	95
CC	0.368	0.262	1.000
ACC	0.582	0.520	1.000
PCC	**0.299**	**0.221**	0.829
PACC	0.575	0.520	1.000
CPCC	0.357	0.309	0.815
BCC	0.394	0.365	**0.792**

Table 6: *SAPE* of different quantification methods when using *out-of-domain* data as Q_{calib}. We report the mean (μ), median (Med.), and 95th percentile (95) of errors for all quantification methods and sample selection methods.

	Random			Quantile		
	μ	Med.	95	μ	Med.	95
PACC-10	0.424	0.294	1.000	0.249	0.163	1.000
PACC-20	0.411	0.289	1.000	0.230	0.142	1.000
PACC-40	0.366	0.246	1.000	0.207	0.117	0.892
PACC-80	0.311	0.173	1.000	0.158	0.081	0.654
PACC-160	0.272	0.136	1.000	0.131	0.065	0.519
CPCC-10	0.175	0.126	0.505	0.180	0.132	0.511
CPCC-20	0.140	0.102	0.385	0.151	0.111	0.432
CPCC-40	0.120	0.079	0.399	0.132	0.088	0.400
CPCC-80	0.080	0.056	0.241	0.094	0.063	0.279
CPCC-160	0.051	0.034	0.158	0.073	0.049	0.224
BCC-10	0.157	0.120	0.441	0.174	0.132	0.486
BCC-20	0.131	0.101	0.358	0.141	0.108	0.396
BCC-40	0.103	0.075	0.342	0.122	0.089	0.368
BCC-80	0.072	0.053	0.216	0.089	0.062	0.264
BCC-160	0.048	0.032	0.160	0.071	0.049	0.219

Table 7: *SAPE* of different quantification methods depending on the number of samples selected for calibration. Sample sizes are indicated as a suffix to the name of the quantification method. We report the mean (μ), median (Med.), and 95th percentile (95) of errors for all quantification methods and sample selection methods.

that $Q_{calib} \neq Q_{test}$, e.g., calibrate on SWAD and test on OLID). We use all 28 classifiers and repeat each experiment 10 times with different random seeds.

The results show that each of the quantification approaches yield relatively high median errors. In fact, none of the approaches that rely on Q_{calib} outperform the ones that do not (*CC* and *PCC*). This is to be expected, as an out-of-domain dataset can yield mismatched TPR and FPR estimates. Similarly, *CPCC* is influenced by the inconsistent prevalence between datasets. This explicitly showcases the need to use in-domain samples for calibration.

6.3 Reducing the Number of Samples to Select

Since using out-of-domain data is inadequate, we now investigate to which degree the number of samples N_{calib} to annotate can be reduced. For this, we repeat the experiments from Table 5 with different values of $N_{calib} \in \{10, 20, 40, 80.160\}$ and compute the resulting *SAPE* for the best-performing methods *PACC*, *CPCC*, and *BCC*. We have already mentioned that reducing N_{calib} can lead to an increase in cases where our sample selection method fails to find positive samples (see Section 5.3). In these cases, we dropped the experiment configuration from the analysis, meaning that those cases do not contribute to the error statistics.

Table 7 shows the results of the sample-reduction experiments. For *CPCC*, the median *SAPE* ranges from 0.034 for 160 samples to 0.126 for 10 samples. For *BCC* the median *SAPE* ranges from 0.032 for 160 samples to 0.120 for 10 samples. Similar to the main results, the *random* selection method performs better than the *quantile* selection. The range is larger for *PACC* (the median *SAPE* lies between 0.065 and 0.163), and similar to the main results, *quantile* outperforms the *random* selection method.

We note that both *CPCC* and *BCC* outperform *CC*, *ACC*, and *PCC* (see Table 5) using only 10 samples. For the random selection method, *CPCC-40* and *BCC-40* are on par with *PACC-80* (0.079, 0.075, and 0.081, respectively). Considering only the 95th percentile, *CPCC-10* and *BCC-10* outperform *PACC-160* (0.505, 0.441, and 0.519 respectively).

Overall, all methods perform better for larger N_{calib}. The performance of *CPCC* and *BCC* degrades relatively gracefully. The sample selection method has a large influence on *PACC*, whereas *CPCC* and *BCC* seem relatively stable with *random* selection performing slightly better.

6.4 Low Prevalence

To study how our methods perform when decreasing the true prevalence of hateful content, we proceed in the same way as in Section 5.3. For each target prevalence, we sub-sample the positive class to that level, making sure there are at least 15 positives. If there are fewer positives when sub-sampling, we ignore the configuration. Similarly, we ignore cases where our sample selection strategy (using $N_{calib} = 100$) does not produce any positives in D_{calib}. We run each experiment 10 times with different random seeds.

We show the *SAPE* for *CC*, *PACC*, *CPCC*, and *BCC* in Tables 8 and 9. For all methods the *SAPE* increases as the prevalence p decreases. This is to be expected, as the denominator in the *SAPE* calculation gets smaller. The median *SAPE* for *CPCC* with *random* sample selection ranges from 0.092 to 0.206. For *BCC* the median *SAPE* ranges from 0.090 to 0.178. Similar to previous results, *quantile* sample selection performs worse for *CPCC* and *BCC*. For *PACC*, *quantile* selection leads to lower median *SAPE*, ranging from 0.165 to 0.361. We note that in the low prevalence setting, the performance difference between *PACC* and our novel methods is even more pronounced. This is also apparent when considering the 95th percentile *SAPE* which is 1.0 for all prevalences for *PACC*.

7 Discussion & Conclusion

Our setting is motivated by a real-world scenario in which black-box classifiers are used without the possibility to fine-tune. Furthermore, our scenario also assumes that there is only a very small set of annotations available making classical domain-transfer extremely difficult.

	CC			PACC			CPCC			BCC		
p	μ	Med.	95	μ	Med.	95	μ	Med.	95	μ	Med.	95
0.020	0.716	0.780	1.000	0.706	0.811	1.000	0.215	0.206	0.527	**0.196**	**0.178**	**0.523**
0.030	0.648	0.724	1.000	0.629	0.677	1.000	0.189	0.157	0.529	**0.169**	**0.155**	**0.358**
0.050	0.578	0.630	1.000	0.542	0.494	1.000	0.179	**0.136**	0.476	**0.158**	0.137	**0.389**
0.070	0.499	0.524	1.000	0.490	0.398	1.000	0.128	0.107	0.298	**0.115**	**0.093**	**0.275**
0.100	0.464	0.481	1.000	0.411	0.302	1.000	0.109	0.092	0.274	**0.106**	**0.090**	**0.243**

Table 8: *SAPE* of different quantification methods with *random* sample selection depending on the prevalence p. We report the mean (μ), median (Med.), and 95th percentile (95) of errors for all quantification methods and sample selection methods.

	CC			PACC			CPCC			BCC		
p	μ	Med.	95	μ	Med.	95	μ	Med.	95	μ	Med.	95
0.020	0.706	0.772	1.000	0.458	0.361	1.000	0.219	0.210	**0.473**	**0.195**	**0.178**	**0.473**
0.030	0.645	0.715	1.000	0.440	0.333	1.000	0.237	0.205	0.551	**0.201**	**0.176**	**0.480**
0.050	0.575	0.624	1.000	0.344	0.240	1.000	0.201	**0.144**	0.635	**0.180**	0.151	**0.455**
0.070	0.501	0.525	1.000	0.299	0.187	1.000	0.170	0.129	0.486	**0.156**	**0.123**	**0.418**
0.100	0.465	0.483	1.000	0.266	0.165	1.000	0.153	0.117	0.440	**0.141**	**0.109**	**0.391**

Table 9: *SAPE* of different quantification methods with *quantile* sample selection depending on the prevalence p. We report the mean (μ), median (Med.), and 95th percentile (95) of errors for all quantification methods and sample selection methods.

In Sections 5.1 and 5.2, we described how we train various classification models for our experiments following the standard procedure of using train (C_{train}) and development data (C_{dev}). In the usual classification setting, one would then evaluate f on the test data C_{test}. In this work, we use C_{test} as the quantification target data Q. This leads to the following issue: our method relies on splitting Q into Q_{calib} and Q_{eval} and, crucially, we assume access to the labels of Q_{calib} in order to calibrate the predictions of f. In the traditional setting, this leads to an unfair advantage. Of course, in this work, this advantage is entirely intentional and we adjusted previous and proposed quantification methods accordingly (see Table 5). We could, instead, have opted to try using C_{dev} as our calibration set Q_{calib}. One issue is that in the real world there might not be any annotated data a-priori for a domain of interest. Further, we have shown in Section 6.2 that using out-of-domain data for calibration yields poor results. We can see in Table 2 that the prevalence of hateful content can differ even between different splits of the same dataset, e.g. in the cases of *HASOC 2019* and *OLID*.

Our experiments show that it is possible to get good quantification estimates with only 100 annotated in-domain samples. The results are robust over a variety of classifiers and datasets, in particular when considering that our datasets had annotations for related but notably different tasks.

It should, therefore, in principle be possible to study the prevalence of hateful content in a new dataset using existing off-the-shelf classifiers. The main motivation to use classifiers for this purpose in the first place is to eschew the laborious annotation process. Our method yields markedly improved results while keeping the annotation effort low.

We note that inaccurate quantification approaches can pose significant risks, as they can lead to wildly inaccurate prevalence estimates. For example, in Table 8 we see that the naive *CC* approach has a median *SAPE* of 0.481 for the moderately low prevalence $p = 0.1$. This means that the estimate under *CC* could be as low as 0.035 or up to 0.285. Conclusions based on such poor estimates can be highly misleading. This can lead to misdirected policies and interventions or a misallocation of resources.

An important future extension of this work is considering how we can properly compare the prevalence of hateful content between two collections. In this case, we would want to gain access to a statistical test that can tell us whether one collection contains significantly more such content than another. This means that we would have to go beyond point estimates of p and consider confidence intervals. Interestingly, *BCC* in principle produces a full posterior distribution that can be used for this (von Däniken et al. 2022).

While our methods could in principle be applied to other binary quantification tasks, this work is limited to hateful, abusive, and toxic language in English. We have relied entirely on existing datasets that are publicly available. Since the main focus of this work were quantification methods, no one had to assess the content of the datasets in detail. Therefore, no one had to come into contact with abusive material or potentially personally identifiable information.

In the regime of very low prevalences our method runs into the particular issue that selecting samples to build D_{calib} can fail. In this work, we focused on evaluating the quantification methods. Better sample selection strategies could be developed based on active learning (Kumar and Gupta 2020) and rare class discovery methods (He and Carbonell 2008).

Acknowledgments

This work was supported by the CHIST-ERA HAMiSoN project grant CHIST-ERA-21-OSNEM002, by SNF

20CH21 209672 and AEI PCI2022-135026-2.

References

Ali, R.; Farooq, U.; Arshad, U.; Shahzad, W.; and Beg, M. O. 2022. Hate speech detection on Twitter using transfer learning. *Computer Speech & Language*, 74: 101365.

Aragón, M. E.; Carmona, M. A. A.; Montes-y Gómez, M.; Escalante, H. J.; Pineda, L. V.; and Moctezuma, D. 2019. Overview of MEX-A3T at IberLEF 2019: Authorship and Aggressiveness Analysis in Mexican Spanish Tweets. In *IberLEF@ SEPLN*, 478–494.

Aragón, M. E.; Jarquín-Vásquez, H. J.; Montes-y Gómez, M.; Escalante, H. J.; Pineda, L. V.; Gómez-Adorno, H.; Posadas-Durán, J. P.; and Bel-Enguix, G. 2020. Overview of MEX-A3T at IberLEF 2020: Fake News and Aggressiveness Analysis in Mexican Spanish. In *IberLEF@ SEPLN*, 222–235.

Barbieri, F.; Camacho-Collados, J.; Espinosa Anke, L.; and Neves, L. 2020. TweetEval: Unified Benchmark and Comparative Evaluation for Tweet Classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, 1644–1650. Online: Association for Computational Linguistics.

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*, 54–63. Minneapolis, Minnesota, USA: Association for Computational Linguistics.

Bella, A.; Ferri, C.; Hernandez-Orallo, J.; and Ramirez-Quintana, M. J. 2010. Quantification via Probability Estimators. In *Proceedings of the 2010 IEEE International Conference on Data Mining*, ICDM '10, 737–742. USA: IEEE Computer Society. ISBN 9780769542560.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; 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.; 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. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 1877–1901. Curran Associates, Inc.

cjadams, J., Sorensen; Elliott, J.; Dixon, L.; McDonald, M.; nithum; and Cukierski, W. 2017. Toxic Comment Classification Challenge.

Clark, K.; Luong, M.; Le, Q. V.; and Manning, C. D. 2020. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Cortes, C.; and Vapnik, V. N. 1995. Support-Vector Networks. *Machine Learning*, 20: 273–297.

Davidson, T.; Warmsley, D.; Macy, M.; and Weber, I. 2017. Automated Hate Speech Detection and the Problem of Offensive Language. *Proceedings of the International AAAI Conference on Web and Social Media*, 11(1): 512–515.

Deriu, J. M.; Weilenmann, M.; Von Gruenigen, D.; and Cieliebak, M. 2017. Potential and Limitations of Cross-Domain Sentiment Classification. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, 17–24. Valencia, Spain: Association for Computational Linguistics.

Desai, S.; and Durrett, G. 2020. Calibration of Pre-trained Transformers. In Webber, B.; Cohn, T.; He, Y.; and Liu, Y., eds., *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 295–302. Online: Association for Computational Linguistics.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 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)*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

Esuli, A.; Moreo, A.; Sebastiani, F.; and Sperduti, G. 2022. A Concise Overview of LeQua@CLEF 2022: Learning to Quantify. In Barrón-Cedeño, A.; Da San Martino, G.; Degli Esposti, M.; Sebastiani, F.; Macdonald, C.; Pasi, G.; Hanbury, A.; Potthast, M.; Faggioli, G.; and Ferro, N., eds., *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, 362–381. Cham: Springer International Publishing. ISBN 978-3-031-13643-6.

Esuli, A.; and Sebastiani, F. 2015. Optimizing Text Quantifiers for Multivariate Loss Functions. *ACM Trans. Knowl. Discov. Data*, 9(4).

FORCE11. 2020. The FAIR Data principles. https://force11.org/info/the-fair-data-principles/.

Forman, G. 2005. Counting Positives Accurately Despite Inaccurate Classification. In Gama, J.; Camacho, R.; Brazdil, P. B.; Jorge, A. M.; and Torgo, L., eds., *Machine Learning: ECML 2005*, 564–575. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN 978-3-540-31692-3.

Forman, G. 2008. Quantifying counts and costs via classification. *Data Mining and Knowledge Discovery*.

Gebru, T.; Morgenstern, J.; Vecchione, B.; Vaughan, J. W.; Wallach, H.; Iii, H. D.; and Crawford, K. 2021. Datasheets for datasets. *Communications of the ACM*, 64(12): 86–92.

González, P.; Castaño, A.; Chawla, N. V.; and Coz, J. J. D. 2017. A Review on Quantification Learning. *ACM Comput. Surv.*, 50(5).

Grimminger, L.; and Klinger, R. 2021. Hate Towards the Political Opponent: A Twitter Corpus Study of the 2020 US Elections on the Basis of Offensive Speech and Stance Detection. In *Proceedings of the Eleventh Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, 171–180. Online: Association for Computational Linguistics.

Guo, C.; Pleiss, G.; Sun, Y.; and Weinberger, K. Q. 2017. On Calibration of Modern Neural Networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, 1321–1330. JMLR.org.

Hajibabaee, P.; Malekzadeh, M.; Ahmadi, M.; Heidari, M.; Esmaeilzadeh, A.; Abdolazimi, R.; and Jones, J. H. J. 2022. Offensive Language Detection on Social Media Based on Text Classification. In *2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*, 0092–0098.

He, J.; and Carbonell, J. 2008. Rare class discovery based on active learning. 7P. 10th International Symposium on Artificial Intelligence and Mathematics, ISAIM 2008 ; Conference date: 02-01-2008 Through 04-01-2008.

Joachims, T. 2005. A Support Vector Method for Multivariate Performance Measures. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, 377–384. New York, NY, USA: Association for Computing Machinery. ISBN 1595931805.

Kong, L.; Jiang, H.; Zhuang, Y.; Lyu, J.; Zhao, T.; and Zhang, C. 2020. Calibrated Language Model Fine-Tuning for In- and Out-of-Distribution Data. In Webber, B.; Cohn, T.; He, Y.; and Liu, Y., eds., *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1326–1340. Online: Association for Computational Linguistics.

Kumar, P.; and Gupta, A. 2020. Active Learning Query Strategies for Classification, Regression, and Clustering: A Survey. *Journal of Computer Science and Technology*, 35(4): 913–945.

Liu, J.; Kong, D.; Huang, L.; Mao, D.; and Xue, H. 2022. Multiple Instance Learning for Offensive Language Detection. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, 7387–7396. Abu Dhabi, United Arab Emirates: 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:1907.11692.

Loshchilov, I.; and Hutter, F. 2019. Decoupled Weight Decay Regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Mandl, T.; Modha, S.; Majumder, P.; Patel, D.; Dave, M.; Mandlia, C.; and Patel, A. 2019. Overview of the HASOC Track at FIRE 2019: Hate Speech and Offensive Content Identification in Indo-European Languages. In *Proceedings of the 11th Annual Meeting of the Forum for Information Retrieval Evaluation*, FIRE '19, 14–17. New York, NY, USA: Association for Computing Machinery. ISBN 9781450377508.

Manning, C. D.; Raghavan, P.; and Schütze, H. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

Markov, I.; and Daelemans, W. 2021. Improving Cross-Domain Hate Speech Detection by Reducing the False Positive Rate. In *Proceedings of the Fourth Workshop on NLP for Internet Freedom: Censorship, Disinformation, and Propaganda*, 17–22. Online: Association for Computational Linguistics.

Minderer, M.; Djolonga, J.; Romijnders, R.; Hubis, F.; Zhai, X.; Houlsby, N.; Tran, D.; and Lucic, M. 2021. Revisiting the Calibration of Modern Neural Networks. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*, volume 34, 15682–15694. Curran Associates, Inc.

Mollas, I.; Chrysopoulou, Z.; Karlos, S.; and Tsoumakas, G. 2022. ETHOS: a multi-label hate speech detection dataset. *Complex & Intelligent Systems*, 8(6): 4663–4678.

Mubarak, H.; Darwish, K.; Magdy, W.; Elsayed, T.; and Al-Khalifa, H. 2020. Overview of OSACT4 Arabic Offensive Language Detection Shared Task. In *4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, 48–52. Marseille, France: European Language Resource Association. ISBN 979-10-95546-51-1.

Nakov, P.; Ritter, A.; Rosenthal, S.; Sebastiani, F.; and Stoyanov, V. 2016. SemEval-2016 Task 4: Sentiment Analysis in Twitter. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, 1–18. San Diego, California: Association for Computational Linguistics.

Pamungkas, E. W.; Basile, V.; and Patti, V. 2020. Do You Really Want to Hurt Me? Predicting Abusive Swearing in Social Media. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, 6237–6246. Marseille, France: European Language Resources Association. ISBN 979-10-95546-34-4.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Duchesnay, E. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830.

Platt, J.; et al. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3): 61–74.

Plaza-del Arco, F. M.; Casavantes, M.; Escalante, H. J.; Martín-Valdivia, M. T.; Montejo-Ráez, A.; Montes, M.; Jarquín-Vásquez, H.; Villaseñor-Pineda, L.; et al. 2021. Overview of MeOffendEs at IberLEF 2021: Offensive language detection in Spanish variants. *Procesamiento del Lenguaje Natural*, 67: 183–194.

Plaza-del Arco, F. M.; Molina-González, M. D.; Ureña-López, L. A.; and Martín-Valdivia, M.-T. 2022. Integrating implicit and explicit linguistic phenomena via multi-task learning for offensive language detection. *Knowledge-Based Systems*, 258: 109965.

Rosenthal, S.; Atanasova, P.; Karadzhov, G.; Zampieri, M.; and Nakov, P. 2021. SOLID: A Large-Scale Semi-Supervised Dataset for Offensive Language Identification. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 915–928. Online: Association for Computational Linguistics.

Rosenthal, S.; Farra, N.; and Nakov, P. 2017. SemEval-2017 Task 4: Sentiment Analysis in Twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, 502–518. Vancouver, Canada: Association for Computational Linguistics.

Roy, P. K.; Bhawal, S.; and Subalalitha, C. N. 2022. Hate speech and offensive language detection in Dravidian languages using deep ensemble framework. *Computer Speech & Language*, 75: 101386.

Siegel, A. A.; Nikitin, E.; Barberá, P.; Sterling, J.; Pullen, B.; Bonneau, R.; Nagler, J.; and Tucker, J. A. 2018. Measuring the prevalence of online hate speech, with an application to the 2016 US election.

Solovev, K.; and Pröllochs, N. 2022. Hate Speech in the Political Discourse on Social Media: Disparities Across Parties, Gender, and Ethnicity. In *Proceedings of the ACM Web Conference 2022*, WWW '22, 3656–3661. New York, NY, USA: Association for Computing Machinery. ISBN 9781450390965.

Talat, Z.; Thorne, J.; and Bingel, J. 2018. *Bridging the Gaps: Multi Task Learning for Domain Transfer of Hate Speech Detection*, 29–55. Cham: Springer International Publishing. ISBN 978-3-319-78583-7.

Toraman, C.; Şahinuç, F.; and Yilmaz, E. 2022. Large-Scale Hate Speech Detection with Cross-Domain Transfer. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, 2215–2225. Marseille, France: European Language Resources Association.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All You Need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, 6000–6010. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781510860964.

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 *Proceedings of the Third Workshop on Abusive Language Online*, 80–93. Florence, Italy: Association for Computational Linguistics.

Vidgen, B.; Thrush, T.; Waseem, Z.; and Kiela, D. 2021. Learning from the Worst: Dynamically Generated Datasets to Improve Online Hate Detection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 1667–1682. Online: Association for Computational Linguistics.

von Däniken, P.; Deriu, J.; Tuggener, D.; and Cieliebak, M. 2022. On the Effectiveness of Automated Metrics for Text Generation Systems. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, 1503–1522. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Wiegand, M.; Siegel, M.; and Ruppenhofer, J. 2019. Overview of the GermEval 2018 Shared Task on the Identification of Offensive Language. In *Proceedings of GermEval 2018, 14th Conference on Natural Language Processing (KONVENS 2018)*, Proceedings of GermEval 2018, 14th Conference on Natural Language Processing (KONVENS 2018), Vienna, Austria – September 21, 2018, 1–10. Vienna, Austria: Austrian Academy of Sciences. ISBN 978-3-7001-8435-5.

Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; and Brew, J. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *CoRR*, abs/1910.03771.

Wulczyn, E.; Thain, N.; and Dixon, L. 2017. Ex Machina: Personal Attacks Seen at Scale. WWW '17, 1391–1399. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee. ISBN 9781450349130.

Zadrozny, B.; and Elkan, C. 2001. Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In Brodley, C. E.; and Danyluk, A. P., eds., *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001*, 609–616. Morgan Kaufmann.

Zadrozny, B.; and Elkan, C. 2002. Transforming Classifier Scores into Accurate Multiclass Probability Estimates. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, 694–699. New York, NY, USA: Association for Computing Machinery. ISBN 158113567X.

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)*, 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*, 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 *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, 1425–1447. Barcelona (online): International Committee for Computational Linguistics.

Zampieri, N.; Illina, I.; and Fohr, D. 2023. Improving Hate Speech Detection with Self-Attention Mechanism and Multi-Task Learning. In *LTC'23 - 10th Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics*. Poznan, Poland.

Paper Checklist

1. For most authors...
   (a) Would answering this research question advance science without violating social contracts, such as violating privacy norms, perpetuating unfair profiling, exacerbating the socio-economic divide, or implying disrespect to societies or cultures? Yes.
   (b) Do your main claims in the abstract and introduction accurately reflect the paper's contributions and scope? Yes.
   (c) Do you clarify how the proposed methodological approach is appropriate for the claims made? Yes. See Section 6
   (d) Do you clarify what are possible artifacts in the data used, given population-specific distributions? We use pre-existing datasets and do not have access to this information.
   (e) Did you describe the limitations of your work? Yes. See Section 7.
   (f) Did you discuss any potential negative societal impacts of your work? While our chosen application domain, hateful and offensive language, deals with content that is harmful by design, our method is a mere tool for analyzing and evaluating this phenomenon.
   (g) Did you discuss any potential misuse of your work? We develop and evaluate quantification methods, which are a foundational tool with a large range of applications.
   (h) Did you describe steps taken to prevent or mitigate potential negative outcomes of the research, such as data and model documentation, data anonymization, responsible release, access control, and the reproducibility of findings? We did not create any new corpora. The code for our experiments is available at https://github.com/vodezhaw/icwsm2024/ mentioned in Section 1.
   (i) Have you read the ethics review guidelines and ensured that your paper conforms to them? Yes.
2. Additionally, if your study involves hypotheses testing...
   (a) Did you clearly state the assumptions underlying all theoretical results? We do not provide any theoretical results.
   (b) Have you provided justifications for all theoretical results? We do not provide any theoretical results.
   (c) Did you discuss competing hypotheses or theories that might challenge or complement your theoretical results? We do not provide any theoretical results.
   (d) Have you considered alternative mechanisms or explanations that might account for the same outcomes observed in your study? No.
   (e) Did you address potential biases or limitations in your theoretical framework? Limitations are discussed in Section 7.
   (f) Have you related your theoretical results to the existing literature in social science? We do not provide any theoretical results.
   (g) Did you discuss the implications of your theoretical results for policy, practice, or further research in the social science domain? We do not provide any theoretical results.
3. Additionally, if you are including theoretical proofs...
   (a) Did you state the full set of assumptions of all theoretical results? We do not provide any theoretical results.
   (b) Did you include complete proofs of all theoretical results? We do not provide any theoretical results.
4. Additionally, if you ran machine learning experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? The URL (https://github.com/vodezhaw/icwsm2024/) is specified in Section 1.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? Yes. See Section 5.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? In Section 6 we give mean, median, and 95th percentile performances.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? No. Our experiments were not extraordinarily resource intensive.
   (e) Do you justify how the proposed evaluation is sufficient and appropriate to the claims made? Yes. See Section 6.
   (f) Do you discuss what is "the cost" of misclassification and fault (in)tolerance? This is somewhat dependent on the end-user's application domain. We point out the risks of bad prevalence estimation in Section 7.
5. Additionally, if you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? See Section 5.1 for data and Sections 5.2 and 3 for models.
   (b) Did you mention the license of the assets? We used existing datasets that were used for shared tasks. They are all intended to be used for research.
   (c) Did you include any new assets in the supplemental material or as a URL? The URL (https://github.com/vodezhaw/icwsm2024/) is specified in Section 1.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? We are working with pre-existing datasets.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? We are working with pre-existing datasets.
   (f) If you are curating or releasing new datasets, did you discuss how you intend to make your datasets FAIR (see FORCE11 (2020))? No new datasets were created.

(g) If you are curating or releasing new datasets, did you create a Datasheet for the Dataset (see Gebru et al. (2021))? No new datasets were created.

6. Additionally, if you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots? We use pre-existing data.

(b) Did you describe any potential participant risks, with mentions of Institutional Review Board (IRB) approvals? We use pre-existing data.

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? We use pre-existing data.

(d) Did you discuss how data is stored, shared, and deidentified? We use pre-existing data.

A Calibration Methods

	Random			Quantile		
	μ	Med.	95	μ	Med.	95
CPCC	**0.073**	0.049	**0.211**	0.085	0.059	0.275
CPCC-ISO	0.074	**0.048**	0.218	0.086	0.058	0.275
CPCC-HB10	0.086	0.054	0.273	0.096	0.064	0.297
CPCC-HB100	0.213	0.183	0.515	0.208	0.177	0.501

Table 10: *SAPE* of *CPCC* with different calibration methods. We report the mean (μ), median (Med.), and 95th percentile (95) of errors for all variants and sample selection methods.

In Section 4 we described the *CPCC* quantification method which relies on re-calibrating classifier scores. So far we have been using *Platt Scaling* (Platt et al. 1999) for this purpose. Here we repeat the main experiment described in Section 6.1 by considering additional calibration methods mentioned by Guo et al. (2017). For *CPCC-ISO*, we use *Isotonic Regression* (Zadrozny and Elkan 2002) instead of *Platt Scaling*. For *CPCC-HB10*, we use *Histogram Binning* (Zadrozny and Elkan 2001) with 10 bins and for *CPCC-HB100* we use 100 bins.

Table 10 shows the *SAPE* for the variants of *CPCC*. *CPCC-ISO* performs almost identically to *CPCC* with only slightly larger 95th percentile performance (0.218 versus 0.211). Both *Histogram Binning* methods perform worse. In particular, increasing the number of bins seems to decrease performance. For all variants, *random* sample selection outperforms *quantile* sample selection.

B Classifier Performance

In this section, we list all the sample-based performances of each of the 28 different classifiers. We report the ROC-AUC scores for the different classifiers. Note that the scores are highly diverse ranging from 0.469 to 0.989. Table 12 shows the scores for all *Electra* classifiers, Table 13 shows the scores for all *Twitter-RoBERTa* classifiers, Table 14 shows the scores for all *TD-IDF SVM* classifiers, and Table 15 shows the scores for the *Perspective API* classifier. We give an overview of the abbreviated dataset names in Table 11.

Name	Short Name
Davidson Hate & Offensive	DHO
Davidson Hate Only	DH
DGH	DGH
ETHOS	ETH
Ex Machina	ExM
HASOC 2019	HSC
HatEval 2019	HEv
Jigsaw	Jig
OLID	OL
SOLID	SOL
SWAD	SWD
WASSA	WSS

Table 11: Overview of the short names of the different datasets from Section 5.1 that are used in Tables 12, 13, 14, and 15.

	DHO	DH	DGH	Eth	ExM	HSC	HEv	Jig	OL	SOL	SWD	WSS
DHO	**0.989**	0.486	0.591	0.727	0.941	0.825	0.582	0.930	0.796	0.957	0.591	0.738
DH	0.442	**0.894**	0.639	0.769	0.852	0.720	0.621	0.800	0.711	0.696	0.620	0.670
DGH	0.718	0.679	**0.901**	**0.886**	0.573	0.610	0.737	0.554	0.487	0.531	0.485	0.510
ExM	0.937	0.587	0.552	0.763	**0.980**	**0.857**	0.603	0.966	0.885	**0.974**	0.631	0.773
HSC	0.876	0.623	0.543	0.733	0.938	0.833	0.598	0.904	0.835	0.913	**0.716**	0.737
HEv	0.920	0.457	0.657	0.830	0.899	0.741	**0.663**	0.890	0.759	0.857	0.642	0.759
Jig	0.913	0.546	0.543	0.775	0.978	0.855	0.601	**0.972**	0.885	**0.974**	0.600	0.788
OL	0.914	0.544	0.537	0.769	0.962	0.854	0.607	0.963	**0.898**	**0.974**	0.640	0.780
WSS	0.901	0.589	0.529	0.664	0.885	0.755	0.582	0.878	0.768	0.905	0.674	**0.869**

Table 12: ROC-AUC performance of *Electra* classifiers. Rows indicate the training set used, while columns indicate the target test set. We show the highest ROC-AUC achieved for each test set in bold. We abbreviated the dataset names, please refer to Table 11.

	DHO	DH	DGH	Eth	ExM	HSC	HEv	Jig	OL	SOL	SWD	WSS
DHO	**0.984**	0.469	0.558	0.703	0.941	0.816	0.588	0.928	0.820	0.968	0.592	0.699
DH	0.397	**0.824**	0.583	0.656	0.632	0.575	0.541	0.588	0.598	0.473	0.557	0.551
DGH	0.736	0.708	**0.878**	**0.834**	0.519	0.604	**0.743**	0.474	0.431	0.448	0.483	0.500
ExM	0.841	0.635	0.515	0.704	0.939	0.790	0.524	0.942	0.786	0.941	0.612	0.708
HSC	0.886	0.675	0.548	0.734	**0.951**	**0.873**	0.574	0.932	0.815	0.929	0.672	0.753
HEv	0.927	0.412	0.633	0.786	0.857	0.728	0.573	0.819	0.688	0.756	**0.701**	0.684
Jig	0.721	0.524	0.523	0.526	0.621	0.550	0.536	0.692	0.580	0.723	0.508	0.519
OL	0.906	0.524	0.506	0.760	0.948	0.858	0.607	**0.954**	**0.890**	**0.970**	0.622	0.836
WSS	0.933	0.655	0.522	0.622	0.845	0.649	0.534	0.870	0.628	0.861	0.641	**0.862**

Table 13: ROC-AUC performance of *Twitter-RoBERTa* classifiers. Rows indicate the training set used, while columns indicate the target test set. We show the highest ROC-AUC achieved for each test set in bold. We abbreviated the dataset names, please refer to Table 11.

	DHO	DH	DGH	Eth	ExM	HSC	HEv	Jig	OL	SOL	SWD	WSS
DHO	**0.982**	0.528	0.494	0.643	0.832	0.702	0.581	0.837	0.688	0.885	0.546	0.629
DH	0.366	**0.839**	0.555	0.670	0.733	0.676	0.548	0.709	0.592	0.619	0.564	0.549
DGH	0.483	0.547	**0.636**	0.612	0.384	0.435	0.559	0.399	0.491	0.296	0.515	0.514
ExM	0.879	0.604	0.503	**0.705**	0.957	0.820	0.580	0.953	0.767	0.955	**0.619**	0.709
HSC	0.639	0.538	0.421	0.568	0.645	0.734	0.556	0.567	0.567	0.818	0.592	0.590
HEv	0.858	0.395	0.541	0.696	0.683	0.618	**0.625**	0.692	0.622	0.703	0.577	0.602
Jig	0.875	0.611	0.507	0.713	**0.965**	**0.833**	0.576	**0.963**	0.796	**0.962**	0.600	**0.739**
OL	0.843	0.509	0.454	0.693	0.858	0.809	0.597	0.877	**0.802**	0.953	0.552	0.676
WSS	0.698	0.519	0.457	0.572	0.776	0.727	0.576	0.743	0.619	0.843	0.565	0.660

Table 14: ROC-AUC performance of *TF-IDF SVM* classifiers. Rows indicate the training set used, while columns indicate the target test set. We show the highest ROC-AUC achieved for each test set in bold. We abbreviated the dataset names, please refer to Table 11.

DHO	DH	DGH	Eth	ExM	HSC	HEv	Jig	OL	SOL	SWD	WSS
0.957	0.594	0.649	0.843	0.980	0.885	0.623	0.978	0.902	0.984	0.612	0.862

Table 15: ROC-AUC performance of *Perspective API* on our various test sets. We abbreviated the dataset names, please refer to Table 11.