Is More Data Better? Using Transformers-Based Active Learning for Efficient and Effective Detection of Abusive Language

Anonymous NAACL-HLT 2021 submission

Abstract

Annotating abusive language content can cause psychological harm; yet, most machine learning research has prioritized efficacy (i.e., F1 or accuracy scores) while little research has analyzed data efficiency (i.e., how to minimize annotation requirements). In this paper, we use a series of simulated experiments over two datasets at varying percentages of abuse to demonstrate that transformers-based active learning is a promising approach that maintains high efficacy but substantially raises efficiency, requiring a fraction of labeled data to reach equivalent performance to passive training over the full dataset.

1 Introduction

Online abuse, such as hate and harassment, can inflict psychological harm on victims (Gelber and McNamara, 2016), disrupts communities (Mohan et al., 2017) and even lead to physical attacks (Williams et al., 2019). Automated solutions can be used to detect abusive content at scale, helping to tackle this growing problem (Gillespie, 2020). An effective model is one which makes few misclassifications. Both false positives and negatives create a risk of harm: false negatives mean that users are not fully protected from abuse while false positives could lead to free expression being constrained. Models to automatically detect abuse are trained to maximize their efficacy using supervised learning techniques over datasets consisting of thousands of labeled examples. Collecting large amounts of social media data is relatively cheap and easy, but annotating data is expensive, logistically complicated and creates a risk of inflicting psychological harm on annotators through vicarious trauma (Steiger et al., 2021). An efficient model, which achieves high levels of performance with few labeled examples, is thus highly desirable for abusive content detection. Two developments have promise for efficient training: 1) for model architecture, pre-trained transformer models can be fine-tuned on relatively small datasets for specific tasks (Devlin et al., 2018; Qiu et al., 2020); 2) for data acquisition, active learning (AL) selects entries for annotation only if they are ‘informative’ (Lewis and Gale, 1994; Settles, 2009). In this paper, we evaluate these developments for building effective and efficient abusive language detection models.

Most papers on automated abuse detection use fully supervised learning (see surveys Ayo et al., 2020; Vidgen and Derczynski, 2020; Fortuna and Nunes, 2018). Even those using transformers, rarely take advantage of their efficient fine-tuning capabilities (e.g. Mutanga et al., 2020; Elmadany et al., 2020; Safi Samghabadi et al., 2020; Mozafari et al., 2019). Some use alternative data acquisition approaches such as adversarial training (Vidgen et al., 2021; Kirk et al., 2021; Xia et al., 2020) and traditional AL (Bashar and Nayak, 2021; Abidin et al., 2021; Charitidis et al., 2020; Mollas et al., 2020; Rahman et al., 2021). To our knowledge, only one paper uses transformers-based AL with an abusive language dataset (Ein-Dor et al., 2020), but the benefit of AL on other classification tasks is clear (Schröder et al., 2021b; Ein-Dor et al., 2020; Yuan et al., 2020). For AL with abusive language, class imbalance is a pressing issue as, although extremely harmful, online abuse is relatively rare...
(Rahman et al., 2021; Vidgen et al., 2019). Prior work only tests datasets at their given class imbalances, and has not disentangled how class imbalance and data features affect the utility of, and design choices needed for, efficient AL.

The class imbalance in real-world settings is unchangeable; so, we use two labeled abusive language datasets and artificially-rebalance each dataset at varying percentages of abuse. Practitioners have to make numerous decisions in the AL process such as the classifier and query strategy used. To assess how these design choices affect the utility of AL for abuse detection, we present a series of simulated experiments.¹ We measure efficacy using the F1 score as well as efficiency using the number of examples needed to reach 90% of supervised learning performance over the full dataset. We find that more data is not always better and can actually be worse. For all class imbalances and for both a BERT-based model and a SVM, AL achieves high performance with only a few hundred examples. AL over 3% of a 20k dataset can even surpass the F1 of a model passively trained over the full dataset by more than 5 percentage points (Fig. 1).

## 2 Methods

### 2.1 Active Learning Set-Up

AL typically consists of four components: 1) a classification model, 2) pools of unlabeled data \(\mathcal{U}\) and labeled data \(\mathcal{L}\), 3) a query strategy for identifying data to be labeled, and 4) an oracle (e.g., human annotators) to label the data. First, seed examples are taken from \(\mathcal{U}\) and sent to the oracle for labeling. These examples are used to initialize the classification model. This is referred to as a ‘cold start’. Second, batches of examples are iteratively sampled from the remaining unlabeled pool, using a query strategy to estimate their ‘informativeness’. Each queried batch is labeled and added to \(\mathcal{L}\). Finally, the classifier is re-trained over \(\mathcal{L}\).²

### 2.2 Dataset Selection and Processing

For feasibility, we use existing labeled datasets but withhold the labels until the model requests their annotation. This allows us to reproduce the process of cold-start and batch selection without labeling new data. We surveyed publicly available, annotated datasets for abusive language detection.³ Of these, two datasets were sufficiently large and contained enough abusive instances to facilitate our experimental approach. The wiki dataset (Wulczyn et al., 2017) contains comments from Wikipedia editors, labeled for whether they contain personal attacks. A pre-defined test set is given; so, we take our test instances from this set. The tweets (Founta et al., 2018) dataset contains tweets which have been assigned to one of four classes. We binarize by combining the abusive and hate speech classes (=1) and the normal and spam classes (=0). A pre-defined test set is not available so we set aside 10% of the data for testing that is never used for training.

To disentangle the merits of AL across class imbalances, we construct three new datasets for both wiki and tweets that have different class distributions: 50% abuse, 10% abuse and 5% abuse. This creates 6 datasets in total (see Tab. 1). To ensure we have sufficient positive instances for all imbalances, we assume that the unlabeled pool has 20,000 examples.⁴ Our AL strategies iterate over 2,000 examples as we find in prior experiments that further iterations did not affect performance.⁵

### 2.3 Evaluation

As a baseline, we use the passive supervised macro-F1 score over the full dataset of 20,000 (F1\textsubscript{20k}). For each AL strategy, we measure efficiency on the held-out test set as the number of examples needed to surpass 90% of F1\textsubscript{20k}, which we call N\textsubscript{90}.⁶ For

---

³https://hatespeechdata.com

⁴The wiki dataset has 10,834 abusive entries; so, at 50% abuse, the upper limit on a rebalanced pool is 21,668.

⁵AL experiments are implemented in Python using the small-text library (Schröder et al., 2021a)

⁶To fairly compare models, we calculate N\textsubscript{90} relative to best F1\textsubscript{20k}, (achieved by dBERT in all cases).
efficacy, we use the maximum F1 score achieved by each AL strategy, which we call $F_{1AL}$.

### 2.4 Experimental Parameters

Table 2: The best AL parameters and performance for each classifier (transformers vs SVM).

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Classifier</th>
<th>Best AL Combinations$^*$</th>
<th>F1$_{20k}$</th>
<th>F1$_{AL}$</th>
<th>$N_{90}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>wiki5</td>
<td>dBERT</td>
<td>20 Random 50 LC</td>
<td>0.920</td>
<td>0.922</td>
<td>170</td>
</tr>
<tr>
<td></td>
<td>SVM</td>
<td>20 Random 50 LC</td>
<td>0.875</td>
<td>0.838</td>
<td>1520</td>
</tr>
<tr>
<td>wiki10</td>
<td>dBERT</td>
<td>20 Heuristic 50 LC</td>
<td>0.859</td>
<td>0.866</td>
<td>170</td>
</tr>
<tr>
<td></td>
<td>SVM</td>
<td>20 Heuristic 50 LC</td>
<td>0.809</td>
<td>0.810</td>
<td>320</td>
</tr>
<tr>
<td>wiki5</td>
<td>dBERT</td>
<td>20 Heuristic 50 LC</td>
<td>0.807</td>
<td>0.855</td>
<td>220</td>
</tr>
<tr>
<td></td>
<td>SVM</td>
<td>20 Heuristic 50 LC</td>
<td>0.785</td>
<td>0.780</td>
<td>170</td>
</tr>
<tr>
<td>tweets5</td>
<td>dBERT</td>
<td>20 Random 50 LC</td>
<td>0.939</td>
<td>0.939</td>
<td>170</td>
</tr>
<tr>
<td></td>
<td>SVM</td>
<td>20 Random 50 LC</td>
<td>0.931</td>
<td>0.926</td>
<td>220</td>
</tr>
<tr>
<td>tweets10</td>
<td>dBERT</td>
<td>20 Heuristic 50 LC</td>
<td>0.904</td>
<td>0.902</td>
<td>220</td>
</tr>
<tr>
<td></td>
<td>SVM</td>
<td>20 Random 50 LC</td>
<td>0.893</td>
<td>0.901</td>
<td>170</td>
</tr>
<tr>
<td>tweets5</td>
<td>dBERT</td>
<td>200 Heuristic 50 LC</td>
<td>0.844</td>
<td>0.856</td>
<td>300</td>
</tr>
<tr>
<td></td>
<td>SVM</td>
<td>20 Heuristic 50 LC</td>
<td>0.825</td>
<td>0.830</td>
<td>170</td>
</tr>
</tbody>
</table>

Notes: $^*$ global metric from passive training over the full dataset
$^*$ calculated by averaging the rank performance on $F_{1AL}$, $N_{90}$.

For each artificially-rebalanced dataset (x5), we vary 5 experimental parameters giving a total of 432 unique experimental runs, each of which we repeat with 3 random seeds and average. Tab. 2 shows the best parameters for each dataset and each classifier. We now briefly explain the experimental variables.$^7$ For clarity, we focus on wiki, and evaluate experimental parameters for transformers-based AL. Results for tweets are similar.

#### Seed and Batch Size

We test two choices for the size of the seed used in the cold start (20, 200), and three choices for batch size (50, 100, 500). We find AL is more efficient with smaller seeds and batch sizes. The F1 score achieved with a seed of 20 and 4 AL iterations of 50 ($|L| = 220$) exceeds that reached with a seed of 200 and 0 iterations ($|L| = 200$) by 55pp for wiki50, 4pp for wiki10, and 10pp for wiki5. Batch sizes of 100 and 500 are less efficient than 50, with 700–1,100 and 150–200 more examples needed for $N_{90}$, respectively.

#### Cold Start

We evaluate two choices to select the examples for the seed. (1) Random: Seed examples are randomly selected. This could result in no abusive content being sampled with imbalanced data. (2) Heuristics: Seed examples are selected using keywords ($n = 652$), taken from the abusive language literature (ElSherief et al., 2018a,b; Gabriel, 2018; Davidson et al., 2017).$^8$ For wiki50, random- and heuristics-based initialization achieve equivalent $N_{90}$. However, with a seed of 20, a third of randomly-initialized experiments fail on wiki10 and all experiments fail for wiki5. This shows that when the data is imbalanced, a random seed is sub-optimal because both class labels are not observed.

#### Classifier and Query Strategy

For transformers-based AL, we use distil-robERTa (dBERT), which is computationally efficient and performs competitively to larger transformer models (Sanh et al., 2019; Schröder et al., 2021b). As a baseline for traditional AL without pre-trained models, we use a linear support vector machine (SVM). Appendix A presents details of model training. In conjunction with these models, we present the impact of Least-Confidence (LC), an active data acquisition strategy that selects items close to the decision boundary (Lewis and Gale, 1994).$^9$ For comparison, we randomly sample items from the unlabeled pool at each iteration. When training over the full dataset, dBERT always outperforms SVM, models have worse performance on more imbalanced datasets, and wiki is harder to predict than tweets (Tab. 2). In all cases, LeastConfidence outperforms the random baseline, and the gain is larger for higher imbalances: for wiki10 and wiki5, $N_{90}$ is lower by 150 and 100 examples, respectively.

### 3 Results

#### Efficiency

For each dataset, we find active strategies that need just 170 examples (0.8% of the full dataset) to reach 90% of passive supervised learning performance (see Tab. 2).

#### Efficacy

AL can even outperform passive learning. For all but one dataset (tweets10), dBERT with LeastConfidence achieves a higher F1 score over just 2,000 examples than passive supervised training over the whole dataset ($F_{1AL} \geq F_{120k}$ in Tab. 2). For wiki5, it achieves 5pp higher (Fig. 1).

#### The Effect of Pre-Training

We find AL makes more of a contribution to an SVM than to dBERT, shown by the larger gap to the random baseline (Fig. 2). With its extensive pre-training, dBERT achieves high performance on few examples, even if randomly selected. Nonetheless, for high imbalances, an AL component still enhances dBERT performance above the random baseline, requiring...

$^7$See Appendix C for additional detail. In each figure, we present the mean run (line) and standard deviation (shaded).

$^8$See Appendix B for details of keyword sampling.

$^9$We test further strategies: GreedyCoreSet (Sener and Savarese, 2017) and EmbeddingKMeans (Yuan et al., 2020), but LeastConfidence outperformed them (see Appendix C).
Train Balance
Train Balance
Macro F1
Macro F1
0.0
0.5
0.0
0.5
0.75
0.50
0.75
1.00
0.25
0.75
0.25
1.00
0.00
0.75

Figure 2: The contribution of pre-trained models vs active data acquisition.

Figure 3: Label imbalance at each iteration (dBERT).

150 and 100 fewer examples for N90, and achieving 2pp and 4pp higher F1 score, for wiki5 and wiki10 respectively.

Train Distribution To assess why active learning is more impactful with imbalanced data, we evaluate the imbalance of the labeled pool at each iteration (Fig. 3). In all class imbalances, the random baseline tends to the dataset distribution as expected. For imbalanced data, the LeastConfidence strategy actively selects abusive examples from the pool and tends toward a balanced distribution.

Out-of-domain Testing Our results show an AL strategy with just 1.5% of the dataset can reach, and even surpass, the F1 score of passive training on the full dataset. This raises a risk that the models are overfitting and may not generalize. We take the models trained on each of the three class imbalances for wiki and test them on their equivalent tweets datasets, and vice versa. As with in-domain results, models trained on wiki and applied to tweets reach F120k within a few iterations. The gap between LeastConfidence and the random baseline is even larger for out-of-domain evaluation versus in-domain (Fig. 4). This suggests that AL does not result in overfitting. A similar pattern is observed for all datasets, including training on tweets and applying to wiki (see Appendix D).

4 Discussion

Coupling pre-trained transformers with AL can create models that require significantly fewer examples to reach equivalent performance to passive training over the full dataset. Note that by training a new transformer model from scratch in each iteration, AL likely has a larger environmental footprint (Bender et al., 2021). Traditional AL (SVM) remains a competitive strategy, especially for the tweets dataset at higher imbalances, suggesting active data acquisition can substantially assist simpler model architectures.

Our findings are subject to some limitations: 1) we evaluate against two datasets with pre-existing labels. The wiki dataset samples banned comments and tweets samples with keywords and sentiment analysis. This reduces the linguistic diversity of the data, potentially making the task easier to learn in fewer examples; 2) we only use two models to represent transformers- and traditional AL; so, it is unclear whether a larger transformers model (e.g., BERT) or a simpler ML model (e.g., logistic regression) would reproduce similar results. Future work is needed to verify these findings against more diverse datasets and alternative model architectures.

Our key finding is that more data is not always better and in the scenarios we tested, it can be worse. These results show that more attention needs to be paid to how data is acquired; the current paradigm might be needlessly expensive and place annotators at unneeded risk of harm.
References


Ioannis Mollas, Zoe Chrysopoulou, Stamatis Karlos, and Grigorios Tsoumakas. 2020. *ETHOS: an Online Hate Speech Detection Dataset*.


Christopher Schröder, Andreas Niekler, and Martin Potthast. 2021b. *Uncertainty-based Query Strategies for Active Learning with Transformers*.


A Details of Dataset Processing and Model Training

We use two English-language datasets which were curated for the task of automated abuse detection (Wulczyn et al., 2017; Founta et al., 2018). The wiki dataset can be downloaded from [https://github.com/ewulczyn/wiki-detox](https://github.com/ewulczyn/wiki-detox) and is licensed under Apache License, Version 2.0. The tweets dataset can be downloaded with tweet ids from [https://github.com/ENCEASE2020/hatespeech-twitter](https://github.com/ENCEASE2020/hatespeech-twitter).

These datasets cover two different domains: Wikipedia and Twitter. Each dataset is cleaned by removing extra white space, dropping duplicates and converting usernames, URLs and emoji to special tokens.

We fine-tune distil-roBERTa using the transformers integration with the small-text python package (Wolf et al., 2019; Schröder et al., 2021). distil-roBERTa has six layers, 768 hidden units, and 82M parameters.

We encode input texts using the distil-roBERTa tokenizer, with added special tokens for usernames,
We use a heuristic to weakly label examples from sklearn with Adam optimizer. All other hyperparameters (Ein-Dor et al., 2020, e.g. see) and searching for Table 3: The effect of varied keyword density thresholds directives. We find a threshold of 5% best balances these reduces false positives but also decreases true positives. Using this threshold, examples are expected to be abusive if the percentage of keywords in token tokens exceeds 5% (see Appendix B). We then sample equal numbers of expected abusive and non-abusive examples from the pool and reveal their true labels.

**C Additional Experimental Analysis**

In Fig. 5, we present the learning curve and comparisons of each experimental variable for both datasets and classifiers to supplement the results discussed in the main paper. For completeness, we make all our experimental results available in a csv file at [GITHUB URL]. In each panel of Fig. 5, we vary one parameter whilst holding all others fixed. This allows us to evaluate the impact of one variable, ceteris paribus. Namely, the reference values are seed size of 20, a cold strategy of heuristics-based sampling, a batch size of 50, and a query strategy of LeastConfidence. In addition to the query strategies discussed in the main paper, we evaluate two further strategies coupled with dBERT: 1) GreedyCoreSet is a data-based diversity strategy which selects items representative of the full set (Sener and Savarese, 2017) and 2) EmbeddingKMeans is a data-based diversity strategy which uses a dense embedding representation (such as BERT embeddings) to cluster and sample from the nearest neighbors of the k centroids (Yuan et al., 2020). On our datasets, these two strategies are high performing in terms of the maximum F1 score they achieve over 2,000 examples, but take longer to learn and are less efficient than Least Confidence.

**B Sampling with Keywords**

We use a heuristic to weakly label examples from the unlabeled pool to be selected for the initial seed. Keywords are a commonly-used approach (Ein-Dor et al., 2020, e.g. see) and searching for text matches is computationally efficient over a large pool of unlabeled examples. However, the keyword heuristic only approximates the true label and can introduce biases due to non-abusive use of offense and profanities. In our data, we rely on a keyword density measure \( K \) which equals the number of keyword matches over the total tokens in a text instance. We then experiment with varied thresholds of \( K \in [1\%, 5\%, 10\%, 25\%] \) for a weak label of abusive text. A higher threshold reduces false positives but also decreases true positives. We find a threshold of 5% best balances these directional effects (see Tab. 3). Making predictions using a keyword heuristic with 5% cut-off achieves an F1-score relative to the true labels of 69% for wiki and 80% for tweets. Using this threshold, expected abusive and non-abusive examples from the pool and reveal their true labels.

**D Generalizability of Performance**

In the main paper, we present the results of a model trained on wiki5, and evaluated on tweets5. In Fig. 6, we demonstrate the equivalent results for all class imbalances and both datasets. In general, tweets is harder to predict than wiki, so see a larger change in performance when training on tweets and evaluating on wiki. For 50% and 10% abuse, performance is similar across test sets. For 5% abuse, there is a larger difference especially for the random baseline. However, in all cases, the performance of the LeastConfidence strategy generalizes well to out-of-domain testing.
Figure 5: Learning curves per dataset-class imbalance pair showing the effect of isolated experimental variables on traditional (SVM) and transformers-based (dBERT) active learning.
Figure 6: For each class imbalance, we train a distil-roBERTa model on wiki and evaluate it on the tweets dataset, and vice versa. In each panel, we show a model’s own test set performance alongside performance on the cross test set to assess generalizability.