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-

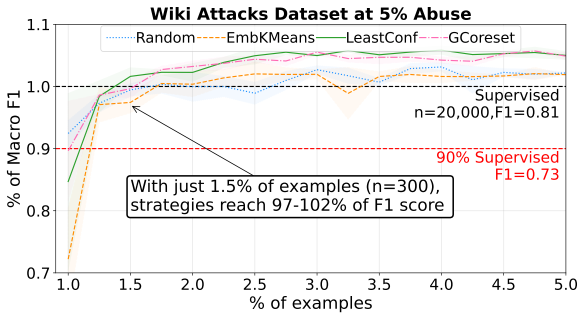


Figure 1: Transformers-based active learning beats fully-supervised baseline with 1.5% of the 20,000 examples.

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.

Table 1: Summary of source datasets (in gray) and their artificially rebalanced versions.

		Train†		Test*	
Dataset	**Imbalance**	**abuse**	**non-abuse**	**abuse**	**non-abuse**
wiki	12%	10,834	81,852	2,756	20,422
wiki50	50%	10,000	10,000	2,500	2,500
wiki10	10%	2,000	18,000	500	4,500
wiki5	5%	1,000	19,000	250	4,750
tweets	32%	28,955	61,041	3,160	6,840
tweets50	50%	10,000	10,000	2,500	2,500
tweets10	10%	2,000	18,000	500	4,500
tweets5	5%	1,000	19,000	250	4,750

Notes: † Train is used as the unlabeled pool ($n = 20{,}000$)
* Test is used for held-out evaluation ($n = 5{,}000$)

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 ($\mathbf{F1_{20k}}$). For each AL strategy, we measure efficiency on the held-out test set as the number of examples needed to surpass 90% of $\mathbf{F1_{20k}}$, which we call $\mathbf{N_{90}}$.[6] For

[1]Code and experimental results available at [Github URL].

[2]We train from scratch to avoid overfitting to previous iterations (Ein-Dor et al., 2020; Hu et al., 2018).

[3]https://hatespeechdata.com

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

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

[6]To fairly compare models, we calculate $\mathbf{N_{90}}$ relative to *best* $\mathbf{F1_{20k}}$ (achieved by dBERT in all cases).

efficacy, we use the maximum F1 score achieved by each AL strategy, which we call $\mathbf{F1_{AL}}$.

2.4 Experimental Parameters

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

Dataset	Classifier	Best AL Combinations*				Metrics		
		Seed	Cold	Batch	Query	$\mathbf{F1_{20k}}$†	$\mathbf{F1_{AL}}$	$\mathbf{N_{90}}$
wiki50	dBERT	20	Random	50	LC	**0.920**	**0.922**	**170**
	SVM	20	Random	50	LC	0.875	0.838	1520
wiki10	dBERT	20	Heuristic	50	LC	**0.859**	**0.866**	**170**
	SVM	20	Heuristic	50	LC	0.809	0.810	320
wiki5	dBERT	20	Heuristic	50	LC	**0.807**	**0.855**	220
	SVM	20	Heuristic	50	LC	0.785	0.780	**170**
tweets50	dBERT	20	Random	50	LC	**0.939**	**0.939**	**170**
	SVM	20	Random	50	LC	0.931	0.926	220
tweets10	dBERT	20	Heuristic	50	LC	**0.904**	**0.902**	220
	SVM	20	Random	50	LC	0.893	0.901	**170**
tweets5	dBERT	200	Heuristic	50	LC	**0.844**	**0.856**	300
	SVM	20	Heuristic	50	LC	0.825	0.830	**170**

Notes: † global metric from passive training over the full dataset
* calculated by averaging the rank performance on $\mathbf{F1_{AL}}$, $\mathbf{N_{90}}$

For each artificially-rebalanced dataset (x6), 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 ($|\mathcal{L}| = 220$) exceeds that reached with a seed of 200 and 0 iterations ($|\mathcal{L}| = 200$) 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 $\mathbf{N_{90}}$, 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 $\mathbf{N_{90}}$. 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**, $\mathbf{N_{90}}$ 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 ($\mathbf{F1_{AL}} \geq \mathbf{F1_{20k}}$ 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).

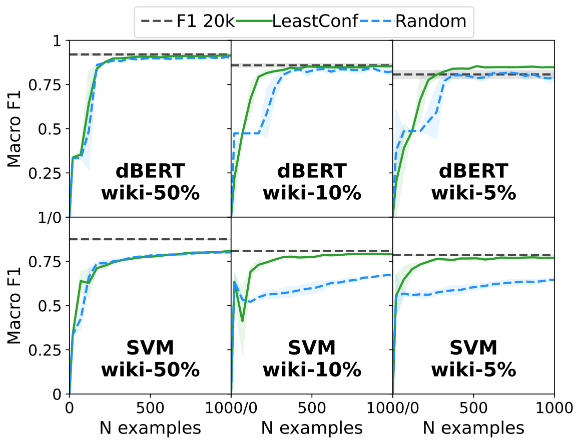


Figure 2: The contribution of pre-trained models vs active data acquistion.

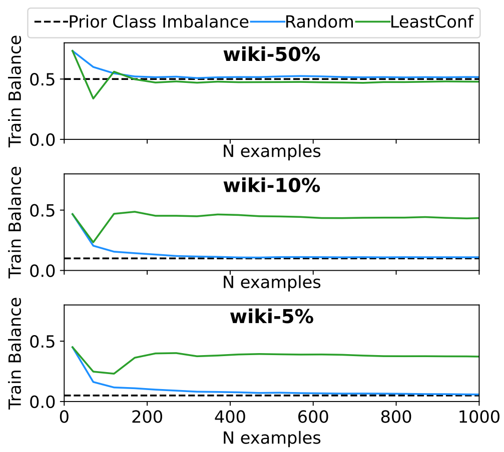


Figure 3: Label imbalance at each iteration (dBERT).

150 and 100 fewer examples for $\mathbf{N_{90}}$, 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-

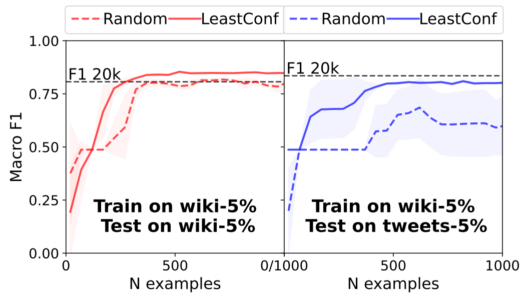


Figure 4: Cross-dataset generalization (dBERT).

domain results, models trained on **wiki** and applied to **tweets** reach $\mathbf{F1_{20k}}$ 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

Muhammad Ilham Abidin, Khairil Anwar Notodiputro, and Bagus Sartono. 2021. Improving Classification Model Performances using an Active Learning Method to Detect Hate Speech in Twitter. *Indonesian Journal of Statistics and Its Applications*, 5(1):26–38.

Femi Emmanuel Ayo, Olusegun Folorunso, Friday Thomas Ibharalu, and Idowu Ademola Osinuga. 2020. Machine learning techniques for hate speech classification of twitter data: State-of-the-art, future challenges and research directions. *Computer Science Review*, 38:100311.

Md Abul Bashar and Richi Nayak. 2021. Active Learning for Effectively Fine-Tuning Transfer Learning to Downstream Task. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 12(2).

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? . In *Conference on Fairness, Accountability, and Transparency (FAccT '21)*. ACM, New York, NY, USA.

Polychronis Charitidis, Stavros Doropoulos, Stavros Vologiannidis, Ioannis Papastergiou, and Sophia Karakeva. 2020. Towards countering hate speech against journalists on social media. *Online Social Networks and Media*, 17:100071.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated Hate Speech Detection and the Problem of Offensive Language. In *Proceedings of the 11th International AAAI Conference on Web and Social Media*, ICWSM '17, pages 512–515.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 1:4171–4186.

Liat Ein-Dor, Alon Halfon, Ariel Gera, Eyal Shnarch, Lena Dankin, Leshem Choshen, Marina Danilevsky, Ranit Aharonov, Yoav Katz, and Noam Slonim. 2020. Active Learning for BERT: An Empirical Study. In *Proceedings ofthe 2020 Conference on Empirical Methods in Natural Language Processing*, pages 7949–7962.

AbdelRahim Elmadany, Chiyu Zhang, Muhammad Abdul-Mageed, and Azadeh Hashemi. 2020. Leveraging affective bidirectional transformers for offensive language detection. *arXiv preprint arXiv:2006.01266*.

Mai ElSherief, Vivek Kulkarni, Dana Nguyen, Wang William Y., and Elizabeth Belding. 2018a. Hate Lingo: A Target-based Linguistic Analysis of Hate Speech in Social Media. In *Proceedings of the 12th International AAAI Conference on Web and Social Media*, ICWSM '18.

Mai ElSherief, Shirin Nilizadeh, Dana Nguyen, Vigna Giovanni, and Elizabeth Belding. 2018b. Peer to Peer Hate: Hate Instigators and Their Targets. In *Proceedings of the 12th International AAAI Conference on Web and Social Media*, ICWSM '18.

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

Antigoni-Maria Founta, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis. 2018. Large Scale Crowdsourcing and Characterization of Twitter Abusive Behavior. *12th International AAAI Conference on Web and Social Media, ICWSM 2018*, pages 491–500.

Robert J Gabriel. 2018. Full List of Bad Words and Top Swear Words Banned by Google.

Katharine Gelber and Luke McNamara. 2016. Evidencing the harms of hate speech. *Social Identities*, 22(3):324–341.

Tarleton Gillespie. 2020. Content moderation, AI, and the question of scale. *Big Data & Society*, 7(2):205395172094323.

Peiyun Hu, Zachary C. Lipton, Anima Anandkumar, and Deva Ramanan. 2018. Active Learning with Partial Feedback. *7th International Conference on Learning Representations, ICLR 2019*.

Hannah Rose Kirk, Bertram Vidgen, Paul Röttger, Tristan Thrush, and Scott A. Hale. 2021. Hatemoji: A Test Suite and Adversarially-Generated Dataset for Benchmarking and Detecting Emoji-based Hate.

David D Lewis and William A Gale. 1994. A Sequential Algorithm for Training Text Classifiers. In *SIGIR '94*, pages 3–12. Springer London, London.

Shruthi Mohan, Apala Guha, Michael Harris, Fred Popowich, Ashley Schuster, and Chris Priebe. 2017. The Impact of Toxic Language on the Health of Reddit Communities. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 10233 LNAI, pages 51–56. Springer, Cham.

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

Marzieh Mozafari, Reza Farahbakhsh, and Noel Crespi. 2019. A bert-based transfer learning approach for hate speech detection in online social media. In *International Conference on Complex Networks and Their Applications*, pages 928–940. Springer.

R Mutanga, Nalindren Naicker, and Oludayo O Olugbara. 2020. Hate speech detection in twitter using transformer methods. *International Journal of Advanced Computer Science and Applications*, 11(01).

Xi Peng Qiu, Tian Xiang Sun, Yi Ge Xu, Yun Fan Shao, Ning Dai, and Xuan Jing Huang. 2020. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences 2020 63:10*, 63(10):1872–1897.

Md Mustafizur Rahman, Dinesh Balakrishnan, Dhiraj Murthy, Mucahid Kutlu, and Matthew Lease. 2021. An Information Retrieval Approach to Building Datasets for Hate Speech Detection.

Niloofar Safi Samghabadi, Parth Patwa, Srinivas PYKL, Prerana Mukherjee, Amitava Das, and Thamar Solorio. 2020. Aggression and misogyny detection using BERT: A multi-task approach. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pages 126–131, Marseille, France. European Language Resources Association (ELRA).

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Christopher Schröder, Lydia Müller, Andreas Niekler, and Martin Potthast. 2021a. Small-text: Active Learning for Text Classification in Python.

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

Ozan Sener and Silvio Savarese. 2017. Active Learning for Convolutional Neural Networks: A Core-Set Approach. *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*.

Burr Settles. 2009. Computer Sciences Department Active Learning Literature Survey.

Miriah Steiger, Timir J Bharucha, Sukrit Venkatagiri, Martin J. Riedl, and Matthew Lease. 2021. The Psychological Well-Being of Content Moderators. pages 1–14.

Bertie Vidgen and Leon Derczynski. 2020. Directions in Abusive Language Training Data: Garbage In, Garbage Out. *PLoS ONE*, 15(12 December).

Bertie Vidgen, Helen Margetts, and Alex Harris. 2019. How much online abuse is there? A systematic review of evidence for the UK. Technical report, The Alan Turing Institute.

Bertie Vidgen, Tristan Thrush, Zeerak Waseem, and Douwe Kiela. 2021. Learning from the Worst: Dynamically Generated Datasets to Improve Online Hate Detection. In *ACL 2021 - 59th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*.

Matthew L. Williams, Pete Burnap, Amir Javed, Han Liu, and Sefa Ozalp. 2019. Hate in the Machine: Anti-Black and Anti-Muslim Social Media Posts as Predictors of Offline Racially and Religiously Aggravated Crime. *The British Journal of Criminology*, 60(1):93–117.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing.

Ellery Wulczyn, Nithum Thain, and Lucas Dixon. 2017. Ex Machina: Personal Attacks Seen at Scale. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1391–1399, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

Mengzhou Xia, Anjalie Field, and Yulia Tsvetkov. 2020. Demoting Racial Bias in Hate Speech Detection. In *Proceedings of the Eighth International Workshop on Natural Language Processing for Social Media*, pages 7–14, Stroudsburg, PA, USA. Association for Computational Linguistics.

Michelle Yuan, Hsuan-Tien Lin, and Jordan Boyd-Graber. 2020. Cold-start Active Learning through Self-supervised Language Modeling.

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` and is licensed under Apache License, Version 2.0. The **tweets** dataset can be downloaded with tweet ids from `https://github.com/ENCASEH2020/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., 2021a). 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,

Table 3: The effect of varied keyword density thresholds on F1, precision, false positive rate (FPR) and false negative rate (FNR).

K	F1	FPR	FNR
		wiki	
1.0%	76.0%	2.7%	52.8%
5.0%	69.0%	0.5%	71.8%
10.0%	91.0%	0.1%	87.4%
25.0%	49.0%	0.0%	98.4%
		Tweets	
1.0%	85.0%	4.5%	29.6%
5.0%	80.0%	2.9%	42.7%
10.0%	83.0%	0.9%	76.4%
25.0%	75.0%	0.2%	98.5%

URLs and emoji. All models were trained for 3 epochs with early stopping based on the validation set loss, a learning rate $2e-5$ and a weighted Adam optimizer. All other hyperparameters are set to their `small-text` defaults. In each active learning iteration, we use 10% of each labeled batch for validation. As a baseline to transformers-based AL, we use a support vector machine with no pre-training which we implement with `sklearn`. To encode a vector representation of input texts, we use a TD-IDF transformation fit to the training dataset.

All experiments were run on the JADE-2 cluster using one NVIDIA Tesla V100 GPU per experiment. For transformer-models, it took on average 1.5 hours to run each experiment. For SVM, it took less than a minute to run each experiment and these can be easily be run on a CPU. We repeat each experiment three times using three seeds to initialize a pseudo-random number generator.

B Sampling with Keywords

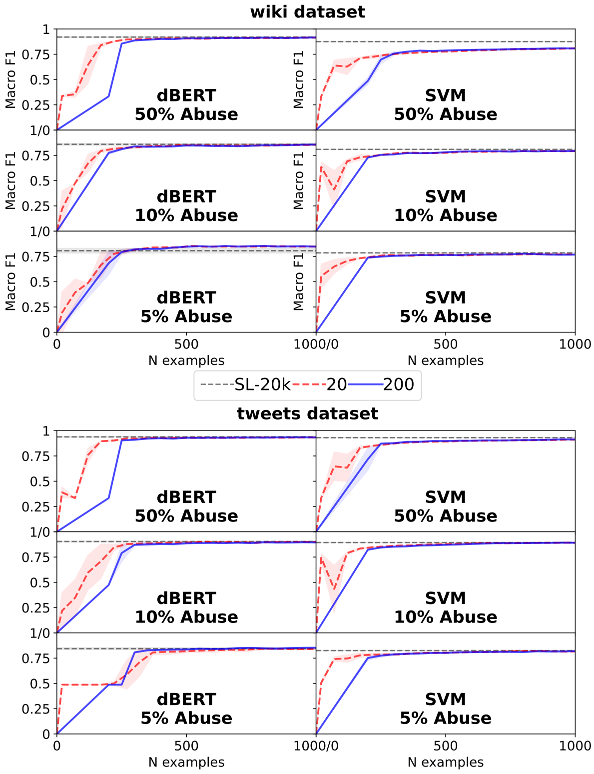
We use a heuristic to weakly label examples from the unlabeled pool to be selected for the initial seed. Keywords are a commonly-used approach (Ein-Dor et al., 2020, e.g. see) and searching for text matches is computationally efficient over a large pool of unlabeled examples. However, the keyword heuristic 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, 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

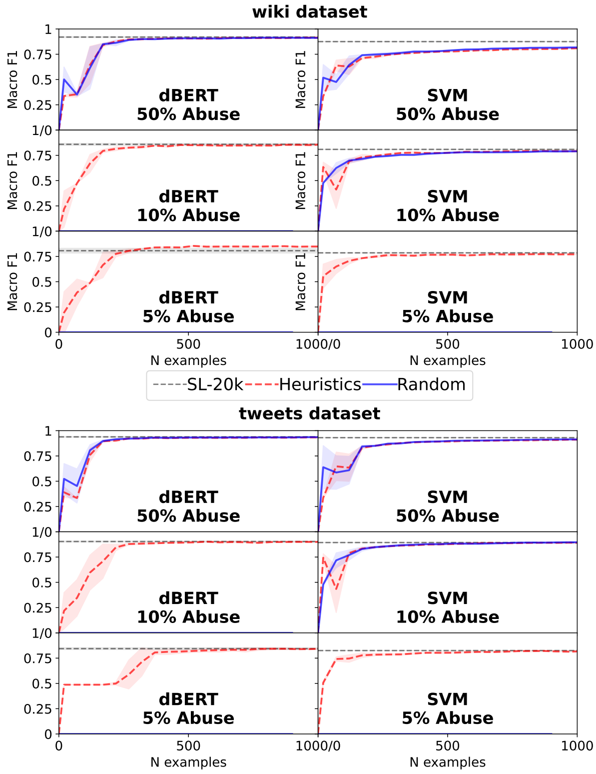
In Fig. 5, we present the learning curve and comparisons of each experimental variable for both datasets and classifiers to supplement the results discussed in the main paper. For completeness, we make all our experimental results available in a `csv` file at [GITHUB URL]. In each panel of Fig. 5, we 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 additional 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.

D Generalizability of Performance

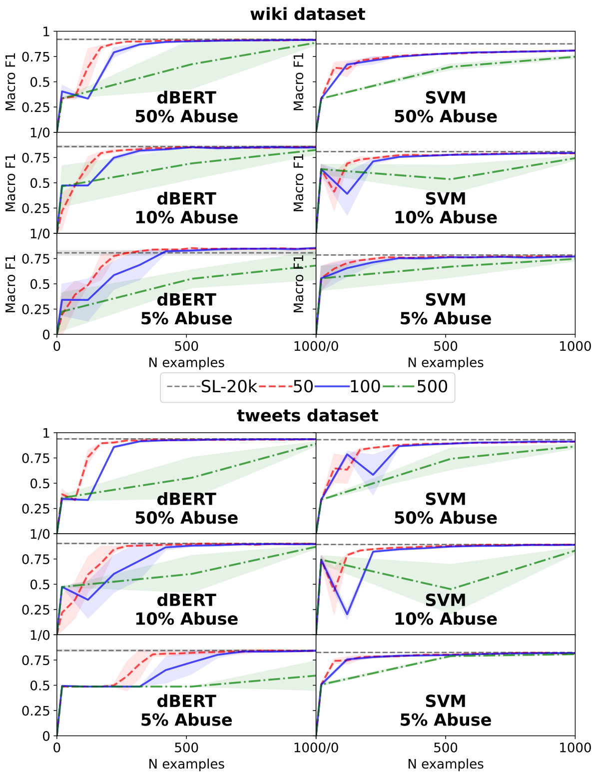
In the main paper, we present the results of a model trained on **wiki5**, and evaluated on **tweets5**. In Fig. 6, we demonstrate the equivalent results for all class imbalances and both datasets. In general, tweets is harder to predict than wiki, so see a larger change in performance when training 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.



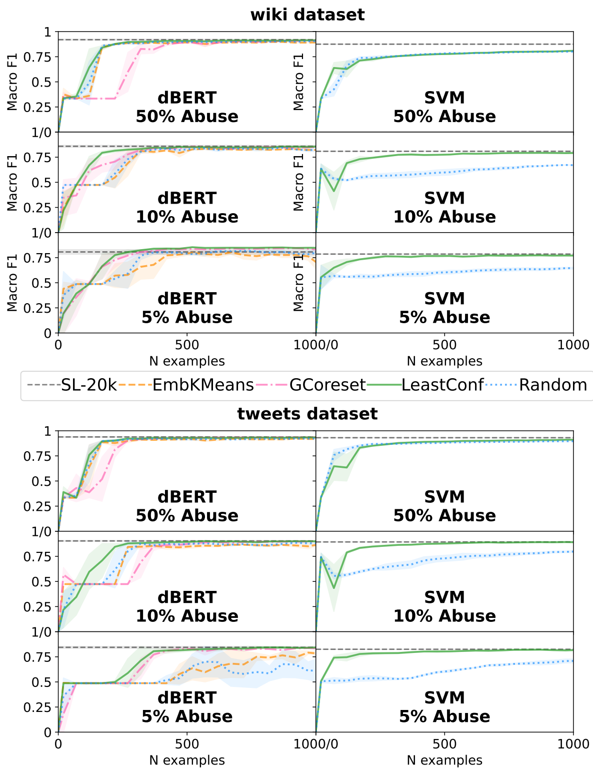
(a) Seed Size



(b) Cold Strategy

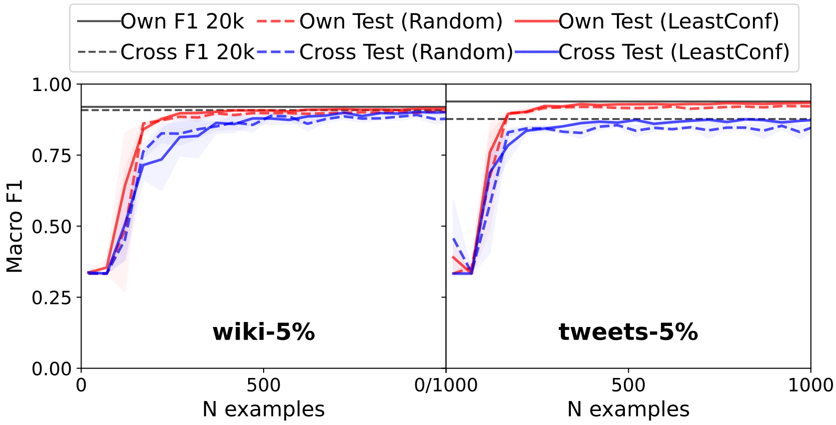


(c) Batch Size

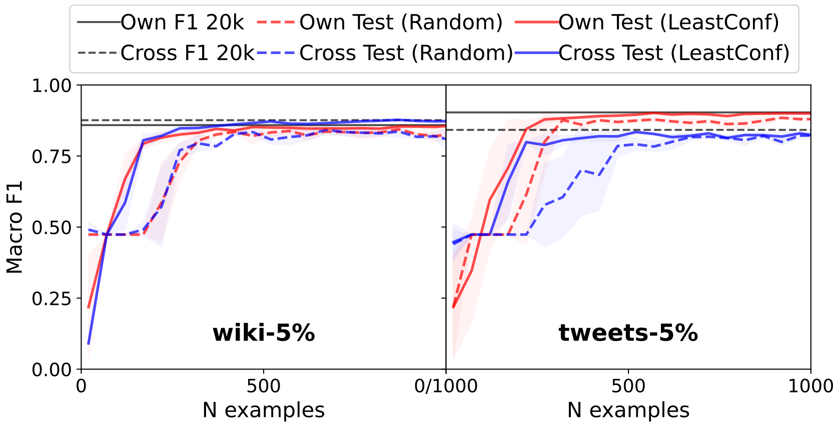


(d) Query Strategy

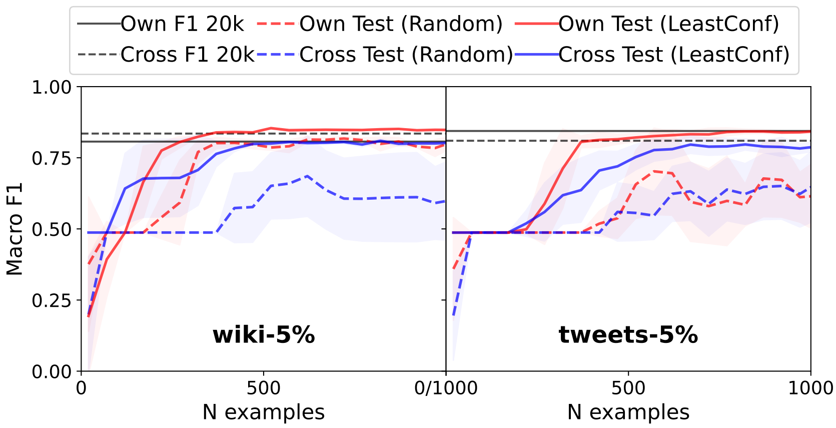
Figure 5: Learning curves per dataset-class imbalance pair showing the effect of isolated experimental variables on traditional (SVM) and transformers-based (dBERT) active learning.



(a) Models trained on 50% abuse



(b) Models trained on 10% abuse



(c) Models trained on 5%

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.