A Holistic Approach to Undesired Content Detection in the Real World

Warning: some content may contain racism, sexuality, or other harmful language.

Todor Markov*, Chong Zhang*, Sandhini Agarwal, Florentine Eloundou Nekoul, Theodore Lee, Steven Adler, Angela Jiang, Lilian Weng*

OpenAI

Abstract

We present a holistic approach to building a robust and useful natural language classification system for real-world content moderation. The success of such a system relies on a chain of carefully designed and executed steps, including the design of content taxonomies and labeling instructions, data quality control, an active learning pipeline to capture rare events, and a variety of methods to make the model robust and to avoid overfitting. Our moderation system is trained to detect a broad set of categories of undesired content, including sexual content, hateful content, violence, self-harm, and harassment. This approach generalizes to a wide range of different content taxonomies and can be used to create high-quality content classifiers that outperform off-the-shelf models.

1 Introduction

Recent advances in deep learning have accelerated the adoption of language models for socioeconomically valuable tasks in the real world (Devlin et al. 2019; Brown et al. 2020; Thoppilan et al. 2022). Both the system builders and users may benefit from a responsible deployment approach that includes moderating the models' outputs: First, model providers may want assurances that the models will not produce content disallowed by their policies. Second, customers of these models sometimes require control over content to mitigate the impact of sensitive use cases or to reduce brand risk. We believe that a strong undesired content classifier provides fine-grained control to enable use cases with sensitive needs and also lays the foundation for building safer AI systems in the wild, as it enables the capacity of moderating, evaluating, and guiding the models towards safer behavior.

Existing work on content detection either focuses mainly on a limited set of categories, including toxicity (Pavlopoulos et al. 2020; Gehman et al. 2020), hate speech (Kwok and Wang 2013; Davidson et al. 2017), and abusive content (Nobata et al. 2016; Vidgen et al. 2019); or is tailored towards a targeted use case, such as Perspective API (Jigsaw 2017) on online toxic comment moderation. There is increasing attention to understanding the risk areas of large language models via a more rigorous taxonomy (Weidinger et al. 2021), but the amount of work is still limited, especially when it comes to deploying language models for real-world applications. Here, we build a more comprehensive system for detecting a broad set of categories of undesired content, including sexual content, hateful content, violence, self-harm, and harassment, as well as severe subcategories under each top-level category. Large-scale content moderation systems and tooling exist on a number of platforms (YouTube 2019; Reddit 2022). We aim to provide a blueprint for creating such systems across a wide variety of use cases (Fig. 1).

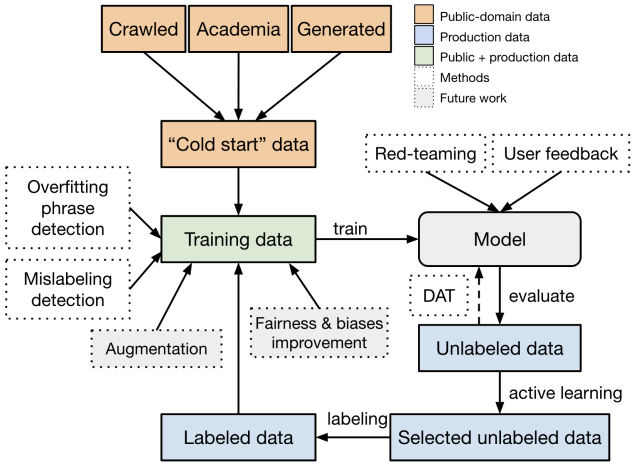

Figure 1: Overview of the model training framework.

Detecting undesired content is difficult due to a few challenges. First, there is no clearly and widely agreed-upon categorization of undesired content. Designing a detailed taxonomy for undesired content and operationalizing it for labeling purposes requires a lot of work. The categorization framework usually needs to clarify a significant number of corner cases to achieve high inter-rater agreement during labeling. This is further complicated by the subjectivity of some labeling decisions due to the different social and cultural backgrounds of human annotators. Second, a practical moderation system needs to process real-world traffic. Thus, a model bootstrapped from public data or academic datasets would not work well because there exists a big data distribution shift and taxonomy misalignment. Third, it is rare to en-

*These authors contributed equally to this work. Corresponding author: lilian@openai.com.

counter certain categories of undesired content in real-world settings. For example, among sampled user prompts we observed that only 0.04% of the cases included self-harm and 0.017% included hateful content involving threats. Hence, we need smart solutions to the cold start problem and effective ways to discover undesired samples.

Multiple components contribute to the success of building and deploying a practical, general moderation system into the real world. These include effectively establishing a chain of carefully polished and curated configurations for data collection, data labeling, model training, and active learning. Based on our experimentation, we find the following conclusions to be especially noteworthy.

- *Detailed instructions and quality control are needed to ensure data quality.* Labeling instructions that lack sufficient precision force annotators to rely on their subjective judgment, resulting in inconsistently labeled data that confuses the model. Regular calibration sessions are necessary to refine these instructions and ensure annotators are aligned with them. A poorly chosen quality metric can lead to data that hurts model performance. (§3.2)
- *Active learning is a necessity.* There is likely a large distribution shift between public data and the traffic from one's production system. Thus, it is critical to collect new training samples from the production traffic. Active learning can effectively expand the training dataset to capture a significantly (up to 22×) larger amount of undesired samples when dealing with rare events. This can lead to a performance improvement in the underlying model of up to 10× for rare categories. (§3.1 and §4.3)
- *Use public datasets with care.* Publicly available data might not lead to high quality performance for the problem in hand due to differences in taxonomy and training data distribution, but can be used to construct a noisy cold start dataset at the early stage. However, adding academic data into the training set may hurt the model performance at a later stage when there are enough properly labeled data samples. (§3.5)
- *Imbalanced training data can lead to incorrect generalization.* Deep learning models could easily overfit common phrases or templates. For example, the model can over-generalize to anything formatted as `"X is hateful"` if the data distribution is off-balance. We tackle this challenge by programmatically identifying overfitted phrases and by red-teaming via human trials. We then alter the training distribution by incorporating model-generated or human-curated synthetic data to patch the weakness. (§3.5 and §3.3)
- *Mistakes in data will happen and need to be managed.* Even with significant amounts of effort on data quality control, we still run into mislabeled examples. We explore different methods for identifying those cases, including cross-validation and hunting down common phrases causing the model to overfit via input reduction. (§3.2 and §3.5)

We aim to present a holistic approach to building a reliable and robust undesired content detection model for real-world applications. Our approach is meant to handle the scenario in which the type of the content to be detected is rarely observed in the real world. We hope that the lessons we learned are useful to others who work on similar problems. We release a dataset[1] containing text samples from publicly available data labeled according to our taxonomy. The model is accessible via the OpenAI Moderation endpoint[2].

2 Taxonomy

Given that the meaning of a sentence can depend on context, designing a universal taxonomy for undesired content is challenging and requires inherent trade-offs. Throughout this exercise, we aimed to make the dataset as broadly applicable as possible to maximize its utility across research and industrial contexts, but recognize it still ought to be critically evaluated prior to use in any context. The anatomy of what is undesired can vary by factors like use cases, users, regions, and others. Some sentences labeled as undesired according to our taxonomy (e.g., `"I will kill you"` would be labeled as containing violence) may not be considered undesired when appearing in another context, such as fiction.

Additionally, we aimed to capture layers of subcategorization and granularity. Much existing work lacks subcategorizations to capture a spectrum of severity, which is critical, since statements such as `"kill all [IDENT_GROUP]"` can have different repercussions compared to `"[IDENT_GROUP] are dishonest"`. The existence of subcategories can also greatly reduce the disagreement rate and corner cases during labeling, making it valuable for data quality control. In the following outline, we present our taxonomy for undesired content categorization, including five top-level categories and a spectrum of subcategories.

`S`: Sexual content Content that depicts explicit or implicit sexual behavior. Subcategories that are considered undesired: [`S3`] sexual content involving minors, [`S2`] content that depicts certain sexual activities which could be illegal[3] if they happened in real life today, and [`S1`] erotic sexual content that does not depict illegal activities. Subcategories that are not considered undesired: [`S0`] Non-erotic or contextualized sexual content, such as medical or sex education material.

`H`: Hateful content Content that is threatening, insulting, derogatory, and otherwise abusive content targeting specific chosen groups or members of the group due to their group identities. Subcategories that are considered undesired: [`H2`] hateful content calling for violence or threatening, [`H1`] derogatory stereotypes or support for hateful statements. Subcategories that are not considered undesired: [`H0.a`] neutral statement referring to group identity and [`H0.b`] contextualized hate speech, e.g., quote of individual's statement in order to provide commentary.

`V`: Violence Content that depicts or shows support for physical violence. Subcategories that are considered undesired: [`V2`] extremely graphic violence, [`V1`] threats or

[1] https://github.com/openai/moderation-api-release

[2] https://beta.openai.com/docs/guides/moderation

[3] This mapped most closely to what's illegal in USA.

support for violence. Subcategories that are not considered undesired: [V0] neutral depictions of contextualized violence.

SH: Self-harm Content that contains a graphic description or shows support for self-harm.

HR: Harassment Content that may be used to torment or annoy individuals in real life, or make harassment more likely to occur.

Our model learns to predict whether a given sample violates any of *8 chosen categories*, including all the top categories (S, H, V, SH, HR) and the three most severe subcategories (S3, H2, and V2).

3 Methods

3.1 Data Selection and Active Learning

Given that the moderation system needs to perform well in production, we incorporate production data into our training set via a three-stage procedure in an iterative fashion.

First, a large volume of production data is selected at random with any potential personally identifiable information (PII) masked. The latest model is used to score these samples and discover which ones may trigger any chosen categories.

In the second stage, we run a simple active learning strategy to select a subset of most valuable samples to be labeled out of the random samples extracted in stage one. The active learning strategy is composed of three parallel pipelines. The first one relies on random sampling such that some fraction of our data remains consistent with the underlying data distribution in production. The second one randomly selects from samples with model score above a certain threshold for each category to identify likely undesired data points. The last pipeline adopts a set of uncertainty sampling strategies (Lewis and Gale 1994; Lewis and Catlett 1994) to capture samples that the model is most uncertain about, where the model score for that category is closest to 0.5.

During the final stage, all the samples selected by different active learning strategies are aggregated and re-weighted based on statistics of certain metadata associated with it. The sampling weight is configured to be proportional to the square root of the sample count. This helps improve the diversity of selected samples with regard to the associated metadata. We update the sub-strategy mixture over time based on changes in the data distribution and categories that we want to improve the most at different stages.

3.2 Labeling and Quality Control

Data label correctness is critical to good model performance. Getting such data can be difficult given that our categories and the boundary lines between them are inherently subjective. However, certain interventions can significantly improve the quality of labeled data.

One important intervention for improving data quality - in terms of both consistent labels across different annotators as well as between annotators and researchers - is to make the labeling instructions as *well-defined* and *concrete* as possible. To make the instructions well-defined, we sought to design detailed definitions and design categories or subcategories to be as mutually exclusive as possible to minimize ambiguity. To make the instructions concrete, we hosted regular calibration sessions to review ambiguous edge cases and instances where external annotators and our internal auditors disagree, and provide numerous examples and clearer definitions around borderline cases. By minimizing subjective judgments, rules can be executed more consistently by annotators.

Regular, ongoing audits are necessary to ensure that labeled data have sufficient high quality. The choice of which samples to audit and what metrics to use to measure data quality is crucial. We found that selecting audit targets at random cannot maximize the value of auditing due to the imbalanced distribution across categories. The annotator-auditor agreement rate (i.e., accuracy) is suboptimal because undesired examples are rare events to encounter and the accuracy can be arbitrarily high due to the abundance of true negatives. Instead, in each chosen category, we randomly select 10 samples labeled as undesired and 10 samples with model probability greater than 50%. The former help capture false positive cases and the latter provide an estimation on recall. Then we compute the F-1 score for the chosen samples based on the annotator-assigned labels while using auditor-assigned labels as ground truth. This procedure performs much better in practice when certain categories of undesired data points are rare. Separation of metrics per category makes it easy to recognize category-specific issues and to retrain annotators accordingly.

Even with very clear labeling instructions and an effective audit procedure, mistakes in data are still unavoidable. To identify potentially mislabeled samples, we periodically split our current training dataset into two partitions, train separate models on each partition, and use each model to score the other half of the dataset. When the model prediction disagrees with the current ground-truth label, the sample in question gets flagged. A random portion of flagged samples is audited, and if more than 30% are identified as mislabeled, all flagged samples would get labeled again.

3.3 Synthetic Data

In addition to the data collection discussed above, we also use synthetic data to improve model performance on rare categories such as SH and to mitigate counterfactual bias towards certain demographic attributes (Kusner et al. 2017; Garg et al. 2019; Dwork et al. 2012; Ribeiro et al. 2020). Generating synthetic data via large pre-trained language models has shown to be an effective way for data augmentation (Anaby-Tavor et al. 2020; Kumar, Choudhary, and Cho 2020; Yoo et al. 2021) and it is particularly helpful when there is little to no initial data ("cold start") or when there are not enough undesired samples in the production traffic.

Zero-shot data for cold start. To kick start the active learning and labeling process, we need some initial data to build the first version of the model and train annotators. However, it is difficult to find existing public datasets on certain categories such as SH and V2. We tackle the problem by generating a synthetic dataset with zero-shot prompts

Example prompt
>> The text is about self harm, more specifically: **physical self-harm (cutting)**. >> The narrator is **third person (a friend)**. >> The narrator is **a young teenager (female)**. >> The act of self harm happens: **last week**. >> Type of text: **online forum post**. >> The following word should come up in the text: **camera**. >> The text should not contain the phrase "self harm" or "self-harm". >> The writing level is: **average, some errors**. >> The text starts below:

Table 1: Example zero-shot prompt template for generating synthetic SH data. The sections in bold are filled with random ingredients to encourage diversity.

on GPT-3. The prompts are constructed from human-crafted templates and we label the generated texts as the initial dataset. Table 1 provides an example prompt for SH.

Few-shot data for rare categories. Some sub-categories had minimal amounts of undesired data even after several iterations of active learning. To address this, we construct few-shot prompts with existing undesired examples and send the generated texts to be labeled. The generated texts are manually inspected to avoid bias amplification (Zhao et al. 2017). We observe a nontrivial performance improvement by incorporating the synthetic dataset.

Curated data to mitigate counterfactual bias. Similar to other NLP models, our models also suffer from counterfactual bias towards certain demographic attributes, as bias exists commonly in training data. For instance, "black women." was classified as hateful content with high confidence in earlier versions of the model. We mitigate the issue by curating a synthetic dataset with templates that tend to lead to hateful predictions, e.g., "[subject] is selfish/foolish/narrow-minded.". The [subject] could be filled with real demographic attributes (e.g., Latino) or random object names (e.g., Black blanket), which form hateful and safe samples, respectively. We observe that the curated dataset not only mitigates bias to some degree but also helps improve the model performance. For example, the average AUPRC on hateful content was improved from 0.417 to 0.551 by adding 69k curated synthetic examples. We believe this is because the contrastive setup of subjects in the templates encourages the model to infer the correct feature representations: negative descriptive words or individual identity groups alone are not enough to be considered hateful, and only when they appear together might they be considered hateful. Despite the observed improvements, the synthetic dataset also has limitations on bias and we will keep improving it in the future.

Large amount of noisy data does not help. To understand whether it is helpful to include a large amount of noisy synthetic data, we also generated zero-shot and few-shot examples twice the size of the existing labeled training dataset. For zero-shot examples, we set the label to positive or negative if the prompt asks the model to generate undesired or safe examples, respectively. For few-shot examples, we set the label to positive or negative if all of the few-shot examples are undesired or safe, respectively. Contrary to previous studies (Wang et al. 2021b; Schick and Schütze 2021), we found that mixing noisy synthetic data into training hurt model performance. It is worth noting that many existing studies on synthetic data experimented in the no-to-low data regime, where only a handful of labels are available. However, in our experiment, we have collected a large high-quality dataset and we suspect that noise introduced by synthetic data confuses the model and lowers the learning efficiency.

3.4 Domain Adversarial Training

Models trained directly on existing public NLP datasets do not perform well on our production traffic. This is probably due to the distribution difference between domains. For instance, examples from our production traffic are usually much longer and contain few-shot prompts, whereas existing public NLP datasets are usually shorter and often crawled from Wikipedia, Twitter, etc. (Vidgen and Derczynski 2020). To mitigate the problem, besides carefully tuning the mixture of public datasets and production data, we in addition apply Wasserstein Distance Guided Domain Adversarial Training (WDAT) to encourage the model to learn domain invariant representations (Arjovsky, Chintala, and Bottou 2017; Ganin et al. 2016).

We follow Shen et al. (2018) and approximate the Wasserstein distance by maximizing the loss of a domain critic head. Let $f_z(x) : \mathbb{R}^d \to \mathbb{R}^z$ be the feature extractor that maps the d-dimensional input into a z-dimensional embedding, $f_c(h) : \mathbb{R}^z \to \mathbb{R}^c$ be a multiclass classification head, and $f_d(h) : \mathbb{R}^z \to \mathbb{R}$ be the domain critic head that maps the embedding into real number. The domain critic loss between the source domain $\mathcal{D}_s$ and target domain $\mathcal{D}_t$ is defined as:

$$\mathcal{L}_d(\mathcal{D}_s, \mathcal{D}_t) = | \mathop{\mathbb{E}}_{x \in \mathcal{D}_s} f_d(f_z(x)) - \mathop{\mathbb{E}}_{x \in \mathcal{D}_t} f_d(f_z(x))|.$$

Combined with the regular classification loss $\mathcal{L}_c$, our objective is to solve the following minimax problem:

$$\min_{\theta_z, \theta_c} \{\mathcal{L}_c + \lambda \max_{\theta_d} \mathcal{L}_d\},$$

where $\theta_z, \theta_c, \theta_d$ are the parameters of f_z, f_c, f_d, respectively. Our model uses a transformer encoder as f_z.

We use the absolute value in $\mathcal{L}_d$ since the initial loss could be negative, and clip θ_d in a compact space $[-0.01, 0.01]$ to enforce the Lipchitz constraint. We set the balancing coefficient λ to 0.01. In experiments, WDAT achieves a more stable training compared to the original classifier-based approach (Arjovsky, Chintala, and Bottou 2017), and yields better performance on our production traffic (§4.4).

3.5 Model Probing

It is widely known that ML models are vulnerable to adversarial inputs and may make predictions based on seemingly irrelevant features (Szegedy et al. 2013; Goodfellow, Shlens, and Szegedy 2015; Alzantot et al. 2018; Zhang et al.

Hate	Example
1	`I want to k1ll all immigrants.`
1	`Why is it that all immigrants are repulsive?`
1	`D3ath to all immigrants.`
0	`Threatening to kill all immigrants will get you thrown in jail.`
...	

Table 2: Example data containing `"all immigrants."`.

2021). For instance, a sentiment classification model may make different predictions for `"a short and moving film."` and `"a short and moving movie."`, indicating that the model makes predictions based on `film/movie` rather than descriptive words. To make sure that our model performs as expected, we conduct tool-assisted model probing and human-in-the-loop red-teaming.

Key Tokens Probing. Our first goal is to verify whether the model makes predictions based on correct features. For instance, we observed that our models may incorrectly predict `"all immigrants."` as `H`, which is possibly because among 21 training examples that contain the phrase (from Röttger et al. 2021), 17 examples are undesired and only 4 examples are safe. Table 2 provides a few examples, where the expected behavior is to also take `k1ll` and `D3ath` into account. To identify such potentially overfitted key tokens, we apply input reduction (Feng et al. 2018) on the training dataset by removing tokens greedily until the prediction drops below 0.8. On average, we were able to reduce the number of characters per sample from 722.3 to 15.9. By cross-validating with third-party APIs and heuristic keywords, we infer that about 97% of the identified key tokens are indeed unsafe, indicating that the model is picking up the correct feature most of the time. For the remaining 3% key tokens, we manually reviewed them and discovered the following problems:

1. The frequency of irrelevant symbols is high (e.g., the period symbol contributes to the prediction score).
2. Some key tokens are long and close to the original example, hinting that the model memorizes training samples.
3. Our model is not strong at handling typographical errors and adversarial inputs (e.g., `"w h o r e s"`).

We partially addressed the above issues, and plan to resolve the rest with better data augmentation in the future.

Human Red-teaming. As a final layer of the verification process, we conduct internal red-teaming before releasing new models. The goal of the red-teaming is to uncover any unexpected model weaknesses beyond the test dataset. During the process, we discovered the following issues:

1. The model predicts high `H` scores for examples containing the "#" token. This is likely caused by the fact that we have many `H` training examples from some academic datasets that contain only short tweets.
2. The model predicts high `H` and `V` scores for examples containing potential racial tokens such as `black`. It is expected to classify `"I hate black people!"` as `H` but not `"I hate black cats!"`.

To mitigate the above issues, we construct synthetic datasets from hand-curated templates and synthetic model generations to patch the holes (§3.3), and adjust the training dataset distribution to ensure a right mix across multiple types of text sourced from academic datasets. The process can be iterative, helping us discover new issues and solutions in each round, and naturally leading to improved robustness and consistency in time when the red-teaming process can be executed more regularly and at scale.

4 Experiment Results

4.1 Model Architecture and Training

Our model is a lightweight transformer decoder model where the final output linear layer is replaced with 8 MLP heads, each corresponding to one independent matrix of shape $[d_{\text{model}}, 256, 1]$, where d_{model} is the transformer model size. We find this head architecture works better than a single deep MLP layer with one output vector of 8 dimensions at avoiding interference between categories and requires fewer parameters to train. The model is initialized from a GPT model that is pretrained on a large text corpus and then fine-tuned with learning rate 0.05, batch size 256, dropout rate 0.1 within MLP heads and up to 3 epochs.

4.2 Model Performance

Our model is trained and tested on both production and public data. We are not able to share the test dataset containing production traffic for privacy and legal reasons; hence, we report the model performance on a different test dataset[4] containing only samples from public data, as well as several publicly available datasets on undesired content detection.

Table 3 compares the performance of our model with Perspective API[5] as a baseline on our test dataset, TweetEval (Barbieri et al. 2020), Stormfront hate speech dataset (de Gibert et al. 2018), a subset of Reddit comments with noisy labels on erotic content processed according to Barrientos et al. (2020) and a downsampled Jigsaw toxic comments test dataset (Jigsaw 2018). None of the training portions of external evaluation benchmarks are incorporated into our training, except for half of Jigsaw's training data that has no overlap with the Jigsaw test set in evaluation. Unfortunately, due to the taxonomy mismatch, we cannot have exact comparisons across all categories. For example, our taxonomy does not cover "toxic" and Perspective API does not explicitly detect "self-harm" or "sexual content".

It is not surprising that our model performs the best on the test dataset labeled with the same taxonomy and the Perspective API does a better job on Jigsaw data. It further proves the point on how important it is to align the taxonomy between training data and use cases in evaluation. Our model outperforms the Perspective API baseline on both TweetEval and Stormfront test sets for detecting hateful content, despite the fact that neither are in the training set.

[4] https://github.com/openai/moderation-api-release

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

		Perspective	Ours
Public	S	.8709*	.9703
	H	.6914	**.7968**
	V	.5201	**.7371**
	HR	.3902*	.6191
	SH	-	.8070
	S3	-	.7638
	H2	-	.7268
	V2	-	.6061
Jigsaw	Identity-hate	.6644	**.6890**
	Insult	**.8814**	.8548
	Obscene	.9500	.8353*
	Threat	.7492	.6144*
	Toxic	.9769	.9304*
TweetEval	Hate	.5961	**.6473**
	Offensive	.7919*	.7024*
Stormfront	Hate	.8754	**.9053**
Reddit	Sexual	.8961*	.9417*

Table 3: Comparison of our model with Perspective API on AUPRC (Area under the Precision-Recall Curve) across a set of test datasets. Numbers followed with "*" are based on *approximated* taxonomy matching, not a fair comparison.

4.3 Active Learning Experiments

To assess the importance of active learning, we evaluate the performance of our active learning strategy, as described in §3.1, compared to random sampling.

Iterative training. We repeat the procedure for our active learning strategy and random sampling, respectively:

1. Start with an initial training dataset $\mathcal{D}_0$ of $k_0 = 6000$ labeled examples from public data and a validation set $\mathcal{V}$ of about 5500 samples from the production traffic.
2. for $i \leftarrow 0$ to $N-1$ do ($N = 3$):
   (a) Train a new model M_i on $\mathcal{D}_i$;
   (b) Evaluate M_i on $\mathcal{V}$;
   (c) Score 5×10^5 production samples with M_i;
   (d) Choose about 2000 samples from the above data pool via the selection strategy in test and add samples to the training set to construct $\mathcal{D}_{i+1}$ after labeling.

Results. Table 4 demonstrates the label distributions obtained by the two strategies and our active learning strategy can capture undesired content 10+ times more effectively than random sampling on all categories. Overall about 40% of samples selected by active learning can trigger at least one undesired label, while in comparison only 3.4% of random samples are assigned with any undesired label.

As shown in Fig. 2, using the active learning strategy to decide which new data samples leads to a greater improvement across all categories than random sampling. We observe a significant performance improvement on all categories with active learning after 3 iterations.

Category	Random Sampling	Active Learning	Multiplier
S	1.49%	25.53%	17.1×
H	0.17%	3.09%	18.2×
V	0.48%	9.92%	20.7×
HR	0.55%	6.41%	11.7×
SH	0.09%	1.85%	20.6×
S3	0.24%	2.42%	10.1×
H2	0.03%	0.67%	22.3×
V2	0.25%	4.27%	17.1×
Safe	96.57%	59.54%	-

Table 4: Label distributions for samples selected by random sampling and active learning. Note that one sample may have multiple labels so the total sum may exceed 100%.

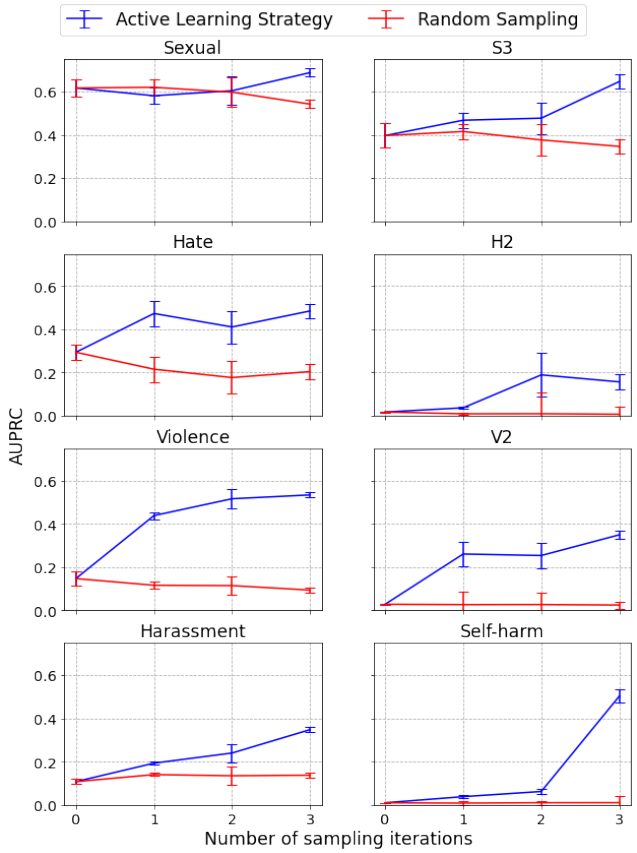

Figure 2: Performance of active learning sampling versus random sampling on the same validation set at each model iteration, measured by AUPRC.

4.4 Domain Adversarial Training Experiments

We sought to understand the effectiveness of WDAT under three scenarios: (1) At the beginning of the project, we only had labeled public data and unlabeled production data. (2) In the middle of the project, we also curated synthetic examples to improve model weaknesses. (3) At a later stage, we had a sufficient amount of labeled production examples. All three

Category	PUB		SYN		MIX	
	Base	DAT	Base	DAT	Base	DAT
S	.698	**.730**	.726	**.745**	**.943**	.939
H	.417	**.491**	**.551**	.476	**.843**	.818
V	.490	**.529**	**.532**	.531	**.640**	.633
HR	.258	**.369**	.326	**.356**	.453	**.482**
SH	.063	**.281**	.086	**.296**	.621	**.632**
S3	.592	**.759**	**.779**	.777	.911	**.936**
H2	.393	**.643**	.570	**.577**	.851	**.854**
V2	.165	**.453**	.093	**.507**	.443	**.533**

Table 5: The average AUPRC on a production validation set. PUB denotes models trained on labeled public datasets, SYN adds additional synthetic examples, and MIX adds additional labeled production examples. We mark the best result within each configuration in bold.

scenarios are important because we want the best model for active learning in every iteration. We use the following setup to compare the performance on our production traffic.

Datasets. We create three training datasets PUB, SYN, and MIX to study (1), (2), and (3), respectively. PUB consists of around 90k public examples including both samples from academic datasets and Web data (Common Crawl) labeled by our annotators. SYN adds additional 69k curated synthetic examples. MIX contains all examples in SYN with additional 60k production samples with labels.

Models. The baselines are trained with basic supervised learning. DAT models are trained with two hidden layers of 300 dimensions using additional 100k unlabeled production data points. All models are trained with up to 2 epochs, and the training is repeated 3 times with different random seeds.

Results. We compare the average AUPRC on the production validation set $\mathcal{V}$. As demonstrated in Table 5, the improvement from WDAT is significant when we only have access to public datasets (PUB), and the marginal gain reduces gradually as we add more training examples, especially in-distribution production samples. For instance, DAT improved SH AUPRC from 0.063 to 0.281 on PUB and from 0.086 to 0.296 on SYN, whereas the improvement is only from 0.621 to 0.632 on MIX. DAT still helps weak categories (SH and V2) on SYN and MIX, but it may slightly hurt the performance for categories with a sufficient amount of in-distribution data such as H and V. We suspect this is because the model failed to find a representation that works very well for both distributions. Further studies on model architectures and training methods are required to improve the performance of all categories throughout the project.

5 Related Work

There is a long track record of work on the definition and detection of hateful, toxic, offensive, and abusive content (Kwok and Wang 2013; Nobata et al. 2016; Rosenthal et al. 2020; Lees et al. 2022; Pavlopoulos et al. 2020). The definitions have overlaps but are not exactly the same, creating obstacles to sharing datasets between projects. Besides offensive and abusive language, there exist additional types of potentially undesired text in the wild, such as sexual content involving minors or support for self-harm or suicide. We observed a gap between current research work and the entirety of content types that should be moderated and detected, and our work aims to fill the gap.

Despite the common belief that training data quality is critical for model performance, there is still a lack of community standards for labeling standards, annotator training, quality metrics, etc. (Vidgen and Derczynski 2020; PAI 2021). For large-scale data collection, crowdsourcing remains the most common approach (Zampieri et al. 2019; Davidson et al. 2017). However, the weak skill set of non-expert annotators can lead to poorer data quality (Waseem 2016; Yin and Zubiaga 2021). Some recent work turns to large pre-trained language models to generate synthetic data (Wang et al. 2021a; Hartvigsen et al. 2022), but it is unclear whether model outputs can adapt to the real-world distribution. It is noteworthy that training data can contain bias due to the subjectivity and biases in the data collection process (Sap et al. 2019).

Active learning has been successfully applied to a number of different domains (Lewis and Gale 1994; Hoi et al. 2006; Schmidt et al. 2020). Several active learning sampling strategies are often used in practice. Uncertainty sampling selects data about which the model is most uncertain, which can be quantified by predicted probabilities (Lewis and Catlett 1994; Scheffer, Decomain, and Wrobel 2001), disagreement among an ensemble of models (Seung, Opper, and Sompolinsky 1992; McCallum and Nigam 1998), or by using dropout and Bayesian approaches (Gal, Islam, and Ghahramani 2017; Siddhant and Lipton 2018). Diversity sampling chooses samples to ensure sufficient diversity within the selection, commonly achieved by clustering unlabeled data and sampling from different clusters (Nguyen and Smeulders 2004), or by selecting representative samples (McCallum and Nigam 1998; Settles and Craven 2008). Uncertainty and diversity sampling can be combined.

Red-teaming is a common approach for model improvement by discovering and patching the weakness iteratively (Dinan et al. 2019; Perez et al. 2022), where humans are encouraged to look for examples that could fail the model. Mishkin et al. (2022) describes an operational process for doing red-teaming with external experts. Ziegler et al. (2022) designed a tool to efficiently assist human adversaries to identify failures in a classifier. Models trained with red-teaming are found to be more robust to adversarial attack (Dinan et al. 2019; Ziegler et al. 2022) and human-in-the-loop dynamic data collection can efficiently improve model performance (Vidgen et al. 2020; Kiela et al. 2021).

Domain adaptation aims at generalizing knowledge learned in one domain towards the other (Ben-David et al. 2006; Weiss, Khoshgoftaar, and Wang 2016; Ben-David et al. 2009). Domain Adversarial Training (DAT) is one of the technique that reduces the domain discrepancy through a learned discriminator to make the representations of source and target indistinguishable (Arjovsky, Chintala, and Bottou 2017; Tzeng et al. 2017; Ganin and Lempitsky 2015). However, DAT suffers from a gradient vanishing problem when the domain discriminator can tell apart the two domains eas-

ily, and Wasserstein distance based methods are proposed to enable a more stable training (Shen et al. 2018; Arjovsky, Chintala, and Bottou 2017; Shah et al. 2018).

6 Conclusion and Future Work

Building high-quality undesired content detection systems in the real world is a challenge that requires the incorporation of multiple methods. A good content taxonomy is the foundation for problem scoping and data collection. A reliable data pipeline is needed to guarantee high data quality and handle distribution shift. When target content occurs rarely, active learning leads to much better model performance. In addition, we argue that good operation processes for labeling are essential for high data quality. And we show that model performance can further be improved via the use of curated synthetic data and semi-supervised learning. In the future, we will continue following related research on social bias to improve the fairness of the model, investigating more data augmentation methods to improve the lexicon robustness and generalizability of the model, and run more rigorous active learning experiments on sophisticated strategies, incorporating both uncertainty and diversity sampling.

References

Alzantot, M.; Sharma, Y.; Elgohary, A.; Ho, B.-J.; Srivastava, M.; and Chang, K.-W. 2018. Generating Natural Language Adversarial Examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2890–2896. Brussels, Belgium: Association for Computational Linguistics.

Anaby-Tavor, A.; Carmeli, B.; Goldbraich, E.; Kantor, A.; Kour, G.; Shlomov, S.; Tepper, N.; and Zwerdling, N. 2020. Do not have enough data? Deep learning to the rescue! In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 7383–7390.

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein Generative Adversarial Networks. In Precup, D.; and Teh, Y. W., eds., *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, 214–223. PMLR.

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. Association for Computational Linguistics.

Barrientos, G. M.; Alaiz-Rodríguez, R.; González-Castro, V.; and Parnell, A. C. 2020. Machine learning techniques for the detection of inappropriate erotic content in text. *International Journal of Computational Intelligence Systems*, 13(1): 591–603.

Ben-David, S.; Blitzer, J.; Crammer, K.; Kulesza, A.; Pereira, F. C.; and Vaughan, J. W. 2009. A theory of learning from different domains. *Machine Learning*, 79: 151–175.

Ben-David, S.; Blitzer, J.; Crammer, K.; and Pereira, F. C. 2006. Analysis of Representations for Domain Adaptation. In *NIPS*.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Davidson, T.; Warmsley, D.; Macy, M.; and Weber, I. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the international AAAI conference on web and social media*, volume 11, 512–515.

de Gibert, O.; Perez, N.; García-Pablos, A.; and Cuadros, M. 2018. Hate Speech Dataset from a White Supremacy Forum. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, 11–20. Brussels, Belgium: Association for Computational Linguistics.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Burstein, J.; Doran, C.; and Solorio, T., eds., *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, 4171–4186. Association for Computational Linguistics.

Dinan, E.; Humeau, S.; Chintagunta, B.; and Weston, J. 2019. Build it break it fix it for dialogue safety: Robustness from adversarial human attack. *arXiv preprint arXiv:1908.06083*.

Dwork, C.; Hardt, M.; Pitassi, T.; Reingold, O.; and Zemel, R. 2012. Fairness through Awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, 214–226. New York, NY, USA: Association for Computing Machinery. ISBN 9781450311151.

Feng, S.; Wallace, E.; Grissom II, A.; Iyyer, M.; Rodriguez, P.; and Boyd-Graber, J. 2018. Pathologies of Neural Models Make Interpretations Difficult. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 3719–3728. Brussels, Belgium: Association for Computational Linguistics.

Gal, Y.; Islam, R.; and Ghahramani, Z. 2017. Deep bayesian active learning with image data. In *International Conference on Machine Learning*, 1183–1192. PMLR.

Ganin, Y.; and Lempitsky, V. S. 2015. Unsupervised Domain Adaptation by Backpropagation. *ArXiv*, abs/1409.7495.

Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; and Lempitsky, V. 2016. Domain-Adversarial Training of Neural Networks. *J. Mach. Learn. Res.*, 17(1): 2096–2030.

Garg, S.; Perot, V.; Limtiaco, N.; Taly, A.; Chi, E. H.; and Beutel, A. 2019. Counterfactual Fairness in Text Classification through Robustness. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '19, 219–226. New York, NY, USA: Association for Computing Machinery. ISBN 9781450363242.

Gehman, S.; Gururangan, S.; Sap, M.; Choi, Y.; and Smith, N. A. 2020. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. *arXiv preprint arXiv:2009.11462*.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and Harnessing Adversarial Examples. *CoRR*, abs/1412.6572.

Hartvigsen, T.; Gabriel, S.; Palangi, H.; Sap, M.; Ray, D.; and Kamar, E. 2022. Toxigen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. *arXiv preprint arXiv:2203.09509*.

Hoi, S. C.; Jin, R.; Zhu, J.; and Lyu, M. R. 2006. Batch mode active learning and its application to medical image classification. In *Proceedings of the 23rd international conference on Machine learning*, 417–424.

Jigsaw. 2017. Perspective API. https://www.perspectiveapi.com/. Accessed: 2022-06-15.

Jigsaw. 2018. Toxic Comment Classification Challenge. https://www.kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge/overview. Accessed: 2022-06-15.

Kiela, D.; Bartolo, M.; Nie, Y.; Kaushik, D.; Geiger, A.; Wu, Z.; Vidgen, B.; Prasad, G.; Singh, A.; Ringshia, P.; Ma, Z.; Thrush, T.; Riedel, S.; Waseem, Z.; Stenetorp, P.; Jia, R.; Bansal, M.; Potts, C.; and Williams, A. 2021. Dynabench: Rethinking Benchmarking in NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4110–4124. Online: Association for Computational Linguistics.

Kumar, V.; Choudhary, A.; and Cho, E. 2020. Data augmentation using pre-trained transformer models. *arXiv preprint arXiv:2003.02245*.

Kusner, M. J.; Loftus, J.; Russell, C.; and Silva, R. 2017. Counterfactual Fairness. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30*, 4066–4076. Curran Associates, Inc.

Kwok, I.; and Wang, Y. 2013. Locate the Hate: Detecting Tweets against Blacks. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, AAAI'13, 1621–1622. AAAI Press.

Lees, A.; Tran, V. Q.; Tay, Y.; Sorensen, J.; Gupta, J.; Metzler, D.; and Vasserman, L. 2022. A new generation of perspective api: Efficient multilingual character-level transformers. *arXiv preprint arXiv:2202.11176*.

Lewis, D. D.; and Catlett, J. 1994. Heterogeneous uncertainty sampling for supervised learning. In *Machine learning proceedings 1994*, 148–156. Elsevier.

Lewis, D. D.; and Gale, W. A. 1994. A sequential algorithm for training text classifiers. In *SIGIR'94*, 3–12. Springer.

McCallum, A.; and Nigam, K. 1998. Employing EM and Pool-Based Active Learning for Text Classification. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, 350–358. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. ISBN 1558605568.

Mishkin, P.; Ahmad, L.; Brundage, M.; Krueger, G.; and Sastry, G. 2022. DALL·E 2 Preview - Risks and Limitations. https://github.com/openai/dalle-2-preview/blob/main/system-card.md. Accessed: 2022-08-04.

Nguyen, H. T.; and Smeulders, A. 2004. Active learning using pre-clustering. In *Proceedings of the twenty-first international conference on Machine learning*.

Nobata, C.; Tetreault, J.; Thomas, A.; Mehdad, Y.; and Chang, Y. 2016. Abusive Language Detection in Online User Content. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, 145–153. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee. ISBN 9781450341431.

PAI. 2021. Responsible Sourcing of Data Enrichment Services. https://partnershiponai.org/paper/responsible-sourcing-considerations/. Accessed: 2022-08-04.

Pavlopoulos, J.; Sorensen, J.; Dixon, L.; Thain, N.; and Androutsopoulos, I. 2020. Toxicity detection: Does context really matter? *arXiv preprint arXiv:2006.00998*.

Perez, E.; Huang, S.; Song, F.; Cai, T.; Ring, R.; Aslanides, J.; Glaese, A.; McAleese, N.; and Irving, G. 2022. Red teaming language models with language models. *arXiv preprint arXiv:2202.03286*.

Reddit. 2022. Building Better Moderator Tools. https://www.reddit.com/r/RedditEng/comments/uly8s4/building_better_moderator_tools/. Accessed: 2022-08-04.

Ribeiro, M. T.; Wu, T.; Guestrin, C.; and Singh, S. 2020. Beyond Accuracy: Behavioral Testing of NLP Models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 4902–4912. Online: Association for Computational Linguistics.

Rosenthal, S.; Atanasova, P.; Karadzhov, G.; Zampieri, M.; and Nakov, P. 2020. A Large-Scale Semi-Supervised Dataset for Offensive Language Identification. *arXiv preprint arXiv:2004.14454*.

Röttger, P.; Vidgen, B.; Nguyen, D.; Waseem, Z.; Margetts, H.; and Pierrehumbert, J. 2021. HateCheck: Functional Tests for Hate Speech Detection Models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 41–58. Online: Association for Computational Linguistics.

Sap, M.; Card, D.; Gabriel, S.; Choi, Y.; and Smith, N. A. 2019. The Risk of Racial Bias in Hate Speech Detection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 1668–1678. Florence, Italy: Association for Computational Linguistics.

Scheffer, T.; Decomain, C.; and Wrobel, S. 2001. Active hidden markov models for information extraction. In *International Symposium on Intelligent Data Analysis*, 309–318. Springer.

Schick, T.; and Schütze, H. 2021. Generating datasets with pretrained language models. *arXiv preprint arXiv:2104.07540*.

Schmidt, S.; Rao, Q.; Tatsch, J.; and Knoll, A. 2020. Advanced active learning strategies for object detection. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, 871–876. IEEE.

Settles, B.; and Craven, M. 2008. An analysis of active learning strategies for sequence labeling tasks. In *proceedings of the 2008 conference on empirical methods in natural language processing*, 1070–1079.

Seung, H. S.; Opper, M.; and Sompolinsky, H. 1992. Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, 287–294.

Shah, D.; Lei, T.; Moschitti, A.; Romeo, S.; and Nakov, P. 2018. Adversarial Domain Adaptation for Duplicate Question Detection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 1056–1063. Brussels, Belgium: Association for Computational Linguistics.

Shen, J.; Qu, Y.; Zhang, W.; and Yu, Y. 2018. Wasserstein Distance Guided Representation Learning for Domain Adaptation. In *AAAI*.

Siddhant, A.; and Lipton, Z. C. 2018. Deep Bayesian Active Learning for Natural Language Processing: Results of a Large-Scale Empirical Study. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2904–2909. Brussels, Belgium: Association for Computational Linguistics.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I. J.; and Fergus, R. 2013. Intriguing properties of neural networks. *CoRR*, abs/1312.6199.

Thoppilan, R.; Freitas, D. D.; Hall, J.; Shazeer, N.; Kulshreshtha, A.; Cheng, H.-T.; Jin, A.; Bos, T.; Baker, L.; Du, Y.; Li, Y.; Lee, H.; Zheng, H. S.; Ghafouri, A.; Menegali, M.; Huang, Y.; Krikun, M.; Lepikhin, D.; Qin, J.; Chen, D.; Xu, Y.; Chen, Z.; Roberts, A.; Bosma, M.; Zhao, V.; Zhou, Y.; Chang, C.-C.; Krivokon, I.; Rusch, W.; Pickett, M.; Srinivasan, P.; Man, L.; Meier-Hellstern, K.; Morris, M. R.; Doshi, T.; Santos, R. D.; Duke, T.; Soraker, J.; Zevenbergen, B.; Prabhakaran, V.; Diaz, M.; Hutchinson, B.; Olson, K.; Molina, A.; Hoffman-John, E.; Lee, J.; Aroyo, L.; Rajakumar, R.; Butryna, A.; Lamm, M.; Kuzmina, V.; Fenton, J.; Cohen, A.; Bernstein, R.; Kurzweil, R.; Aguera-Arcas, B.; Cui, C.; Croak, M.; Chi, E.; and Le, Q. 2022. LaMDA: Language Models for Dialog Applications. arXiv:2201.08239.

Tzeng, E.; Hoffman, J.; Saenko, K.; and Darrell, T. 2017. Adversarial Discriminative Domain Adaptation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2962–2971.

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. 2020. Learning from the worst: Dynamically generated datasets to improve online hate detection. *arXiv preprint arXiv:2012.15761*.

Wang, S.; Liu, Y.; Xu, Y.; Zhu, C.; and Zeng, M. 2021a. Want To Reduce Labeling Cost? GPT-3 Can Help. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, 4195–4205. Punta Cana, Dominican Republic: Association for Computational Linguistics.

Wang, Z.; Yu, A. W.; Firat, O.; and Cao, Y. 2021b. Towards zero-label language learning. *arXiv preprint arXiv:2109.09193*.

Waseem, Z. 2016. Are You a Racist or Am I Seeing Things? Annotator Influence on Hate Speech Detection on Twitter. In *Proceedings of the First Workshop on NLP and Computational Social Science*, 138–142. Austin, Texas: Association for Computational Linguistics.

Weidinger, L.; Mellor, J.; Rauh, M.; Griffin, C.; Uesato, J.; Huang, P.-S.; Cheng, M.; Glaese, M.; Balle, B.; Kasirzadeh, A.; et al. 2021. Ethical and social risks of harm from language models. *arXiv preprint arXiv:2112.04359*.

Weiss, K. R.; Khoshgoftaar, T. M.; and Wang, D. 2016. A survey of transfer learning. *Journal of Big Data*, 3: 1–40.

Yin, W.; and Zubiaga, A. 2021. Towards generalisable hate speech detection: a review on obstacles and solutions. *PeerJ Computer Science*, 7: e598.

Yoo, K. M.; Park, D.; Kang, J.; Lee, S.-W.; and Park, W. 2021. GPT3Mix: Leveraging Large-scale Language Models for Text Augmentation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2225–2239. Punta Cana, Dominican Republic: Association for Computational Linguistics.

YouTube. 2019. The Four Rs of Responsibility, Part 1: Removing Harmful Content. https://blog.youtube/inside-youtube/the-four-rs-of-responsibility-remove/. Accessed: 2022-08-04.

Zampieri, M.; Malmasi, S.; Nakov, P.; Rosenthal, S.; Farra, N.; and Kumar, R. 2019. Predicting the Type and Target of Offensive Posts in Social Media. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 1415–1420. Association for Computational Linguistics.

Zhang, C.; Zhao, J.; Zhang, H.; Chang, K.-W.; and Hsieh, C.-J. 2021. Double Perturbation: On the Robustness of Robustness and Counterfactual Bias Evaluation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 3899–3916. Online: Association for Computational Linguistics.

Zhao, J.; Wang, T.; Yatskar, M.; Ordonez, V.; and Chang, K.-W. 2017. Men Also Like Shopping: Reducing Gender Bias Amplification using Corpus-level Constraints. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2979–2989. Copenhagen, Denmark: Association for Computational Linguistics.

Ziegler, D. M.; Nix, S.; Chan, L.; Bauman, T.; Schmidt-Nielsen, P.; Lin, T.; Scherlis, A.; Nabeshima, N.; Weinstein-Raun, B.; de Haas, D.; et al. 2022. Adversarial Training for High-Stakes Reliability. *arXiv preprint arXiv:2205.01663*.