Reducing Non-Normative Text Generation from Language Models

Xiangyu Peng*, **Siyan Li***, **Spencer Frazier, and Mark Riedl**
Georgia Institute of Technology
Atlanta, GA 30332
{xpeng62, sli613, sfrazier7, riedl}@gatech.edu

Abstract

Large-scale, transformer-based language models such as GPT-2 are pretrained on diverse corpora scraped from the internet. Consequently, they are prone to generating non-normative text (i.e. in violation of social norms). We introduce a technique for fine-tuning GPT-2, using a policy gradient reinforcement learning technique and a normative text classifier to produce reward and punishment values. We evaluate our technique on five data sets using automated and human participant experiments. The normative text classifier is 81-90% accurate when compared to gold-standard human judgements of normative and non-normative generated text. Our normative fine-tuning technique is able to reduce non-normative text by 27-61%, depending on the data set.

1 Introduction

Human societies implicitly establish codes of acceptable behavior in social contexts. *Normativity* is behavior that conforms to expected societal norms and contracts, whereas non-normative behavior aligns to values that deviate from these expected norms. Sumner (1967) defines norms as: "...informal rules that are not written, but, when violated, result in severe punishments and social sanction upon the individuals, such as social and religious exclusions." Non-normativity does not connote behavior devoid of value or immoral, but behavior that fails to conform to social standards shared by other individuals in the relevant group, organization or society. Norms can also be thought of as actions taken by an entity which conform to an identity (Katzenstein, 1996), thus allowing others to categorize behavior as in-group or out-group. Different societies and groups collectively have different ideals about what actions constitute normative behavior; group members use these ideals to heuristically guide their actions to avoid social ostracization. For example, many societies have norms against violence, or certain behaviors being conducted in public. Conflicts between individuals can arise when enacting non-normative behaviors or uttering non-normative speech.

This paper examines generative language models and the frequency at which they generate descriptions of non-normative behavior. Large-scale, transformer-based neural language models such as ELMo (Peters et al., 2018), BERT (Devlin et al., 2018), GPT-2 (Radford et al., 2019), GPT-3, Grover (Zellers et al., 2019), CTRL (Keskar et al., 2019), T5 (Raffel et al., 2019), and XLNet (Yang et al., 2019) are trained on very large corpora such as text scraped from the internet, books, or both.

These language models generate text that is statistically representative of the corpora they were trained on. As such, text scraped from the internet co-mingles text produced by many groups with differing norms, as well as text produced by people intentionally using non-normative speech, like "trolling" language. Models trained on these data can then produce undesirable, harmful output. Stories from the internet and books also contain normative and non-normative situations (e.g., antagonists, as well as protagonists conducting conventionally non-normative behaviors). Consequently, it is possible, and often likely, for language models to generate non-normative descriptions of behavior (murder, crime, suicide, racist actions, rude behavior, etc.), exhibit biases against certain demographics groups (Sheng et al., 2019; Solaiman et al., 2019), stereotypical biases (Nadeem et al., 2020) or racist text when prompted with trigger phrases (Wallace et al., 2019).

Value alignment (Russell et al., 2015) is the concept that an agent is unable to perform actions that cause harm to humans. Harmful behavior is not limited to physical actions by robots, the focus

*Equal contributions

of some AI value alignment research. We recognize that natural language communication can also cause harm. For example, Amazon Alexa, a virtual assistant AI, was reported to suggest a user commit suicide.[1] Frazier et al. (2019) developed a classifier for normative behavior which exhibits strong zero-shot and few-shot transfer across a variety of text corpora. The authors speculate that their model—which they call a *value-aligned prior*—can bias model output perceived as more normative. In this paper we ask a different question: whether a value-aligned prior can be used to reduce the generation of descriptions of non-normative behavior by neural language models.

The common approach to fine-tuning language models is to provide additional corpora of exemplars. If a corpus of exemplars is normative, the language model can be trained to emulate this over time. Generally, in the absence of very large normative corpora, we need an alternative approach to fine-tuning language models. We use a reinforcement learning approach to fine-tuning language models, using the normative behavior classifier of Frazier et al. (2019) as a non-differentiable reward function. Our method back-propagates reward relative to the degree of non-normativity of text generated by the language model.

We evaluate our reinforcement learning fine-tuning technique with three sets of experiments. First, we replicate the experiments by Frazier et al. (2019) on text generated by a language model instead of originally held-out corpus text. Second, we show with automated and human participant experiments that fine-tuning on reward generated by a normative classifier model can reduce the generation of non-normative text by $27 - 61\%$. Third, we ablate our technique and show with automated and human participant experiments that the fine-tuning technique works with classifiers other than the normative classifier—specifically models trained to classify negative-sentiment and toxic language.

2 Background and Related Work

2.1 Value Alignment and Normative Priors

Humans have expectations that — just like other humans — agents will avoid harmful actions, conform to personal values and to social norms (Bicchieri, 2005), even when not explicitly communicated. This is referred to as the *value alignment problem* (Soares and Fallenstein, 2014; Russell et al., 2015; Arnold et al., 2017; Abel et al., 2016). Harmful agent behavior can theoretically be mitigated by casting values as preferences over action sequences. For example Christiano et al. (2017) collected human preferences to shape rewards for game-playing agents in reinforcement learning.

Instead of preference learning, Frazier et al. (2019) used the BERT (Devlin et al., 2018) language model's token embeddings to train a binary classifier. This model is used to differentiate between normative and non-normative natural language sentences containing events, utterances and descriptions of behavior. They obtained training data from *Goofus & Gallant (G&G)*, a children's educational comic strip featuring two characters of the same names. Goofus always deviates from the "proper" way to behave, while Gallant always performs the behavior of an exemplary child in western society at the time the comics were created. As a result, *G&G* is a naturally labeled source of normative and non-normative text, for the specific society it represents.

Frazier et al. (2019) demonstrated this method could accurately classify descriptions of behavior as normative or non-normative. Furthermore, this classifier retained high performance in zero-shot and few-shot transfer tasks. For example, they show that their classifier, trained on *G&G* comics, can classify normative event descriptions in contemporary collections of popular plot points and science fiction plot summaries, instances of medium- and far-transfer, respectively. The authors speculate that their classifier model can bias agent behavior toward normative courses of action in other contexts. However, this was not directly shown. We ask whether a normative classifier can be used to fine-tune the "behavior" of a large-scale transformer-based language model.

2.2 Language Model Training & Fine-Tuning

Large-scale transformer-based neural language models such as BERT and GPT-2 are trained on large corpora of text scraped from the web and books. They can be fine-tuned to a specific domain of interest, commonly accomplished by providing a corpus of exemplars from that domain. Over time, the weights of the pre-trained model will shift and increasingly generate passages which better emulate the corpus of exemplars. If the fine-tuning corpus of exemplars is normative, the language model

[1] https://www.newsweek.com/amazon-echo%2Dtells-uk-woman-stab-herself-1479074/

will, in theory, learn to prefer normative language over time. *Goofus & Gallant* is one such normative corpus, and if a language model is fine-tuned on it then it may prefer to generate normative language.

GPT-2 (Radford et al., 2019), in particular, is a large-scale transformer-based language model trained on a large corpus of text scraped from web pages and social media. Applying the concept of value alignment as preference learning, Ziegler et al. (2019) use a reinforcement learning method on the 774M-parameter version of GPT-2 to favor human-preferred text. Crowd workers were asked to select generated text completions from a set of given prompts that had positive sentiment. These preference values were used to fine-tune GPT-2. This is one possible technique for reinforcement-based fine-tuning; sentiment is, however, not necessarily a good measure of adherence to norms. We replace the linear reward model for fine-tuning GPT-2 (Ziegler et al., 2019) with a pre-trained normative text classifier.

The Plug and Play Language Models (PPLM) (Dathathri et al., 2019) also apply attribute classifiers to fine-tune language models; the technique is demonstrated via generating text with a target sentiment and also decreasing the frequency of toxic language. There are two limitations: (a) PPLM trains a model to operate on a *fixed* set of prefix input, and (b) the classification must be done on a word-by-word basis and thus cannot easily be applied to problems where the normative valence of individual words relies on a single or multiple sentence context (e.g. quoting and admonishing toxic speech). Our fine-tuning technique, in contrast, works on arbitrary prefixes and assesses the the normativity of entire sentences.

2.3 Datasets

We make use of five datasets, chosen to represent a diverse set of domains. The normative text classifier by Frazier et al. (2019) was tested on a corpus of science fiction plot summaries (Ammanabrolu et al., 2019) as well as a new *Plotto* dataset, based on a book by the same name that catalogues plot points for scaffolding fictional story-writing. Story corpora are particularly good for testing problems pertaining to textual descriptions of normative and non-normative behavior. Stories contain antagonists that frequently violate societal norms and protagonists who are more likely to exemplify contemporary social norms. We recognize many stories require protagonists to perform non-normative behaviors like violence against others to achieve normative ends, further indicating the importance of accounting for a broader frame of context when determining normativity.

The science-fiction plot summary corpus (Ammanabrolu et al., 2019) is a collection of 2,276 stories scraped from crowd-sourced plot summaries on fan sites. These stories have an average length of 89.23 sentences. Sentences in this corpus tend to give high-level overviews of the actions that characters are performing (e.g. "Lyta accuses Sinclair of attempting to murder the ambassador"). The sci-fi corpus also presents a transfer challenge because it involves a lot of novel entities—aliens, spaceships, laser weapons, etc.—that do not exist in *Goofus & Gallant*. It is notable that a normative text classifier trained on *G&G* would do well on zero-shot transfer to the sci-fi corpus. This makes it an attractive dataset for our experiments for the same reasons.

The *Plotto* dataset consists of 900 sentences extracted from a book, which catalogues plot points used in popular fiction. Frazier et al. (2019) pruned some exceptionally anachronistic and misogynistic sentences from the corpora. These sentences approximate the level of abstraction in the sci-fi corpus but have more contemporary narratives.

The *ROCstories* (Mostafazadeh et al., 2016) corpus contains 52,666 five-sentence stories, often about everyday life situations (e.g. going for a jog, taking a test in school, etc.). Unlike the previous two corpora, it covers a different space of more common, mundane events which usually do not have strong normativity connotations.

Sentiment is often used as a surrogate for normativity under the belief that non-normative behavior would be associated with negative sentiment. The relationship between normativity and sentiment is not that simple, as we will show in Section 4.3. We include sentiment experiments using large review datasets from IMDb, Yelp, and Amazon (Kotzias et al., 2015) because (1) previous value alignment research has incorporated sentiment analysis, and (2) we want to test our techniques on classifiers other than the normative text classifier.

Non-normativity is a superset of toxic language in the sense that toxic language is non-normative, but not all non-normative descriptions are toxic. We also conduct experiments using toxic language classifiers - fine-tuned on sentiment corpora like the dataset from the Toxic Comment Classification

Challenge[2] as an alternative to the normative text classifier fine-tuned on *G&G*.

3 Normative Fine-Tuning

The GPT-2 model is trained by minimizing its cross-entropy loss given by (Radford et al., 2019):

$$loss_{\mathrm{w}}(X, y) = -\log\left(\frac{\exp(X[y])}{\sum_{i \in V}\exp(X[i])}\right) = -\log\left(\sigma(X)_y\right) \quad (1)$$

where X is a vector containing output logits and y is the index of the word from the ground truth in X. V is the model's vocabulary, σ is the softmax function, and $\sigma(X)_y$ is the ground truth probability of the word.

To punish GPT-2 for producing non-normative text, we use a normative text classifier to evaluate the model's performance and produce a reward value, which is applied to the loss and back-propagated through GPT-2. Given that the normativity of a sentence can only be determined by reading the entire sentence, the classifier must therefore produce a single numeric value per sentence. Specifically, we augment the cross-entropy loss computation with predictions from the pre-trained classifier. We define the sentence loss as:

$$loss_{\mathrm{s}}(s) = \frac{1}{n}\sum_{j \in s}\mathrm{loss}_{\mathrm{w}}(X_j, y_j) + u(s) \quad (2)$$

where s is the continuation sentence generated by the neural language model, $n = |s| - 1$ is the number of the words in the continuation sentence, X_j is the j^{th} logit vector, and y_j is the ground truth index for the j^{th} word. $u(s)$ is a function of the output of the classifier converted into a *punishment* value; a value of zero indicates no punishment, and higher positive values indicating increasingly non-normative sentences. The $loss_{\mathrm{w}}$ counter-balances the reward and prevents the generated texts from descending into incoherent fragments.

The punishment function $u(s)$ generates a value proportional to the average word loss so that it does not become overwhelmed by $loss_{\mathrm{w}}$:

$$u(s) = \rho\beta(1 - C(s))\left(\frac{1}{n}\sum_{j \in s} loss_{\mathrm{w}}(X_j, y_j)\right) \quad (3)$$

where s is a continuation sentence, $C(s)$ is the binary $\{0, 1\}$ label given by the normative classifier,

[2] https://www.kaggle.com/c/jigsaw%2Dtoxic-comment-classification-challenge/

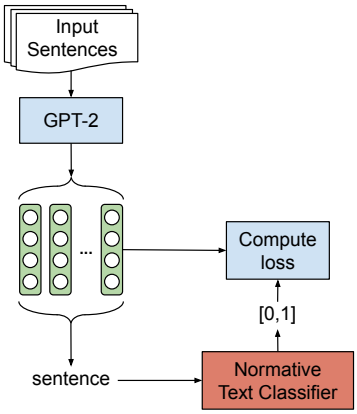

Figure 1: Pipeline for fine-tuning GPT-2 with the classifier. Loss is backpropagated through the output logits to GPT-2.

ρ is a hyper-parameter to control the strength of the penalty, and $\beta = (1 - i \times 0.05)$ decreases the penalty as the number of fine-tuning iterations i increases. That is, if the generated sentence is classified as normative, a loss close to zero will be applied to each logit generated by the language model. If the generated sentence is non-normative, a higher total sentence loss will be applied to each logit. β decreases the step size during back-propagation to avoid over-shooting the local minima. $loss_{\mathrm{w}}$ acts as a cycle loss component, punishing the sentences with undesirable characteristics.

The fine-tuning process is as follows: given a set of input sentences from a corpus, GPT-2 is used to generate successor sentences. We generate 60 tokens and truncate at the first punctuation mark (e.g. periods, question marks). These continuation sentences are fed through a classifier, which outputs the binary label we treat as a reward $C(s) \in \{0, 1\}$. Sentences labeled as 0 are those with undesirable characteristic. As per Equation (3), the reward is used to calculate the punishment score by subtracting from 1.0 and scaling by the average word loss of the sentence. This value is then used to compute a sentence loss as in Equation (2). The sentence loss is averaged to obtain the token-level loss, which is then added to each logit from the continuation sentence and the loss is back-propagated into GPT-2. The process is illustrated in Figure 1.

To prevent the model from deviating too much from the language in the original dataset, we feed the fine-tuned model with the same set of input sentences at every loop and use the output sentences to even further fine-tune the model. As the model is trained, the output sentences will differ, and the

Dataset	Accuracy (continuations)	Accuracy (test corpora)	# of sent.
Plotto	81.25	89.67	100
Sci-fi	82.11	87.51	300
ROCstories	90.57	94.56	100
Toxic	86.84	94.27	400
Sentiment	88.14	93.90	200

Table 1: Results of Mechanical Turk study. Accuracy on generated continuations equals to the percentage of Mechanical Turk worker labels equivalent to labels produced by the normative classifier when classifying generated sentences, since Mechanical Turk worker labels are considered as ground truth label of generated continuations. Accuracy on original corpora is measured by the classifier on the held-out test sets of corpora sentences.

reward value $C(s)$ may change after every iteration as the model shifts its distribution.

4 Experiments

We conduct three sets of experiments to (1) verify the normative text classifier on generated continuations, (2) evaluate our reward-based fine-tuning with the normative text classifier, (3) evaluate our reward-based fine-tuning on other classifiers.

4.1 Experiment 1: Replication of the Normative Classifier

The normative text classifier by Frazier et al. (2019) was evaluated on original sentences from a number of corpora, including the science fiction story corpus (sci-fi) we use in subsequent evaluation experiments. Generated text potentially constitutes a shift in the text distribution. Hence, the accuracy of classifiers on generated continuations must be validated.

The 117M parameter GPT-2 is fine-tuned with training sets from Plotto, ROCstories, sci-fi, Toxic and Sentiment datasets, separately, in order to shift the output probability distribution of GPT-2 and to make it generate text similar to the corpus we used for training. The sci-fi and Plotto corpora were used for fine-tuning two different versions of the normative text classifier, starting with the classifier by Frazier et al. (2019). Thus the original classifier originally trained on *G&G* was updated to the respective domains; a few-shot transfer paradigm. We fine-tuned the classifiers for 2-5 iterations.

For the ROCstories, Toxic and Sentiment datasets, we directly train a BERT-based classifier on the given labels instead of fine-tuning the classifier that was first trained on *G&G*. We found the *G&G*-trained classifier did not transfer well to ROCstories and thus collected our own normative and non-normative labels. Toxic and Sentiment experiments do not look at normativity so we did not use the normative classifier.

A human participant study was then conducted to validate the fine-tuned classifier's accuracy. Sentences from each corpus test set were randomly chosen and used as prompts for GPT-2 to generate continuation sentences. 70 crowd workers on Mechanical Turk labeled those generated sentences as normative or non-normative (or positive or negative sentiment, or toxic or non-toxic). Each sentence received at least three labels. We treat the majority label from humans participants as the ground-truth.

Table 1 shows the accuracy of classifiers on generated continuations and on sentences directly from the test sets. Accuracy decreases on generated continuations, but are on par with accuracy on sentences taken directly from the test corpora, and on par with the results from Frazier et al. (2019). This indicates that any distributional shift during generation is likely inconsequential and the classifer achieves good zero-shot transfer to more datasets.

4.2 Experiment 2: Decreasing Non-Normative Generation

In this set of experiments, we seek to determine if, and by how much, the amount of non-normative behavior descriptions generated by GPT-2 decreases when fine-tuned with the normative text classifier. We emphasis the decrease of non-normative language because both normative and neutral languages are acceptable.

Consistent with Experiment #1, we first fine-tune the 117M parameter version of GPT-2 with sentences sampled from three datasets: ROCstories, sci-fi, and Plotto. This gives us three versions of GPT-2: *GPT-ROCstories*, *GPT-scifi* and *GPT-plotto*, respectively. The 117M GPT-2 model is fine-tuned for two, three and five iterations separately on ROCstories, Plotto and sci-fi to avoid overfitting. We then fine-tune each of these models a second time using the reinforcement learning, reward-based technique in Section 3. We refer to these models as *GPT-ROCstories-norm*, *GPT-sci-fi-norm*, and *GPT-Plotto-norm*, respectively. Due to the small size of the datasets, GPT-2 easily overfits during training. Therefore, we only fine-tune one of its 12 attention heads to avoid overfitting.

We evaluate the performance of our fine-tuned

Model	% non-norm. Auto	Human	Perpl.	Test size
GPT-ROCstories	64.15	58.49	81.097	50
GPT-ROCstories-norm	26.42	22.64	82.958	50
GPT-Plotto	81.25	72.91	34.271	50
GPT-Plotto-norm	59.18	53.06	32.322	50
GPT-scifi	35.11	26.58	23.885	300
GPT-scifi-norm	15.79	18.27	24.522	300

Table 2: The proportion of non-normative behavior and events (% Non-norm) generated by different fine-tuned models on different datasets. Ratios are measured using the normative text classifier (automated) and Mechanical Turk studies (human labeling).

Label	Sentence
Non-norm.	Mollari now refuses to pay the two parents' expenses and lives.
Non-norm.	Garibaldi slaps the door behind them and locks it behind them.
Non-norm.	He considers himself morally superior to his family because he is wealthy.
Norm.	Nathaniel repays his debt through an honest act of honest enterprise.
Norm.	He then makes a generous and appropriate sacrifice.
Norm.	He returns home to support his country.

Table 3: Examples of generated normative and non-normative sentences from *GPT-Plotto* and *GPT-scifi*.

GPT-X-norm models (X=*plotto*, *scifi*, *ROCstories*) by analyzing the change in the proportion of generated text that is non-normative. We measure the ratio of non-normative to normative text in two ways. First, we use the normative text classifier on continuations generated by baselines and fine-tuned models. This is an automated evaluation; Experiment 1 suggests the normative text classifier has high accuracy on the continuations. However, the gold standard is the human participant labels. For our second evaluation metric, we hired 70 Mechanical Turk workers to label generated continuation sentences as normative (including neutral) or non-normative. At least 3 crowd workers labeled each sentence and the majority vote is considered as the ground truth label.

Table 2 shows the proportions of non-normative sentence continuations for both the baseline and the fine-tuned models. The results are summarized below. We note percentage decreases, which are the relative percentage decrease compared to the original statistics.[3]

- *GPT-ROCstories* generates non-normative continuations 64% of the time according to the normative text classifier, which reduces to 26% after further fine-tuning, a 59% decrease. Humans label *GPT-ROCstories* continuations as non-normative 58% of the time, which drops to 22%, a 61% decrease.

- *GPT-Plotto* generates non-normative continuations 81% of the time according to the normative text classifier, which reduces to 59% after further fine-tuning, a 27% decrease. Humans label *GPT-Plotto* continuations as non-normative 72% of the time, which drops to 53%, also a 27% decrease.

- *GPT-scifi* generates non-normative continuations 35% of the time according to the normative text classifier, which reduces to 15% after further fine-tuning, a 55% decrease. Humans label *GPT-scifi* continuations as non-normative 26% of the time, which drops to 18%, a 31% decrease.

We observe that the classifier results are generally in line with the human evaluation results. The models fine-tuned on the Plotto dataset generally generate more non-normative continuation sentences and are more difficult to induce normativity. The models fine-tuned on the sci-fi dataset have the lowest frequency of non-normative generations, but can still be induced to produce lower frequencies with this method.

The perplexity remains steady after fine-tuning with the normative text classifier, indicating that the *GPT-X-norm* models are not overfitting nor losing their fluency. Table 3 shows some examples of sentences generated by the fine-tuned GPT-2 models for the sci-fi and Plotto domains.

4.3 Experiment 3: Other Classifiers

In the previous sections we evaluate how the normative text classifier and our fine-tuning technique work together to decrease the generation of non-normative text. In this section, we ablate our system and evaluate our fine-tuning method independently of the normative text classifier. We seek to determine whether the fine-tuning technique is general enough to work with other classifiers.

We replicate the experimental methodolgy in Section 4.2 but with the Toxic Comment Classification dataset and the Sentiment dataset. Sentiment is often used as a surrogate for normativity because

[3] Percentage decrease is calculated by $(p - \hat{p})/p$, where p and $\hat{p}$ are the proportion of non-normative behavior (% Non-norm) generated by *GPT-X* and *GPT-X-norm*, respectively.

Model	% Non-norm Auto	Human	perpl.	Test size
GPT-toxic-ext	59.03	57.01	54.819	200
GPT-toxic-ext-norm	27.75	30.70	62.302	200
GPT-senti-ext	71.13	76.29	83.717	100
GPT-senti-ext-norm	45.36	42.27	83.443	100
GPT-toxic-bal	37.89	33.04	50.535	200
GPT-toxic-bal-norm	24.79	28.76	60.100	200
GPT-senti-bal	44.33	36.08	91.927	100
GPT-senti-bal-norm	35.05	29.90	90.261	100

Table 4: The proportion of non-normative behavior and events (% non-norm.) generated by different fine-tuned models on different datasets.

non-normative behavior might be inferred to be perceived with negative sentiment, and because labeled sentiment data is more readily available. Toxic language is a subclass of non-normative behavior.

First, we train two classifiers using the same technique as Frazier et al. (2019). Specifically, we fine-tune BERT on the datasets with ground-truth sentiment and toxicity labels. Both classifiers are fine-tuned 5 times on training set of corpora and tested on test sets (see Table 1).

We follow the methodology in Section 4.2 to produce new GPT-2 baseline models. To make the improvement from applying our technique more visible, we fine-tuned the 117M GPT-2 with only toxic or negative sentences when producing the baseline models, and obtained *GPT-senti-ext* and *GPT-toxic-ext*, which frequently produce negative or toxic generated continuations. We also fine-tuned the 117M GPT-2 with balanced datasets (half negative texts and half toxic texts, respectively), and refer to these two models as *GPT-senti-bal* and *GPT-toxic-bal*. We then further fine-tune these models using their respective classifiers per the technique in Section 4.2.

Table 4 shows the percentage of textual continuations that are either toxic or contain negative sentiment. Fine-tuning *GPT-senti-ext* with the sentiment classifier can reduce negative sentiment from 76% to 42%, a 45% reduction. Fine-tuning *GPT-senti-bal* with the sentiment classifier can reduce negative sentiment from 36% to 29%, a 17% reduction. Fine-tuning *GPT-toxic-ext* with the toxic classifier can reduce toxic language from 57% to 30%, a 46% reduction. Fine-tuning *GPT-toxic-bal* with the toxic classifier can reduce toxic language from 33% to 28%, a 13% reduction. This shows that the fine-tuning technique working on sentence loss is agnostic to which classifier is used.

We did not compare our results directly to the Plug and Play Langauge Models (PPLM) work (Dathathri et al., 2019), which also uses a toxic word classifier to fine-tune a language model. PPLM fine-tunes on a word-by-word basis instead of at the sentence unit, making it difficult to account for the context needed for determining normativity. For toxic language reduction, their model is trained to operate on a pre-given set of prompts such as "black" or "asian". For these prompts, given during training, they can reduce GPT-2's toxic language frequency from ∼10% to ∼6%. This is non-significant ($p < 0.23$) though it is challenging to reduce a number that is already close to zero. When we prompt our *GPT-toxic-bal* and *GPT-toxic-bal-norm* with "asian" and "black", we see reductions from 52% to 32% and from 70% to 50%, respectively. Our classifier has only ever seen the word "black" once and has never seen the word "asian". We see higher occurrences of toxicity because GPT-2 is fine-tuned on the equal numbers of toxic and non-toxic sentences from the Toxic dataset (whereas PPLM compares against a non-fine-tuned version of GPT-2) and because GPT-2 has a pre-existing unjust bias toward these words.

To address the relationship between sentiment and normativity, we sample 300 sentences from the sci-fi corpus and 300 sentences from the generation results of the trained GPT-2 model and classify them using the normative text classifier and SentiWordNet (Esuli and Sebastiani, 2006). Figure 2 shows the percentage of sentences were classified as (a) both normative and positive/neutral sentiment (orange), (b) both non-normative and negative sentiment (blue), (c) normative but negative sentiment (green), and (d) non-normative but positive/neutral sentiment (brown). Only about half the sentences tested (53.08%) had sentiment and normativity labels that matched, whereas 46.92% of sentences have conflicting labels.

5 Discussion

We demonstrate how value-aligned priors such as normative text classifiers can act as a reward provider to nudge the GPT-2 language model towards producing more normative and neutral descriptions of behaviors and events. Applying this reward-based fine-tuning technique reduces the likelihood of generating sentences containing non-normative behavior by approximately ∼27-61%, depending on the dataset. Some datasets

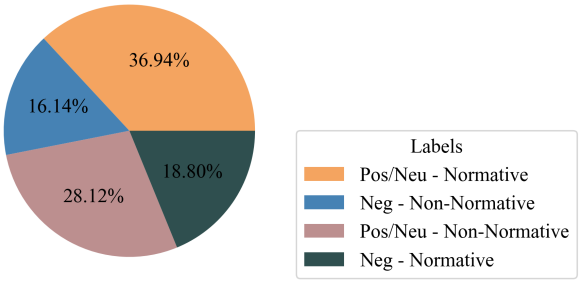

Figure 2: Differences between normative classification and sentiment classification.

are more non-normative and more resistant to reduction of non-normativity. Datasets with low non-normativity to begin with are, naturally, more resistant.

Beyond the numerical results, this shows evidence that policy-gradient based reinforcement learning approaches to fine-tuning can be an effective means for reducing the generation of non-normative descriptions. By using a normative text classifier—a *value aligned prior*—one can fine-tune a language model to the desired domain and then fine-tune using the prior again to reduce non-normative generation that stems either from GPT-2 or from the domain corpus. We provide this approach as an alternative—or in complement to—debiasing techniques that attempt to correct for prejudicial bias in datasets prior to training. Our approach is roughly equivalent to teaching a language model to censor itself.

The policy-gradient based reinforcement learning technique using a value-aligned prior can be even more valuable with *GPT-3*, where fine-tuning to a domain is less necessary. GPT-3 has been demonstrated to be capable of non-normative, toxic, and prejudicially biased generation.

By using the normative text classifier trained on *Goofus & Gallant* comics, our results are limited to Western—and in particular American—mainstream ideals of normative behavior. We acknowledge that culture is not monolithic, even within the United States of America, and this represents only one of many possible sources of normativity. We cannot conclusively say that our results will hold if we had normative text classifiers trained from different source materials. Because general sources of normative behavior are currently hard to come by, we attempt to show generalization of our technique with experiments using sentiment and toxic language.

One limitation of our work is that the normative text classifier can only classify individual sentences without context. Given the context-dependent nature of normativity, the normative text classifier may overlook non-normative sentences that may appear normative out-of-context. This may lead to GPT-2 still producing this sentence in its non-normative context. Another limitation is that fine-tuning GPT-2 on Plotto, ROCstories and sci-fi datasets leads to it generating both neutral and normative sentences. If a model that generates solely normative sentences is desired, one can substitute the normative classifier with a ternary classifier with labels for normative, non-normative, and neutral sentences, and adjust the reward signals accordingly. Furthermore, fine-tuning classifiers requires datasets with labeled exemplars, hence, in order to replicate our work to generate texts with some other desirable characteristics, datasets with labels would be prerequisites.

The motivation of our work is to show how those who are concerned with generating undesirable text language models can obtain some control over the generation process. We look at normativity, but also show how other criteria can be applied. However, as is true for most algorithms, those with malicious intent can find ways to corrupt the intentions of the work. For example, equation (3) can be trivially modified to punish normative text instead of non-normative text.

6 Conclusions

We have shown that large-scale transformer-based neural language models can be made to generate text containing fewer descriptions of non-normative behavior by applying data-efficient, policy-gradient reinforcement learning. As most large-scale language models are trained on datasets from the internet and from books, the potential for intentional or unintentional non-normative language persists. We see this as a first step toward decreasing the potential for unintended, unacceptable, anachronistic or harmful language.

While our primary result is to show that we can decrease the generation of non-normative behavior descriptions, our normative classifier of choice is rooted in Western/American norms and values. Normative classifiers are rare and datasets containing normative or preference learning examples are difficult to obtain. However, our results show that even small datasets of normative examples can be converted into few-shot classifiers and applied to

new domains. By replicating our results with sentiment and toxic classifier, we show that our technique is not specific to any one classifier.

References

David Abel, James MacGlashan, and Michael L Littman. 2016. Reinforcement learning as a framework for ethical decision making. In *Workshops at the Thirtieth AAAI Conference on Artificial Intelligence*.

Prithviraj Ammanabrolu, Ethan Tien, Wesley Cheung, Zhaochen Luo, William Ma, Lara J Martin, and Mark O Riedl. 2019. Story realization: Expanding plot events into sentences. *arXiv preprint arXiv:1909.03480*.

Thomas Arnold, Daniel Kasenberg, and Matthias Scheutz. 2017. Value alignment or misalignment–what will keep systems accountable? In *Workshops at the Thirty-First AAAI Conference on Artificial Intelligence*.

Cristina Bicchieri. 2005. *The grammar of society: The nature and dynamics of social norms*. Cambridge University Press.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, pages 4299–4307.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2019. Plug and play language models: a simple approach to controlled text generation. *arXiv preprint arXiv:1912.02164*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Andrea Esuli and Fabrizio Sebastiani. 2006. Sentiwordnet: A publicly available lexical resource for opinion mining. In *LREC*, volume 6, pages 417–422. Citeseer.

Spencer Frazier, Md Sultan Al Nahian, Mark Riedl, and Brent Harrison. 2019. Learning norms from stories: A prior for value aligned agents. *arXiv preprint arXiv:1912.03553*.

Mary Fainsod Katzenstein. 1996. *The culture of national security: Norms and identity in world politics*. Columbia University Press.

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*.

Dimitrios Kotzias, Misha Denil, Nando De Freitas, and Padhraic Smyth. 2015. From group to individual labels using deep features. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 597–606.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849.

Moin Nadeem, Anna Bethke, and Siva Reddy. 2020. Stereoset: Measuring stereotypical bias in pretrained language models. *arXiv preprint arXiv:2004.09456*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Stuart Russell, Daniel Dewey, and Max Tegmark. 2015. Research priorities for robust and beneficial artificial intelligence. *Ai Magazine*, 36(4):105–114.

Emily Sheng, Kai-Wei Chang, Premkumar Natarajan, and Nanyun Peng. 2019. The woman worked as a babysitter: On biases in language generation. *arXiv preprint arXiv:1909.01326*.

Nate Soares and Benja Fallenstein. 2014. Aligning superintelligence with human interests: A technical research agenda. *Machine Intelligence Research Institute (MIRI) technical report*, 8.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, and Jasmine Wang. 2019. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*.

Leonard Wayne Sumner. 1967. Normative ethics and metaethics. *Ethics*, 77(2):95–106.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing nlp. *arXiv preprint arXiv:1908.07125*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. *CoRR*, abs/1905.12616.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.