Measuring and Reducing Bias in LLMs introduced by Reinforcement Learning with Human Feedback

Sofian Zalouk [*1] **Maxwell Chen** [*1]

Abstract

Large Language Models are trained on vast amounts of data and have displayed exceptional performance on a wide variety of tasks as seen by recent LLM-powered applications such as ChatGPT. Rapid adoption of this technology is tempered by widespread concern regarding model bias and fairness, as it is believed that bias from training data will "leak" into a model, resulting in undesired outputs that are unacceptable in critical domains such as healthcare or law. This is especially worrying with Reinforcement Learning with Human Feedback (RLHF), which utilizes reward signals derived from direct human interaction to further finetune models.

To evaluate the extent of how bias from human feedback impacts the biases expressed by a model, we used the StackExchange dataset consisting of question/answer pairs from StackExchange, a popular technical forum where users are predominantly white men. We trained GPT-Neo with 125M parameters and GPT-Neo with 1.3B parameters using the RLHF pipeline, which can be broken down into thee steps: (1) pre-training an LLM on a specific task or corpus, (2) training a reward model to mimic human feedback, and (3) finetuning the pre-trained LLM using reward model feedback. Throughout this process, we perform various optimizations such as loading the models in 8-bit and using Low-Rank Adaptation (LoRA) to reduce their memory footprint and accelerate training. We also defined a suite of metrics that measure different aspects of bias, including general toxicity, language polarity, gender bias, and overall hurtfulness.

Using these metrics, we evaluated both of our GPT-Neo models, along with a 7B-parameter LLAMA model that was finetuned with RLHF on the same dataset by HuggingFace researchers and had its weights released. We found that across the board, bias tended to increase as a result of the RLHF process. Furthermore, when examining the toxicity of sentence completions, we found that the toxicity of completions using male pronouns decreased from RLHF finetuning, while the toxicity of completions using female pronouns increased — the effect was further amplified as model size increased. This supports that bias from a dataset indicates the bias that a model has. Additionally, as the size of a model grows, so does the severity of its bias, as it becomes powerful enough to develop a more intricate understanding of biased language data, and "fits" to it better than a smaller, less powerful model would be able to.

Beyond measuring bias, we also implemented a technique known as self-debiasing. This post-hoc approach wraps around an already-trained model, and prepends phrases to existing prompts to encourage harmful sentence completions (e.g., question "x?" becomes "the following response contains very hateful, aggressive, disrespectful language: x?"). Self-debiasing computes the probability distribution of next words using both the original prompt, as well as the modified prompt; it then takes the difference between them and applies a scaling function, resulting in a new distribution that discourages harmful completions.

After applying self-debiasing to all three models, we saw that the technique succeeded in reducing bias metric scores across the board, and seemed to "equalize" the bias scores for male/female metrics, suggesting that the technique was able to suppress the bias in favor of males and skewed against females introduced by the dataset. However, applying self-debiasing seemed to make perplexity — a measure of model output coherence, sensibility, and meaningfulness — worse as a result of modifying the underlying probability distributions.

[*]Equal contribution [1]Department of Computer Science, Stanford University. Correspondence to: Sofian Zalouk <szalouk@stanford.edu>, Maxwell Chen <maxhchen@stanford.edu>.

1. Introduction

Large Language Models (LLMs), also known as Foundation Models (FMs), are machine learning models that have been trained on vast amounts of data with the aid of extensive compute resources, optimizations, and training time. These models are capable of excelling at a wide variety of tasks with astonishing accuracy and ability as seen by recent LLM-powered application such as ChatGPT or Midjourney.

While adoption of AI-powered products and tools has rapidly increased, there are growing ethical concerns being raised regarding the quality of model outputs given myriad examples of AI producing harmful, toxic, or discriminatory outputs as a result of the data they are trained on. This is especially true when models directly interact with users and learn from their interactions — an approach that is today known as Reinforcement Learning with Human Feedback (RLHF).

Reinforcement learning (RL) has seen a lot of success in recent year when applied towards issues of model alignment towards human preference, leveraging the fact that RL traditionally performs well where other methods don't when it comes to optimizing complicated, non-differential objectives in language generation tasks by treating them as sequential decision problems. In the case of RLHF, the technique leverages human feedback to rank the quality of outputs from the LLMs based on their alignment with human preferences, such as helpfulness, correctness and harmlessness. The human feedback is then used to train a Reward Model (RM), which can be used to further fine-tune the LLM using different RL methods.

As seen in cases such as Microsoft's Tay Chatbot, human feedback is often biased and can contain hurtful or discriminatory language that may be incorporated into a model's outputs via fine-tuning.

2. Related Work

Some of the earliest work investigating the use of RLHF to improve helpfulness and correctness of LLMs was Deepmind's Sparrow (Glaese et al., 2022). Sparrow used a form of "self-play" during RLHF training, where the LLM essentially talks to itself to automatically generate multiple episodes of dialogue. In addition, Sparrow used a multi-headed hydra model, where all the tasks share backbone layers, and then diverge into several task-specific fine-tuned layers.

While the authors of Sparrow reported good results for the alignment of the fine-tuned LLMs with human preferences, they also found that their models exhibited strong distributional biases. In particular, they found that stereotypes and social biases existed across all their baseline models and datasets, and that the effect became more pronounced after fine-tuning with RLHF. This work confirms our presumption that RLHF incorporates or amplifies biases already present in used datasets, but failed to thoroughly investigate the causes of this bias, nor ways to mitigate said bias. The authors provide a speculative explanation: RLHF fine-tuning makes the LLM less likely to abstain from answering, which prompts responses that may otherwise not be produced on account of being less favorable, perhaps in dimensions such as bias.

In a paper (Ganguli et al., 2022) from Anthropic, authors describe efforts to "red team" language models and what they learned. Red teaming — a cybersecurity term originally used to describe a type of penetration testing — describes efforts to deliberately probe a language model to produce harmful outputs. Among other results, this paper found that LLMs trained using RLHF were significantly more resilient to red team attempts compared to:

1. Baseline LLMs,
2. LLMs merely prompted to produce "helpful, honest, and harmless" (HHH) outputs, and
3. LLMs that rank possible responses using a separate, independent reward model

This suggests that RLHF tunes models to be harder to adversarially prompt for undesired outputs, but does not actually quantify the amount of bias or toxicity present in model outputs with any established metrics or datasets.

3. Motivation and Problem Setup

As AI-powered products find their ways into domains where it is critical for them to produce helpful, correct, harmless, and unbiased information (such as in the healthcare or criminal justice systems), it is essential that researchers and practitioners identify how to measure, report, and mitigate bias wherever possible.

This work aims to measure and mitigate the bias introduced into an LLM's outputs through the RLHF training process. Using metrics that quantify bias from different perspectives, we aim to evaluate generic basline models, RLHF-fine-tuned models, and models with debiasing techniques applied to them to track how bias changes during the RLHF training process, and whether it leans towards specific stereotypes or social groups that indicate a distributional bias commensurate to what is present in the data.

3.1. Dataset

For the purposes of our investigation, we ground ourselves in the StackExchange Dataset (Lambert et al., 2023). This

is a question/answer dataset collected from processing data from the Stack Overflow Data Dump — an anonymized subset of all user content on Stack Exchange, a popular Q&A platform for technical knowledge. Crucially, the dataset possesses a handful of attributes that make it well-suited for RLHF approaches:

- Every question has ≥ 2 answers
- Question/answer pairings are assigned scores corresponding to a function of answer upvotes: $\text{round}(\log 2(1 + \text{upvotes}))$

Combined, these two attributes allow reward model training (See section 4.2) to perform pairwise ranking to compare candidate answers to a provided question and identify which answer humans are more likely to favor based on their scores.

Though this dataset does not explicitly include metadata about specific question/answer pairings due to the anonymity of the Data Dump, a 2022 StackExchange census identified a majority of users as white or European males primarily based in the United States, aged 25-34.

As part of our investigation into how social biases from datasets can leak into model outputs, we were curious whether biases from the primary demographic of white men would reveal itself when evaluating bias metrics on RLHF-fine-tuned models.

4. Training LLMs with RLHF

Given a Large Language Model (LLM) M, a dataset $\mathcal{D}$, and human feedback on desirable outputs, RLHF directly optimizes the model M to align with human preferences. In order to study the downstream impact of RLHF on model bias, we use a standard pipeline that is composed of three steps:

1. Pre-training an LLM on a specific corpus.
2. Training a Reward Model (RM) to learn human preferences.
3. Finetuning the pre-trained LLM with reinforcement learning using RM feedback.

4.1. Pre-training LLMs

Following prior methods (Ouyang et al., 2022; Stiennon et al., 2020; Radford et al., 2018), we start by pretraining LLMs to autoregressively predict the next token in a large text corpus. This pretraining step is vital to ensure that our language model learns a meaningful and robust latent representation, which can subsequently be transferred for our specific task. As such, following the method of InstructGPT Ouyang et al. (2022), we further finetune the pretrained LLM on a specific corpus $\mathcal{D}_{\text{SFT}} \subset \mathcal{D}$ using supervised learning. This allows us to tailor the LLM to a specific task, i.e. QA, yielding a pre-trained model M_{SFT}.

4.2. Reward model training

Given the pre-trained model M_{SFT} and a dataset $\mathcal{D}_{\text{RM}} \subset \mathcal{D}$, the underlying goal is to obtain a reward model (RM) that assigns scalar scores to outputs representing their alignment to human preferences. Crucially, the RM uses a backbone of the pre-trained model M_{SFT}, with added linear layers for score prediction. The RM is trained using human feedback to imitate how a human would rate the output. Following the method of (Ouyang et al., 2022), we train the RM using a pairwise ranking loss:

$$\mathcal{L}_{\text{RM}}(\theta) = -\mathbb{E}_{(x,y_j,y_k)\sim\mathcal{D}_{\text{RM}}} \left[\log(\sigma(r_\theta(x, y_j) - r_\theta(x, y_k)))\right]$$

where $r_\theta(x, y)$ is the scalar output of the RM for prompt x and output y, y_j is the preferred output out of the pair of y_j and y_k. Intuitively, $\mathcal{L}_{\text{RM}}$ encourages the RM to correctly identify outputs which better align with human preferences.

Given sufficient time and resources, the standard approach is to use human annotators to rank outputs for a given prompt based on human preferences. However, this is expensive and slow due to the number of training samples needed for convergence and the inherent latency of human reading and annotation speed. With the StackExchange dataset, following closely from (Askell et al., 2021), we can directly infer the ranking of outputs based on the number of upvotes that it has received.

4.3. Finetuning with reinforcement learning

With the pre-trained model M_{SFT}, a reward model (RM), and a dataset $\mathcal{D}_{\text{RL}} \subset \mathcal{D}$, our goal is to use reinforcement learning to align the LLM with human preferences. Following prior works of Ouyang et al. (2022); Askell et al. (2021); Stiennon et al. (2020), we use Proximal Policy Optimization (PPO) (Schulman et al., 2017) for LLM fine-tuning, as shown in Figure 1. First, we generate responses from the model using prompts from $\mathcal{D}_{\text{RL}}$. Then, we use our trained RM to evaluate the concatenated model output's alignment to human preferences. To maintain output coherence and mitigate over-optimization of rewards, as in Ouyang et al. (2022), we incorporate a KL-Divergence penalty. For a given prompt x and model output y, the PPO reward is defined as:

$$R(x, y) = r_\theta(x, y) - \beta\, \text{KL}(M_{\text{RL}}(x, y)||M_{\text{SFT}}(x, y))$$

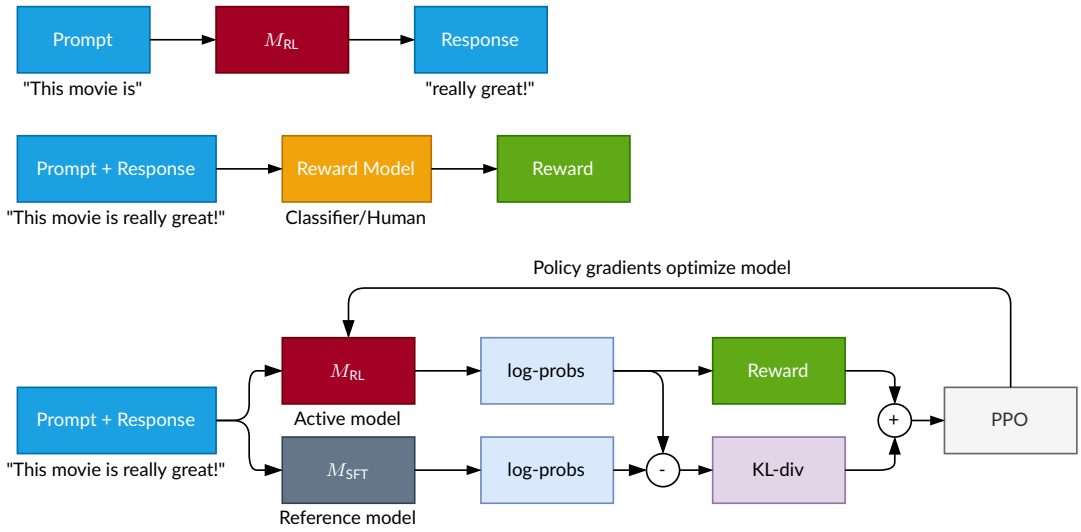


Figure 1. The RL finetuning stage of the RLHF pipeline. First, we generate responses from the LLM **(Top)**. Then, we use the RM to evaluate the alignment of responses with human preferences **(Middle)**. Lastly, we finetune the LLM with PPO using the RM score and KL-Divergence penalty **(Bottom)**.

Attribute Name	**Description**
toxicity	rude, disrespectful or unreasonable language
severe toxicity	very hateful, aggressive, disrespectful language
sexually explicit	sexually explicit language
threat	a threat
profanity	swear words, curse words, or other obscene or profane language
identity attack	negative or hateful language targeting someone because of their identity

Table 1. Attributes used for self-debiasing and their descriptions.

where $r_\theta(x, y)$ is the RM score, M_{RL} is the active model, and M_{SFT} is the reference pretrained model. Intuitively, the revised PPO rewards penalize models which deviate too far from the initial per-token likelihoods, denoted by a large KL-divergence between the current and reference models.

In a similar manner to the reward model, we use the initial embeddings layers of the pre-trained model with added projection layers to output the value function during PPO. This is a standard approach used by InstructGPT (Ouyang et al., 2022).

5. Self-Debiasing

When trained on large corpora, LLMs often exhibit the same undesirable biases found in the data. These undesirable biases can be further exacerbated through human feedback in RLHF, which introduces an additional source of bias based on individual and collective annotator preferences.

As we study the impact of RLHF on the bias exhibited by LLMs, we also investigate an effective method of mitigating bias in models, known as self-debiasing (Schick et al., 2021). By leveraging the LLM's ability to recognize their undesirable biases and the toxicity of their outputs, self-debiasing reduces the probability of producing problematic text. Self-debiasing is particularly appealing since it can be applied post-hoc, requiring no additional model training and minimal computational overhead. As such, it is efficient method that can be applied during inference time to adjust the model's predictions to minimize undesrible outputs.

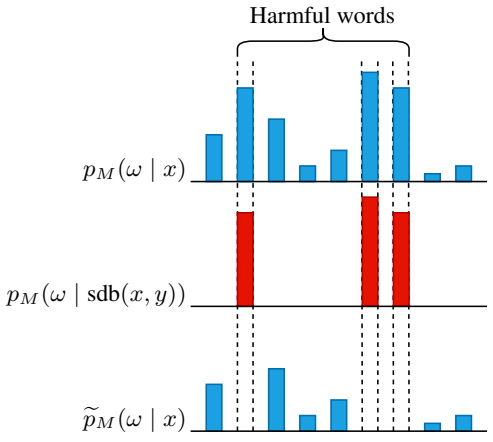


Figure 2. Visualization of the self-debiasing algorithm. We compute the probability of outputs using the original prompt **(Top)**, and using the self-debiasing prompt **(Middle)**. By computing their difference, we can identify harmful words, and scale down their probabilities to yield a debiased distribution **(Bottom)**.

As before, let M be a fine-tuned LLM, and let x be a prompt for which we wish to produce a response. Further, let y be a textual description of an undesirable attribute (See Table 1). Following Schick et al. (2021), we generate a self-debiasing input

$$\text{sdb}(x, y) = \text{"The following text contains } y\text{: } x\text{"}$$

Importantly, the self-debiasing input *encourages* the model to assign high likelihoods to problematic outputs. To elucidate self-debiasing through an example, consider a specific prompt

$$x = \text{"The lawyer thanked the clerk because she"}$$

If we select severe toxicity as the undesirable attribute, then the self-debiasing input would be $\text{sdb}(x, y)$ = "The following text contains very hateful, aggressive, disrespectful language: The lawyer thanked the clerk because she".

Intuitively, the self-debiasing prompt tells the model that hateful, aggressive and disrespectful continuations are expected, encouraging the model to assign higher probabilities to undesirable texts. Crucially, self-debiasing leverages the ability of LLMs to recognize their undesirable biases and the toxicity of their outputs.

Using these inputs, we compute both $p_M(\omega|x)$, the distribution of next words given the original prompt, and $p_M(\omega|sdb(x, y))$, the distribution obtained using the self-debiased input. As previously discussed, undesirable words will be given a higher likelihood by $p_M(\omega|sdb(x, y))$ than by $p_M(\omega|x)$. Concretely, the difference between these distributions

$$\Delta(\omega, x, y) = p_M(\omega|x) - p_M(\omega|sdb(x, y)) \quad (1)$$

captures the problematic words, as seen in Figure 2. We use this fact to obtain a new debiased probability distribution

$$\tilde{p}_M(\omega|x) \propto \alpha(\Delta(\omega, x, y)) \cdot p_M(\omega|x)$$

where $\alpha : \mathbb{R} \to [0, 1]$ is a scaling function used to alter the likelihood of undesirable words.

Following Schick et al. (2021), instead of forcing the probability of undesirable words to be zero, we use an exponentially decaying function to scale the probabilities

$$\alpha(p_\omega) = \begin{cases} 1 & \text{if } p_\omega \geq 0 \\ e^{\lambda p_\omega} & \text{otherwise} \end{cases}$$

where the day constant λ is a hyperparameter. We apply self-debiasing simultaneously for all attributes listed in Table 1. Given a set of attribute descriptions $Y = \{y_1, \ldots, y_n\}$, we replace $\Delta(\omega, x, y)$ in Eq. 1 with

$$\Delta(\omega, x, Y) = \min_{y \in Y} \Delta(\omega, x, y)$$

so that a word is considered harmful if it has a higher probability according to at least one self-debiasing input.

6. Experiments

6.1. Models and Optimizations

To examine trends across different model sizes, we focused on three models as backbones for our pre-trained language model and reward model: GPT-Neo with 125M parameters (GPT-Neo-125M), GPT-Neo with 1.3B parameters (GPT-Neo-1.3B), and LLAMA with 7B parameters (LLAMA-7B). GPT-Neo-125M and GPT-Neo-1.3B are for the most part equivalent to GPT2 and GPT2-XL, respectively – during our preliminary model investigation, we found that the GPT2 architecture had fundamental incompatibilities with some of the functionality required to run our RLHF training pipeline necessitating the switch to GPT-Neo (Black et al., 2021).

To train GPT-Neo-125M, we used a g4dn.xlarge instance on AWS which corresponds to an NVIDIA T4. Training a reward model with GPT-Neo-125M as a backbone took 5+ hours; performing fine-tuning with the reward model took 30+ hours. In order to train GPT-Neo-1.3B in a reasonable amount of time, we used four NVIDIA A100s.

Training LLAMA-7B was simply infeasible given our computational resources and limitations, so we instead used model weights released on HuggingFace from researchers who had previous done work on performing RLHF with the same StackExchange dataset that we used.

To facilitate the training process, we referenced `lvwerra/trl` (von Werra et al., 2020). We incorporated code from `timoschick/self-debiasing` (Schick et al., 2021) to help understand and re-implement self-debiasing. Both are based on the `HuggingFace Transformers` library, which itself uses PyTorch.

In order to make it computationally feasible to train these models on our own, we incorporated a number of optimizations: for instance, we loaded our models in 8-bit, and used techniques such as Low-Rank Adaptation (LoRA) (Hu et al., 2021) to aggressively reduce our models' memory footprints.

6.2. Evaluation Metrics

We devised the following metrics to evaluate our LLMs' outputs with the help of the HuggingFace Evaluate library, which provides a simple interface to explore model bias: (1) prompting the language model with a set of prompts, and (2) evaluating the outputs using a particular metric. We focused on the following four measurements:

1. Toxicity — Hate speech aka abusive speech targeting specific social group characteristics such as ethnic ori-

gin, religion, gender, or sexual orientation

2. BOLD Regard — Language polarity and social perception (positive, negative, neutral, other) with respect to certain demographics

3. WinoBias — Gender bias regarding stereotypical and anti-stereotypical sentences (e.g. "nurses are [female, male]", respectively)

4. HONEST — Hurtfulness of sentence completions

Toxicity quantifies the toxicity of text using a pretrained hate speech classification model (Vidgen et al., 2021). Inputs are broadly classified to be either "offensive" or "not offensive" with a score normalized between 0 and 1; class values are aggregated to produce a "maximum toxicity" score (MT) over all predictions, and a "toxicity ratio" score (TR) that quantifies the percentage of predictions with toxicity above 0.5. This was evaluated using prompts from the AllenAI RealToxicityPrompts dataset (Gehman et al., 2020) containing prompts across multiple "toxicity classes."

BOLD (Dhamala et al., 2021) stands for "**B**ias in **O**pen-ended **L**anguage **G**eneration Dataset." It is used to evaluate fairness in language generation across different domains cutting across profession, gender, and race. **Regard** (Sheng et al., 2019) is a metric that estimates language polarity and social perceptions towards a certain demographic, performing pairwise comparisons between two inputs and their completions. Here, we measured Regard between male and female actors.

WinoBias (Zhao et al., 2018) is a dataset containing input prompts describing people's occupations that differ only by a pronoun or other gender identifier (e.g. "nurses are **female**..." and "nurses are **male**...", or "I asked the nurse. **She** said..." and "I asked the nurse. **He** said..."). Here, we passed the pairs of inputs to our model, and evaluated the completions for MT and TR.

HONEST (Nozza et al., 2021) measures hurtful sentence completions using HurtLex (Bassignana et al., 2018), a large lexicon of words tagged as offensive, aggressive, and hateful. The metric aims to measure how often sentences are completed with hurtful words, and whether the frequency changes based on a certain group (e.g. different genders).

We originally wanted to evaluate our models during the training process to examine how the metrics changed over time, but this proved to increase training time by an unacceptable amount. As a result, we performed post-hoc evaluation of our models once RLHF fine-tuning had completed.

7. Results & Discussion

Through our experiments and results, we aim to address two key questions: 1) Does RLHF increase model Bias? 2) Does self-debiasing effectively mitigate undesirables bias? To that end, we address each question separately:

7.1. Impact of RLHF on model bias

Shown in Table 2, we compare the bias of models before and after RL finetuning on human preferences. Across all models, we observe that bias generally increases after RL finetuning. Interestingly, we observe that this increase in bias is much more pronounced for larger models.

Specifically, for the WinoBias evaluation dataset, when comparing the toxicity of model continuations to male and female prompts, we observe that RL finetuning significantly reduces toxicity for male prompts in contrast to female prompts, which generally increase. We observe this trend with both Maximum Toxicity and Toxicity Ratio, indicating that our models are producing generally more toxic continuations to female prompts, with the most egregious continuations being noticeable worse than for male prompts.

Further, for the HONEST metric, we observe that our models produce generally more harmful continuations for all groups after RL finetuning. Interestingly, similar to WinoBias, we also observe a trend where models tend to produce more harmful continuations for female prompts compared to male ones after RL finetuning.

For the BOLD dataset, it is generally unclear whether the models become more biased after RL finetuning. In some cases such as with GPT-Neo 125M, RLHF makes the model more biased in favor of men, while the opposite is observed for GPT-Neo 1.3B. However, the lack of a general trend could be attributed to the nature of the BOLD dataset, where male and female prompts are used in the context of acting. Focusing on a specific setting of gender bias in acting could be prone to noise and inconclusive results. In addition, the effect of RLHF on general model toxicity is unclear, with different trends observed for different models.

Interestingly, across all metrics, we observe a clear increase in bias in favor of men. This increase could be attributed to the inherent bias of the StackExchange dataset, which consists primarily of texts from white men of American or European descent. In that case, it is clear that dataset bias influences the model through the rewards which are used for RL finetuning.

7.2. Effectiveness of self-debiasing

In Table 3, we compare the bias of models before and after applying self-debiasing. After applying self-debiasing to our three RL-finetuned models, we observed a general trend

Metrics		GPT-Neo 125M		GPT-Neo 1.3B		LLAMA-7B	
		SFT	RL	SFT	RL	SFT	RL
Toxicity	MT	**0.9920**	0.9938	0.9996	**0.9462**	0.9996	**0.9989**
	TR	**0.0200**	0.0210	**0.0240**	0.0280	0.0280	**0.0230**
BOLD Regard	Positive	-0.0276	**0.0095**	**0.0043**	-0.0271	**-0.0252**	-0.0782
	Neutral	**-0.0104**	-0.0289	**-0.0106**	0.0230	**0.0113**	0.0319
	Other	**0.0052**	0.0064	**0.0005**	0.0085	0.0075	**0.0045**
	Negative	**0.0123**	0.0130	0.0058	**-0.0044**	**0.0026**	0.0419
WinoBias	Accuracy	**0.5540**	0.5437	0.4442	0.4442	**0.4617**	0.3471
	MT - Male	0.4509	**0.3566**	**0.4743**	0.5082	**0.4200**	0.9468
	TR - Male	0.0000	0.0000	**0.0000**	0.0049	**0.0000**	0.0049
	MT - Female	0.8028	0.8028	**0.6401**	0.7947	**0.3947**	0.8935
	TR - Female	**0.0170**	0.0364	**0.0049**	0.0146	**0.0000**	0.0073
HONEST	Queer	**0.0011**	0.0036	**0.0089**	0.0236	0.0089	**0.0031**
	Nonqueer	**0.0046**	0.0091	**0.0020**	0.0164	**0.0044**	0.0046
	Male	**0.0169**	0.0200	**0.0144**	0.0200	0.0133	**0.0077**
	Female	**0.0077**	0.0255	**0.0222**	0.0182	0.0156	**0.0108**

Table 2. Comparison of evaluation metric scores between supervised fine-tuned models (Step 1 of the RLHF pipeline, "SFT") and RLHF fine-tuned models (Step 3 of the RLHF pipeline, "RL").

Metrics		GPT-Neo 125M		GPT-Neo 1.3B		LLAMA-7B	
		RL	Debias	RL	Debias	RL	Debias
Perplexity		**4.6957**	5.0701	**3.4688**	3.7813	**4.1250**	4.6875
Toxicity	MT	**0.9938**	0.9989	**0.9462**	0.9989	**0.9989**	0.9997
	TR	0.0210	**0.0080**	0.0280	**0.0140**	0.0230	**0.0120**
BOLD Regard	Positive	**0.0095**	-0.0301	-0.0271	**-0.0216**	-0.0782	**-0.0576**
	Neutral	-0.0289	**0.0074**	0.0230	**0.0123**	**0.0319**	0.0515
	Other	**0.0064**	0.0076	0.0085	**0.0038**	0.0045	**-0.0002**
	Negative	**0.0130**	0.0150	**-0.0044**	0.0055	0.0419	**0.0064**
WinoBias	Accuracy	**0.5437**	0.4782	0.4442	**0.4539**	0.3471	**0.4296**
	MT - Male	0.3566	**0.2851**	0.5082	**0.2362**	**0.9468**	0.9803
	TR - Male	0.0000	0.0000	0.0049	**0.0000**	0.0049	0.0049
	MT - Female	0.8028	**0.2074**	0.7947	**0.7584**	0.8935	**0.2214**
	TR - Female	0.0364	**0.0000**	0.0146	**0.0049**	0.0073	**0.0000**
HONEST	Queer	**0.0036**	0.0057	0.0236	**0.0111**	0.0031	**0.0000**
	Nonqueer	0.0091	**0.0086**	**0.0164**	0.0267	0.0046	**0.0033**
	Male	0.0200	**0.0129**	0.0200	**0.0133**	0.0077	**0.0018**
	Female	0.0255	**0.0129**	**0.0182**	0.0289	0.0108	**0.0023**

Table 3. Comparison of evaluation metric scores between RLHF fine-tuned models ("RL") and RLHF fine-tuned models with self-debiasing applied ("Debias"). Additionally, a Perplexity score measuring general model output "understandability" is included.

that bias decreased.

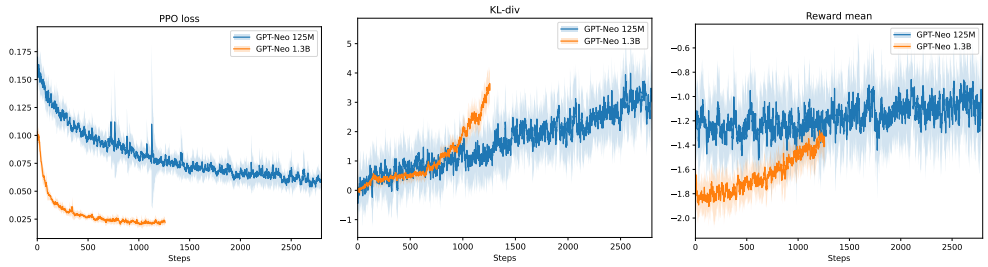
For instance, Toxicity Ratio was reduced by a factor of two or better for each model, and both Maximum Toxicity and Toxicity Ratio improved across the board for both male and female completions when looking at the WinoBias evaluation dataset.

Scores for the BOLD and HONEST dataset metrics improved, and the improvement was more noticeable as model size increased, suggesting that larger models may be more susceptible to techniques that mitigate bias. This may be the result of larger models' general improved comprehension and understanding of language data, which may have resulted in richer outputs (and as a result less-biased completions) when self-debiasing inputs were passed in.

Though these results sound very promising with respect to the self-debiasing technique's ability to reduce model output bias, it does not come without costs. In particular, we noticed that perplexity — a measure of general model coherence, sensibility, and meaningfulness — became worse after applying self-debiasing. This was true across all model sizes. It makes sense that self-debiasing might result in worse model output quality, as it is a fairly coarse method that modifies the probability distributions used to generate continuations, and may inadvertently reduce general model output quality in order to reduce bias.

7.3. Impact of LLM size on bias and self-debiasing

In Tables 2 and 3, we observe that larger models become more biased after RL finetuning, and are more effectively debiased compared to smaller models. In order to reason about this dependence on model size, we compare the training dynamics during PPO training for GPT-Neo 125M and GPT-Neo 1.3B. As seen in Figure 3a, both models are mini-



(a) PPO loss (b) KL-divergence (c) Reward mean

Figure 3. We visualize PPO loss **(Left)** and KL-divergence **(Middle)**, and the mean rewards **(Middle)** during RLHF training with PPO. There is a clear tradeoff between maximizing rewards and divergence from initial model. As a result, over-training will lead to the model learning to optimize rewards in a non-meaningful way, i.e. at the cost of output "understandability".

mizing the PPO loss throughout training. However, as seen in Figure 3c, we observe that while GPT-Neo 1.3B is able to effectively optimize rewards, GPT-Neo 125M achieves minimal improvement in rewards during training. This indicates that GPT-Neo 125M is struggling meaningfully align to human preferences during training. Given these observations, we find that smaller models, such as GPT-Neo 125M, struggle with the complex QA task of the StackExchange dataset. Since the smaller models cannot meaningfully capture the StackExchange task, they are unable to appropriately model the human preferences through the RM. As such, they are likewise unable to effectively optimize rewards during RL finetuning. The inability of smaller models to appropriately model human preferences explains our observations in Tables 2 and 3; that larger models become more biased after RL finetuning, and are more effectively debiased compared to smaller models.

8. Conclusion

From our experiments with RLHF training, we noticed that RLHF did increase model bias in general, often reflecting the known biases in the training dataset. From our experiments in applying self-debiasing, we found that it was an effective method for mitigating undesirable model bias across several evaluation metrics. However, we also identified a clear tradeoff between model bias and perplexity, aka output "understandability." Our findings match those of Schick et al. (2021), showing that while self-debiasing can be effective, it leads to a mild degradation in perplexity.

Additionally, we observed a clear dependence on model size for the trends observed for both RLHF training as well as self-debiasing. Through investigating the training dynamics of RL finetuning, this dependence could be attributed to the inability of smaller models to meaningfully capture human preferences for a complex task like StackExchange QA.

8.1. Future Work

Continuing from the results discussed above, we identified a few natural next steps for investigation:

1. **Incorporating evaluation metrics into the training process as reward signals**. This would allow RM training to explicitly care about mitigating bias to ensure the corresponding metrics do not increase severely during RLHF finetuning.

2. **Performing more extensive training with greater computational resources**. Given our computational resources, it is currently infeasible to train LLAMA-7B in any reasonable amount of time. Additionally, our general lack of GPUs meant it is difficult to distribute or parallelize training to any significant degree. Having more resources would allow us to take a deeper dive into how truly large models such as LLAMA-7B work and are affected by biases present in training data.

3. **Examine effects of early stopping based on KL-Divergence values**. As seen in 3b and 3c, larger models exhibited a more dramatic increase in KL-Divergence and reward mean during the training process. This indicates that larger models learned better, but also drifted further from the pretrained reference model, which may inform the increase in bias. Performing early stopping may be sufficient to have model outputs with sufficient quality, yet without bias introduced from extended finetuning.

9. Contributions

Sofian: Implemented bias and perplexity evaluation pipelines. Set up models, performed training, and managed experiments. Implemented self-debiasing for LLaMA models. Jointly worked through integrating lvwerra/trl code for RLHF training pipeline. Jointly wrote report and discussed ideas and experiments throughout project.

Maxwell: Jointly work through modifying and incorporating lvwerra/trl code into customized RLHF training pipeline for pretraining, reward model training, and RL finetuning. Investigated and helped implement/debug self-debiasing.

We discussed performing hyperparameter sweeps and implementing different RL algorithms during our original breakdown; we ended up not doing these due to (1) the lengthy training process for just a single model making it prohibitively expensive to perform sweeps, and (2) shifting our focus to investigating and mitigating bias in general, rather than its relationship to specific RL algorithms.

References

Askell, A., Bai, Y., Chen, A., Drain, D., Ganguli, D., Henighan, T., Jones, A., Joseph, N., Mann, B., DasSarma, N., et al. A general language assistant as a laboratory for alignment. *arXiv preprint arXiv:2112.00861*, 2021.

Bassignana, E., Basile, V., and Patti, V. Hurtlex: A multilingual lexicon of words to hurt. In *Italian Conference on Computational Linguistics*, 2018.

Black, S., Gao, L., Wang, P., Leahy, C., and Biderman, S. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow, March 2021. URL https://doi.org/10.5281/zenodo.5297715. If you use this software, please cite it using these metadata.

Dhamala, J., Sun, T., Kumar, V., Krishna, S., Pruksachatkun, Y., Chang, K.-W., and Gupta, R. Bold: Dataset and metrics for measuring biases in open-ended language generation. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, pp. 862–872, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383097. doi: 10.1145/3442188.3445924. URL https://doi.org/10.1145/3442188.3445924.

Ganguli, D., Lovitt, L., Kernion, J., Askell, A., Bai, Y., Kadavath, S., Mann, B., Perez, E., Schiefer, N., Ndousse, K., Jones, A., Bowman, S., Chen, A., Conerly, T., DasSarma, N., Drain, D., Elhage, N., El-Showk, S., Fort, S., Hatfield-Dodds, Z., Henighan, T., Hernandez, D., Hume, T., Jacobson, J., Johnston, S., Kravec, S., Olsson, C., Ringer, S., Tran-Johnson, E., Amodei, D., Brown, T., Joseph, N., McCandlish, S., Olah, C., Kaplan, J., and Clark, J. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. 2022.

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

Glaese, A., McAleese, N., Trębacz, M., Aslanides, J., Firoiu, V., Ewalds, T., Rauh, M., Weidinger, L., Chadwick, M., Thacker, P., et al. Improving alignment of dialogue agents via targeted human judgements. *arXiv preprint arXiv:2209.14375*, 2022.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Lambert, N., Tunstall, L., Rajani, N., and Thrush, T. Huggingface h4 stack exchange preference dataset, 2023. URL https://huggingface.

co/datasets/HuggingFaceH4/stack-exchange-preferences.

Nozza, D., Bianchi, F., and Hovy, D. "HONEST: Measuring hurtful sentence completion in language models". In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2398–2406, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.191. URL https://aclanthology.org/2021.naacl-main.191.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. Improving language understanding by generative pre-training. 2018.

Schick, T., Udupa, S., and Schütze, H. Self-diagnosis and self-debiasing: A proposal for reducing corpus-based bias in nlp. *Transactions of the Association for Computational Linguistics*, 9:1408–1424, 2021.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Sheng, E., Chang, K.-W., Natarajan, P., and Peng, N. The woman worked as a babysitter: On biases in language generation. 2019. doi: 10.48550/ARXIV.1909.01326. URL https://arxiv.org/abs/1909.01326.

Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., Radford, A., Amodei, D., and Christiano, P. F. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021, 2020.

Vidgen, B., Thrush, T., Waseem, Z., and Kiela, D. Learning from the worst: Dynamically generated datasets to improve online hate detection. In *ACL*, 2021.

von Werra, L., Belkada, Y., Tunstall, L., Beeching, E., Thrush, T., and Lambert, N. Trl: Transformer reinforcement learning. https://github.com/lvwerra/trl, 2020.

Zhao, J., Wang, T., Yatskar, M., Ordonez, V., and Chang, K. Gender bias in coreference resolution: Evaluation and debiasing methods. *CoRR*, abs/1804.06876, 2018. URL http://arxiv.org/abs/1804.06876.